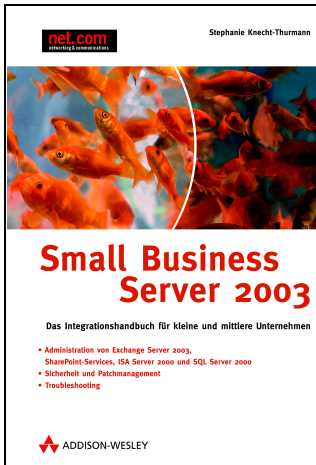


VPN – Virtuelle Private Netzwerke

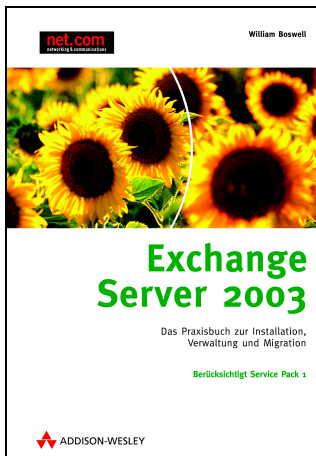
Netzwerke, Betriebssysteme, Sicherheit ... hierzu bietet Ihnen die Reihe net.com umfassende, praxisnahe Information. Neben Fragen der Systemverwaltung greift sie auch Themen wie Protokolle, Technologien und Tools auf. Profitieren Sie bei Ihrer täglichen Arbeit vom Praxiswissen unserer erfahrenen Autoren.



Small Business Server 2003

Stephanie Knecht-Thurmann
504 Seiten, 49,95
3-8273-2213-8

rieren und betreiben. Eine klare Gliederung in die Themen Betriebssystem, Server und Troubleshooting ermöglichen eine schnelle Orientierung. Ein besonderer Schwerpunkt ist auf die Vermittlung von Hintergrundwissen gelegt, denn nur wer weiß, wie und warum etwas funktioniert, ist vor bösen Überraschungen gefeit.



Exchange Server 2003

William Boswell
848 Seiten, € 69,95 [D]
ISBN 3-8273-2227 8

William Boswell beantwortet die drei Schlüsselfragen eines jeden Administrators: Wie funktionieren die Features tatsächlich? Wie hole ich das meiste aus meinem System heraus? Was mache ich bei Problemen? Dabei legt er ein besonderes Augenmerk auf Systemabhängigkeiten und beschreibt z.B., wie Exchange problemlos mit Outlook und anderen E-Mail-Clients zusammenarbeitet. Außerdem berücksichtigt er zahlreiche nützliche third-party-tools, die dem Administrator die Arbeit erleichtern und zeigt, dass Exchange 2003 nicht nur ein Mail-Server, sondern ein fester Bestandteil der Kommunikations-Infrastruktur des Unternehmens ist.

Manfred Lipp

VPN Virtuelle Private Netzwerke

Aufbau und Sicherheit



ADDISON-WESLEY

An imprint of Pearson Education

München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <<http://dnb.ddb.de>> abrufbar.

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

Fast alle Hard- und Softwarebezeichnungen und weitere Stichworte und sonstige Angaben, die in diesem Buch verwendet werden, sind als eingetragene Marken geschützt. Da es nicht möglich ist, in allen Fällen zeitnah zu ermitteln, ob ein Markenschutz besteht, wird das ® Symbol in diesem Buch nicht verwendet.

Umwelthinweis:

Dieses Buch wurde auf chlorfrei gebleichtem Papier gedruckt.

10 9 8 7 6 5 4 3 2 1

08 07 06

ISBN-13: 978-3-8273-2252-4

ISBN-10: 3-8273-2252-9

© 2006 by Addison-Wesley Verlag,

ein Imprint der Pearson Education Deutschland GmbH,

Martin-Kollar-Straße 10–12, D-81829 München/Germany

Alle Rechte vorbehalten

Einbandgestaltung: Marco Lindenbeck, webwo GmbH, (mlindenbeck@webwo.de)

Lektorat: Sylvia Hasselbach, shasselbach@pearson.de

Korrektur: Sandra Gottmann, Münster

Herstellung: Claudia Bäurle, cbaeurle@pearson.de

Satz: mediaService, Siegen, www.media-service.tv

Druck und Verarbeitung: Bercker, Kvelaer

Printed in Germany

Inhaltsverzeichnis

Vorwort	15
Aufbau	15
Danksagung	16
1 Virtuelle Private Netze	17
1.1 Was ist ein VPN?	17
1.2 Netze im Wandel	19
1.2.1 Übertragungstechnologie	19
1.2.2 Datennetze	23
1.2.3 Netzwerkapplikationen	26
1.3 Warum VPN?	29
1.3.1 Veränderung der Geschäftsprozesse	29
1.3.2 Globalisierung und Dezentralisierung	31
1.3.3 Veränderung der Wettbewerbssituation	31
1.3.4 Mobilität und Flexibilität	31
1.3.5 Kostenoptimierung	32
1.3.6 Sicherheit	33
1.4 Randbedingungen für den Einsatz	34
1.5 Der VPN-Markt	35
1.6 Internet-VPN	37
1.7 VPN-Typen	40
1.7.1 Remote Access VPN	41
1.7.2 Branch Office VPN	43
1.7.3 Extranet-VPN	45
1.8 VPN-Dienste	46
1.8.1 Eigenbetrieb	46
1.8.2 Access-Equipment-Outsourcing	47
1.8.3 VPN- und Access-Equipment-Outsourcing	47
1.8.4 Vollständiges Outsourcing	48
1.9 Intranet-VPN	48

2	Anforderungen an VPN	49
2.1	Sicherheit	49
2.1.1	Datenvertraulichkeit	49
2.1.2	Schlüsselmanagement	50
2.1.3	Paketauthentifizierung	50
2.1.4	Datenintegrität	51
2.1.5	Benutzerauthentifizierung	51
2.1.6	Benutzerautorisierung	52
2.1.7	Schutz vor Sabotage	52
2.1.8	Schutz vor unerlaubtem Eindringen	53
2.2	Verfügbarkeit	54
2.2.1	Die Verfügbarkeit von Wahl- und Festverbindungen	54
2.2.2	Die Verfügbarkeit von Internet-VPN	55
2.3	Quality of Service	56
2.3.1	QoS in Sprachnetzen	56
2.3.2	QoS im LAN	57
2.3.3	QoS im WAN	58
2.3.4	QoS in IP	63
2.3.5	QoS im VPN	64
2.4	Skalierbarkeit und Migrationsfähigkeit	65
2.5	Integration in existierende Netze	65
2.5.1	Management	66
2.5.2	Sicherheit	69
2.6	Koexistenz zu traditionellen WAN	71
2.7	Adressmanagement	72
2.8	Interoperabilität	73
3	Tunneling	75
3.1	Tunneling-Modelle	76
3.1.1	Das Intra-Provider-Modell	76
3.1.2	Das Provider-Enterprise-Modell	76
3.1.3	Das Ende-zu-Ende-Modell	77
3.2	Standardisierte Tunneling-Protokolle	78
3.2.1	Layer-2-Protokolle	78
3.2.2	Layer 2 Tunneling Protocol (L2TP)	79
3.2.3	Layer-3-Protokolle	81
3.2.4	IP Security (IPSec) im Tunnel-Modus	81
3.2.5	Multi Protocol Label Switching (MPLS)	82
3.3	Nichtstandardisierte Tunneling-Protokolle	86
3.3.1	Layer 2 Forwarding (L2F)	87
3.3.2	Point-to-Point Tunneling Protocol	87

3.4	Verschachtelte Tunneling-Protokolle	88
3.5	Welches Protokoll für welchen Zweck?	90
3.6	Auswahlkriterien	91
4	Sicherheitstechnologie	93
4.1	Sicherheit in VPN	93
4.1.1	Sicherheit in Unternehmensdatennetzen	93
4.1.2	Sicherheitsverfahren in VPN	96
4.1.3	Sicherheit in der Netzwerkschicht mit IP-Security	98
4.1.4	Sicherheit auf der Transportschicht mit Transport Layer Security (TLS) und Secure Socket Layer (SSL)	100
4.2	Die Grundlagen der Kryptografie	101
4.2.1	Geschichtliches	101
4.2.2	Datenvertraulichkeit	103
4.2.3	Verschleierung und Verschlüsselung	103
4.2.4	Die Kunst der Kryptoanalyse	105
4.2.5	Einführung in die Kryptografie	107
4.2.6	Verschlüsselungsverfahren	113
4.3	Symmetrische Verschlüsselungsverfahren	116
4.4	Der Data Encryption Standard (DES)	118
4.4.1	Ein Überblick über DES	120
4.4.2	Die DES-Schlüsseltransformation	120
4.4.3	Die DES-Funktion	121
4.4.4	Die DES-Entschlüsselung	123
4.4.5	Die Kryptoanalyse von DES	123
4.5	Triple-DES	123
4.5.1	Die Kryptoanalyse von Triple-DES	125
4.6	Cipher Block Chaining (CBC)	126
4.6.1	Die Funktionsweise von CBC	127
4.7	Neu: Advanced Encryption Standard (AES)	127
4.8	Rijndael	129
4.8.1	Die Mathematik hinter Rijndael	130
4.8.2	Der Rijndael-Algorithmus	133
4.8.3	Entschlüsselung	139
4.8.4	Die Kryptoanalyse von Rijndael	139
4.9	RC4	141
4.9.1	Geschichtliches	142
4.9.2	Der RC4-Algorithmus	142
4.9.3	Die Kryptoanalyse von RC4	145

4.10	Asymmetrische Verschlüsselungsverfahren	146
4.10.1	Die kurze Geschichte der Public-Key-Kryptografie	146
4.10.2	Das Grundprinzip der Public-Key-Kryptografie	148
4.10.3	Mathematische Grundlagen	149
4.11	Das Diffie-Hellman-Verfahren	153
4.11.1	Die Kryptoanalyse des Diffie-Hellman-Verfahrens	154
4.12	Das RSA-Verfahren	155
4.12.1	Die Kryptoanalyse von RSA	156
4.13	Elliptic Curve Cryptography (ECC)	158
4.13.1	Mathematische Grundlagen	158
4.13.2	ECDH (Elliptic Curve Diffie-Hellman)	160
4.13.3	Die Kryptoanalyse von ECC	161
4.14	Zufallszahlen	161
4.15	Hash-Funktionen	163
4.15.1	Hash-Algorithmen	164
4.15.2	Die Kryptoanalyse von MD5 und SHA1	166
5	IP Security (IPSec)	169
5.1	IP Security im Überblick	169
5.1.1	Paketintegrität	169
5.1.2	Paketauthentifizierung	170
5.1.3	Paketvertraulichkeit	170
5.1.4	Verkehrsflussvertraulichkeit	170
5.1.5	Schutz vor wiederholtem Senden von Paketen (Replay-Angriff)	170
5.1.6	Schutz vor weiteren Denial-of-Service-Angriffen	171
5.2	Kryptografische Verfahren in IPSec	171
5.2.1	Datenverschlüsselung	171
5.2.2	Integritätsprüfung und Authentifizierung	173
5.2.3	Schutz vor Replay-Angriffen	174
5.3	Die IPSec-Standards	176
5.3.1	Die IPSec-Architektur	176
5.3.2	Die aktuelle IPSec-Standardisierung	177
5.4	Die IPSec Security Association	177
5.5	Die IPSec Security Policy	180
5.5.1	Die Security Policy in IPSec	180
5.5.2	Die IPSec-Selektoren	180
5.6	Die IPSec-Betriebsmodi	181
5.6.1	Tunnel-Modus	181
5.6.2	Transport-Modus	181
5.7	IPSec-Einsatzszenarien	182
5.7.1	Gateway-zu-Gateway	182
5.7.2	Host-zu-Gateway	183
5.7.3	Host-zu-Host	183

5.8	Die IPSec-Protokolle	183
5.8.1	Die Paketverarbeitung in IPSec	183
5.8.2	Authentication Header (AH)	184
5.8.3	Encapsulating Security Payload (ESP)	187
5.9	IPSec-Implementierung	191
5.9.1	Betriebssystemebene, IPSec-Stack	191
5.9.2	Bump-in-the-Stack (BITS)	191
5.10	Betrachtungen zur IPSec Performance	192
5.11	Zukünftige Entwicklungen	194
6	Das IKE-Protokoll	195
6.1	Das Henne-Ei-Problem	195
6.2	ISAKMP	196
6.2.1	Die Sicherheit von ISAKMP	197
6.2.2	Der ISAKMP-Header	202
6.2.3	Der ISAKMP-Nutzdaten-Header	204
6.3	ISAKMP-Nutzdaten	204
6.3.1	Security Association Payload	204
6.3.2	Proposal Payload	205
6.3.3	Transform Payload	206
6.3.4	Key Exchange Payload	206
6.3.5	Identification Payload	206
6.3.6	Certificate Payload	207
6.3.7	Certificate Request Payload	208
6.3.8	Hash, Signature und Nonce Payload	208
6.3.9	Notification Payload	209
6.3.10	Delete Payload	210
6.3.11	Vendor ID Payload	211
6.4	Die ISAKMP-Austauschvorgänge	211
6.4.1	Phase 1	212
6.4.2	Phase 2	212
6.4.3	Die Austauschvorgänge	212
6.4.4	Das Oakley Key Determination Protocol	213
6.5	Der Aufbau von IKE	216
6.5.1	Perfect Forwarding Secrecy	216
6.5.2	Die Attribute einer IPSec Security Association	217
6.5.3	Die Attribute einer IKE Security Association	218
6.5.4	IKE-Sicherheitsverfahren	221
6.5.5	Die Schlüsselerzeugung in IKE	221
6.5.6	Authentifizierung in IKE	224

6.6	Der IKE-Mainmode	225
6.6.1	Authentifizierung mit Pre-Shared Key	226
6.6.2	Authentifizierung mit digitaler Signatur	229
6.6.3	Authentifizierung mit Public-Key-Verschlüsselung (RSA)	230
6.6.4	Authentifizierung mit revidierter Public-Key-Verschlüsselung (RSA)	232
6.7	Der IKE Aggressive Mode	233
6.7.1	Authentifizierung mit Pre-Shared Secret	235
6.7.2	Authentifizierung mit digitaler Signatur	236
6.7.3	Authentifizierung mit standardisierter und revidierter Public-Key-Verschlüsselung (RSA)	236
6.8	Der IKE Quick Mode	237
6.9	Die Performance von IKE	239
6.9.1	IKE und Hardware-Beschleuniger	240
6.10	NAT mit IKE und IPSec	242
6.10.1	NAT und IPSec	242
6.10.2	Automatische Erkennung von NAT-Routern	243
6.10.3	UDP Encapsulation von IPSec-ESP	247
6.11	IKE Dead Peer Detection (DPD)	250
6.11.1	DPD nach RFC3706	250
6.11.2	Andere DPD-Verfahren	251
6.12	Die Sicherheit von IKE	252
SSL-PN		257
7.1	Geschichtliches	257
7.2	Secure Socket Layer (SSL)	258
7.2.1	Transport Layer Security (TLS)	260
7.2.2	Kryptografie in SSL	261
7.2.3	Die Schlüsselerzeugung in SSL	262
7.2.4	Verschlüsselungsalgorithmen	266
7.2.5	Integritätssicherung und Recordauthentifizierung	267
7.2.6	Schutz vor Replay-Angriffen	268
7.2.7	Fazit	268
7.3	SSL-Funktionsblöcke	269
7.3.1	SSL Handshake Protocol	270
7.3.2	SSL Change Cipher Spec Protocol	276
7.3.3	SSL Alert Protocol	276
7.3.4	SSL Record Protocol	277
7.4	Betrachtungen zur Performance von SSL	282
7.5	SSL-VPN	283
7.5.1	Das SSL-VPN-Gateway	288

7.6	Die Sicherheit von SSL	289
7.6.1	Angriffe auf SSL-Anwendungen und -Implementierungen	289
7.6.2	Protokollimmanente Angriffspunkte	290
7.6.3	Fazit	293
8	L2TP/IPSec-Transport	295
8.1	Das Point-to-Point Protocol (PPP)	295
8.1.1	Remote Access mit PPP	296
8.1.2	Die Komponenten von PPP	296
8.1.3	PPP-Steuerprotokolle und -Dienste	298
8.1.4	Der PPP-Verbindungsaufbau	302
8.2	L2TP	303
8.2.1	Virtueller Remote Access mit L2TP	304
8.2.2	Der LAC (L2TP Access Concentrator)	305
8.2.3	Der LNS (L2TP Network Server)	305
8.2.4	Die L2TP-Tunneling Modelle	305
8.2.5	L2TP-Paketformate	306
8.2.6	L2TP Attribute Value Pairs (AVP)	308
8.2.7	Auf- und Abbau von Tunneln und Calls in L2TP	310
8.2.8	Die Sicherheit von L2TP	311
8.3	L2TP-over-IPSec	312
8.3.1	Die Performance von L2TP/IPSec	314
8.3.2	Die Erzeugung von L2TP/IPSec-Paketen	314
9	Quality of Service in VPN	317
9.1	Quality of Service (QoS)	317
9.2	Qualitätskriterien	320
9.2.1	Verfügbarkeit	320
9.2.2	Verzögerung	322
9.2.3	Verzögerungsvarianz (Jitter)	324
9.2.4	Fehlerrate, Bitfehlerrate	325
9.2.5	Bandbreite	325
9.3	QoS-Technologien	326
9.3.1	TCP-Flusssteuerung	326
9.3.2	Weighted Fair Queueing (WFQ)	328
9.3.3	Random Early Discard (RED), Weighted Random Early Discard (WRED)	332
9.3.4	Strict Priority Queueing	333
9.3.5	Fazit	333
9.4	Flussbasierende Dienstgüte	333

9.5	Klassenbasierende Dienstgüte, DiffServ	335
9.5.1	Die DiffServ-Architektur	335
9.5.2	Die DiffServ-Service-Klassen	337
9.5.3	Der DiffServ-Edge-Router	337
9.5.4	Der DiffServ-PHB-Router	338
9.5.5	Die Implementierung von DiffServ	339
9.6	QoS in VPN	342
9.7	QoS in IPSec VPN	346
9.7.1	DiffServ	346
9.7.2	Einflüsse von IPSec auf die Dienstgüte	347
9.8	QoS in SSL VPN	348
9.8.1	DiffServ	348
9.8.2	Einflüsse von SSL auf die Dienstgüte	349
9.9	QoS in L2TP und L2TP/IPSec VPN	350
9.9.1	DiffServ	350
9.9.2	Einfluss von L2TP/IPSec auf die Dienstgüte	351
10	Access-Technologien	353
10.1	Mobile Technologien	353
10.1.1	GPRS	353
10.1.2	UMTS	356
10.1.3	WLAN	360
10.1.4	Satellitenverbindungen	360
10.2	Ortsgebundene Technologien	362
10.2.1	ADSL	362
10.2.2	SDSL	365
10.2.3	Breitbandkabel	366
10.2.4	Digitale Standardfestverbindungen	367
10.2.5	HSSI	370
10.3	Wählverbindungen	370
10.3.1	Analoge Verbindungen	370
10.3.2	ISDN	371
11	Design und Realisierung	373
11.1	Ein VPN ist auch nur Netzwerk	373
11.2	Die Ist-Aufnahme	374
11.2.1	Technische Aspekte	375
11.2.2	Betriebswirtschaftliche Aspekte	377
11.2.3	Sicherheit	378

11.3	Der Sollzustand	378
11.3.1	Randbedingungen	379
11.3.2	Technische Aspekte	380
11.3.3	Betriebswirtschaftliche Aspekte	381
11.4	Die Übergangsphase	382
11.5	Die Umsetzung der Security Policy	383
11.6	Remote Access VPN	385
11.6.1	VPN-Router für Remote Access	385
11.6.2	VPN-Clients	391
11.6.3	VPN-Router spezifische Clients	391
11.6.4	Universelle VPN-Clients	395
11.7	VPN zur Standortanbindung	396
11.7.1	VPN-Router für kleine und Heimbüros	396
11.7.2	VPN-Router zur Standortverbindung	397
11.8	Beispiele	404
11.8.1	Remote Access in kleineren Netzen	404
11.8.2	Remote Access in großen Netzen	411
11.8.3	Anbindung von kleinen Büros	416
11.8.4	Standortverbindungen	418
A	Anhang	429
A.1	Literatur	429
A.1.1	Access- und Netzwerktechnologie	429
A.1.2	VPN-Technologie	429
A.1.3	Sicherheitstechnologie	429
A.1.4	Netzwerkdesign und Quality of Service	430
A.2	Links	431
	Stichwortverzeichnis	433

Vorwort

Es ist nunmehr fast fünf Jahre her, dass die Erstausgabe dieses Buches im Jahr 2001 erschienen ist. Fünf Jahre, in dieser Branche ist das eine kleine Ewigkeit. So ist denn diese vollständig überarbeitete und ergänzte Auflage auch praktisch ein neues Buch geworden.

Denn aufgrund des Erfolgs der VPN-Technologie, die zunehmend herkömmliche WAN- und Remote-Access-Lösungen ablöst, hat sich folgerichtig auch die Ausrichtung dieses Buches verschoben. Für viele Unternehmen stellt sich mittlerweile nicht mehr die Frage ob, sondern wie VPN eingesetzt werden. Allerdings ist die seit der Jahrtausendwende exponentiell angestiegene Zahl von Angriffen aus dem Internet ein Thema geworden, das in diesem Zusammenhang etwas intensiver beleuchtet werden muss.

So wird in diesem Buch auch verstärkt auf mögliche Probleme der verschiedenen Sicherheitsverfahren eingegangen, um dem Anwender eine fundierte Beurteilungsgrundlage zu geben. Auch aktuelle Tendenzen wie Netzkonvergenz, Quality of Service und SSL-VPN finden ausreichend Berücksichtigung.

Dieses Buch wendet sich an all diejenigen, die ein VPN betreiben oder planen, dies zu tun, und sich deshalb etwas intensiver mit der zugrunde liegenden Technologie befassen möchten. Neben theoretischen Erwägungen und der Erklärung der den heutigen VPNs zugrunde liegenden Standards wird das Ganze durch etliche Tipps und Designbeispiele abgerundet.

Dieses Buch ist jedoch kein Hohelied auf VPNs, sondern deren Technologie, ihre Möglichkeiten, aber auch ihre Grenzen werden dem Leser klar aufgezeigt.

Auch in dieser Auflage wurde besonderer Wert auf Herstellerneutralität gelegt.

Aufbau

Dieses Buch ist in drei Teile gegliedert. Im ersten Teil, den Kapiteln 1 und 2, werden allgemeine Themen rund um virtuelle private Netzwerke behandelt, ebenso die Anforderungen, die heutige Unternehmensnetze an VPN stellen.

Im Mittelteil, der von Kapitel 3 bis Kapitel 8 geht, werden ausführlich die modernen, VPN zugrunde liegenden Technologien, Verfahren und Standards besprochen. Hier werden einmal Tunneling-Verfahren (Kap. 3) und ausführlich die für IPSec- und SSL-VPN relevanten kryptologischen Verfahren (Kap. 4) beschrieben. Die VPN-Protokolle IPSec, IKE, SSL und L2TP werden in den vier darauf folgenden Kapiteln behandelt.

Im letzten Teil, den Kapiteln 9, 10 und 11, werden die Lösungsmöglichkeiten und möglichen Einschränkungen von VPN diskutiert. Insbesondere das Thema Quality of Service, gerade auch wichtig für Voice over IP, findet in Kapitel 9 einen angemessenen Raum. Nachdem in Kapitel 10 die wichtigsten Access-Technologien beleuchtet wurden, geht es in Kapitel 11 an Planung und Design von VPN.

Danksagung

Dieses Buch ist neben meiner hauptberuflichen Tätigkeit entstanden, und aus diesem Grund konnte ich in den letzten Monaten nur sehr wenig Zeit für meine Familie aufbringen. Insbesondere meiner Frau Gaby und meiner kleinen Tochter Marie möchte ich daher an dieser Stelle dafür danken, dass sie meiner Arbeit an diesem Buch trotz der Entbehrungen sehr viel Geduld und Verständnis entgegengebracht haben.

Mein weiterer Dank gilt der Firma Nortel, die mir gestattet hat, dieses Buch als Nebentätigkeit zu veröffentlichen. Insbesondere dafür, dass Nortel überhaupt nicht versucht hat, Einfluss auf den Inhalt dieses Buches nehmen, gilt meine ganz besondere Anerkennung.

Auch dem Team bei Addison-Wesley, hier möchte ich stellvertretend für alle an der Entstehung dieses Buches beteiligten Personen Frau Sylvia Hasselbach, Frau Claudia Bäurle und Frau Sandra Gottmann nennen, gilt mein ganz besonderer Dank.

Manfred Lipp

1 Virtuelle Private Netze

1.1 Was ist ein VPN?

Was ist ein VPN? Ein VPN (virtuelles privates Netzwerk) ist ein Netzwerk, das ein anderes, öffentliches Netzwerk benutzt, um private Daten zu transportieren. Dies ist zugegebenermaßen eine recht allgemein gehaltene Definition, die aber allen Arten von VPN gerecht wird. In der heutigen Zeit wird ein VPN sehr oft nur auf die Benutzung des Internets als öffentliches Netzwerk reduziert, was aber so nicht stimmt.

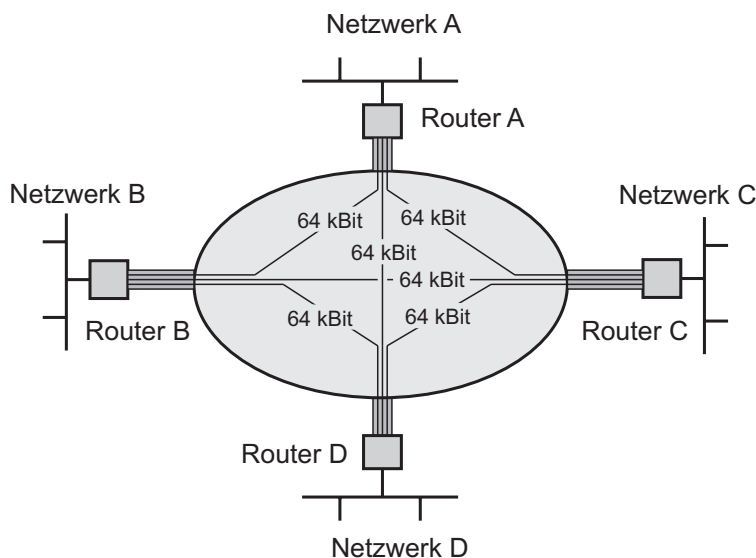


Abbildung 1.1: Ein klassisches Netzwerk basierend auf Festverbindungen

Wie Sie in diesem einleitenden Kapitel sehen werden, gibt es eine ganze Reihe anderer, teilweise schon recht alter VPN-Technologien, die überhaupt keinen Gebrauch vom Internet machen. Auch spricht man in diesem Zusammenhang oft nur von der Übertragung von Daten, aber es gibt auch reine Sprach-VPNs und zunehmend auch so genannte konvergente VPNs, also Netze, die zur Übertragung von Sprache, Daten und interaktiven Video-konferenzen über das IP-Protokoll geeignet sind. Diese Tendenz, für die sich mittlerweile der Ausdruck Konvergenz oder auch Konvergenz der Netze als allgemein gültiger Begriff etabliert hat, führt zu einigen unverzichtbaren Anforderungen an solche Netzwerke. Denn im Gegensatz zur reinen Datenübertragung, bei der es in den meisten Fällen nicht so sehr darauf ankommt, ob die Pakete verzögert ankommen, und die auch mit zeitweise geringen Bandbreiten auskommt, müssen diese neuen, konvergenten Netze für Datenströme bestimmter Dienste gewisse Qualitäten netzwerkweit garantieren können. Diese Forde-

nung, die in lokalen Netzen und bestimmten Arten von öffentlichen Netzen durchaus realisierbar ist, wird in IP-beziehungsweise Internet-VPN zu einem extrem kritischen Faktor, dem in diesem Buch ausführlich Rechnung getragen wird.

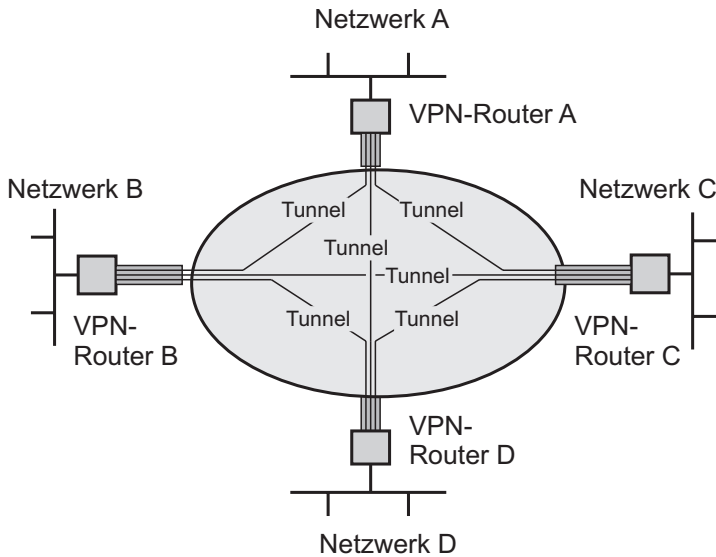


Abbildung 1.2: In einem VPN sind die Festverbindungen durch virtuelle Verbindungen (Tunnel) ersetzt.

Aber zurück zur Definition eines virtuellen privaten Netzwerks. Das Gegenstück zum VPN ist ein echtes privates Netzwerk, also ein Netzwerk, das exklusiv von einem Unternehmen oder einer Organisation betrieben wird. Das heißt, alle Übertragungseinrichtungen und alle Übertragungsmedien gehören diesem Unternehmen oder sind ihm zur exklusiven Nutzung überlassen. Beispiele sind die so genannten Mietleitungen oder Standardfestverbindungen, die einer Organisation zur ausschließlichen Nutzung vermietet werden. Mit geeignetem Equipment zur Daten- oder Sprachübertragung über diese Leitungen wird ein scheinbar rein privates Netzwerk betrieben. Scheinbar deshalb, weil die Verbindungen zwar ausschließlich vom Mieter derselben benutzt werden, jedoch meist, zumindest in Europa, Teil einer öffentlichen Netzwerkinfrastruktur sind. Diese Infrastruktur bietet jedoch umfassende Möglichkeiten zum „Anzapfen“ dieser Verbindung und stellt somit in jedem Fall ein mögliches Sicherheitsrisiko dar – auch wenn die Betreiber solcher Netze natürlich keinen Missbrauch treiben dürfen und ein Abhören nur aufgrund einer richterlichen Anordnung durch bestimmte Personen oder Organisationen zulässig ist.

Ein öffentliches Netzwerk hingegen ist eine Kommunikationsinfrastruktur, die von einem gewinnorientierten Dienstleistungsunternehmen betrieben wird, das die Benutzung des Netzes jedermann gegen ein entsprechendes Verbindungsentgelt ermöglicht. Ein Beispiel hierfür sind öffentliche Telefonnetze. Jeder kann gegen eine entsprechende Gebühr dieses Netz benutzen.

Ein VPN versucht, private und öffentliche Netzwerke zu kombinieren, indem das öffentliche Netzwerk als Trägernetzwerk für die private Kommunikation benutzt wird.

Im weiteren werden Sie allerdings sehen, dass der Begriff VPN mittlerweile sehr unscharf geworden ist und teilweise völlig aus seiner exakten Definition herausgelöst wurde. Insbesondere wird mit VPN oft auch Sicherheit im Hinblick auf Vertraulichkeit oder Integrität übertragener Informationen assoziiert, was jedoch schlichtweg falsch ist: Ein VPN trennt den Transport privater Datenpakete oder privater Datenframes von anderen – sonst nichts. Die einzige Sicherheit, die ein VPN bieten muss, ist die, dass eigene Datenpakete nicht zum falschen Empfänger geleitet werden und umgekehrt und dass sie innerhalb vorbestimmbarer Wege übertragen werden. Weitere Sicherheitsmaßnahmen sind optional.

Leider wird aber auch oft ein Umkehrschluss geführt, und eine sichere Datenübertragung, meist mit Verschlüsselung und Integritätsprüfung, wird als VPN bezeichnet. So haben zum Beispiel Technologien wie SSL (vgl. Kapitel 7) oder der Transportmode von IPSec (vgl. Kapitel 5) eigentlich gar nichts mit VPN zu tun, denn sie bilden kein privates Netz in einem öffentlichen ab. Aber da diese Protokolle oft mit anderen Tunneling-Verfahren kombiniert werden, haben sich diese Bezeichnungen eingebürgert und werden deshalb auch in diesem Buch verwendet.

1.2 Netze im Wandel

Bevor man sich mit dem Thema VPN auseinander setzt und versucht, in dessen Zukunft zu blicken, sollte man sich vielleicht einmal die noch sehr junge Geschichte der Daten-netzwerke ins Gedächtnis rufen, um sich einen Überblick über den Gesamtzusammenhang zu verschaffen. Denn obwohl es elektronische Kommunikation schon seit langem gibt, die Telegrafie und die Telefonie wurden bereits im vorletzten Jahrhundert erfunden, ist die Datenkommunikation im Sinn von rein auf den Transport von Datenpaketen bezogenen Technologien noch keine 50 Jahre alt. Man muss zudem auch, wenn man sich die Entwicklung von Ende der 50er-Jahre bis heute vor seinem geistigen Auge ablaufen lässt, auch zwischen den reinen Übertragungstechnologien, den darauf beruhenden Netzwerktechnologien, insbesondere die Internettechnologien, und den verschiedenen Applikationen, die letztendlich auch nicht nur davon profitiert haben, sondern auch gleichzeitig treibende Kräfte waren, differenzieren.

1.2.1 Übertragungstechnologie

Unter dem Begriff Übertragungstechnologie ist hier ganz allgemein all das zusammengefasst, was an Protokollen und Technologien auf den unteren beiden Schichten des OSI-Referenzmodells bzw. des TCP/IP-Modells angesiedelt ist.

In ihren Anfängen war die Übertragungstechnologie vor allem noch durch die Anforderungen aus der klassischen, leitungsvermittelnden Telefonie geprägt. Auch die bis Anfang der 70er-Jahre des letzten Jahrhunderts noch extrem seltene Übertragung von Computerdaten über längere Entfernungen benutzte diese Technologie, indem hierfür eine entsprechende Sprachverbindung geschaltet wurde und mit entsprechenden Modems eine Übertragung über diesen Sprachkanal stattfand.

Eine der ersten wichtigen Evolutionen dieser Netze war die Digitalisierung der Sprachübertragung in den Telefonnetzen außerhalb des Netzanschlussbereichs. So wurden Anfang der 70er-Jahre nach und nach die Vermittlungsstellen und Weitverkehrsnetze von analoger Multiplextechnik auf die neue, digitale Technologie umgerüstet. Diese Technologie wurde aufgrund ihrer hierarchischen und asynchronen Struktur PDH (Plesiochrone Digitale Hierarchie) genannt und ist der Vorläufer der digitalen Telefonnetze, wie wir sie heute kennen. Sie arbeitet vorerst noch mit den bisher verwendeten Koaxialkabeln, aber auch das sollte sich bald ändern.

Die Grundlagenentwicklung für den nächsten Meilenstein, die optische Übertragungstechnik, begann in verschiedenen Labors bereits schon in den 50er- und 60er-Jahren mit der Entwicklung von modulierbaren Lasern, die mit Informationen codiertes Licht aussenden konnten. Auf Halbleitern basierende Umsetzer auf der Empfängerseite wandelten diese Lichtimpulse wieder in elektrische Signale zur Weiterverarbeitung um. Diese und weitere Forschungen auf dem Gebiet der optischen Übertragungsmedien, vor allem der Entdeckung von Lichtwellenlängenbereichen, in denen sich das Licht extrem störungs- und verlustfrei durch Glasfaserleitungen ausbreiten kann, führte zum Einsatz der ersten optischen Netze im Jahr 1977 in den USA und England. Zur Jahrtausendwende waren bereits etliche 10-Gigabit/s-Systeme im Einsatz.

Neue Entwicklungen, wie WDM und DWDM (Dense Wavelength Division Multiplexing), machen es zusätzlich möglich, die Übertragungskapazität einer einzigen Faser bei gleicher Modulationsfrequenz noch weiter zu vervielfachen, indem man verschiedene Lichtfarben gleichzeitig hindurchschickt. In den Labors der großen und innovativen Telekommunikationsausrüster und etlicher Forschungseinrichtungen arbeiten mittlerweile die ersten Prototypen für den nächsten Schritt, die 1-Tbit/s-Technologie (= 1.000.000.000.000 Bit/s), bereits stabil.

Als in den 70er-Jahren immer mehr Daten über öffentliche Netze transportiert wurden, machten sich etliche Leute darüber Gedanken, dass die Technik der Leitungsvermittlung für die damaligen Datenmengen, Datentypen und Verkehrsprofile ziemlich ineffizient war. So begann man mit der Entwicklung einer speziell für die Übertragung von Daten geeigneten, paketorientierten Vermittlungstechnik. Mit dieser wird nicht mehr für eine bestimmte Zeit ein Übertragungskanal reserviert, egal ob nun Informationen übertragen werden oder nicht. Die zu übertragenden Daten werden stattdessen in kleine Teile, genannt Pakete, zerlegt, die dann der Reihe nach vom Sender zum Empfänger transportiert werden.

Die Verbindung zwischen Sender und Empfänger war nun also keine reservierte Leitung bzw. kein Kanal in der digitalen Multiplexhierarchie mehr, sondern sie bestand nur noch scheinbar (virtuell) und wurde deshalb auch als VC (Virtual Circuit) bezeichnet. Diese Technik wurde damals unter dem Namen X.25 bekannt, und einige Netze existieren sogar heute noch. Diese Art von Netzen ist bereits einer der Vorläufer heutiger VPNs, denn sie weisen ein dafür unabdingbares Merkmal auf: virtuelle Verbindungen.

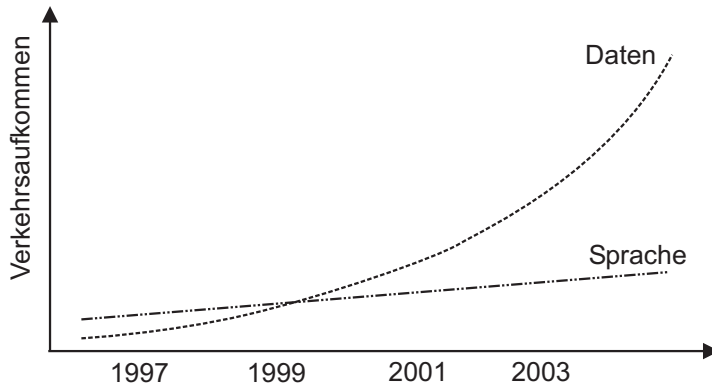


Abbildung 1.3: Im Jahr 1999 wurden erstmals in der Geschichte mehr Daten- als Sprachsignale über öffentliche Netze transportiert.

Aber auch hier entwickelte sich die Technik immer weiter. Da die Übertragungstechnik, über welche die X.25-Pakete transportiert wurden, von ihrer Qualität her gesehen immer besser wurde und sich ihre Geschwindigkeit gleichzeitig immer mehr erhöhte, wurde das X.25-Protokoll immer ineffizienter. Denn der komplexe Mechanismus zur Korrektur von Übertragungsfehlern, der X.25 unter anderem auszeichnet, war mittlerweile überflüssig, und die Zeit war reif für ein modernes, schlankes und schnelleres paketorientiertes Protokoll.

Möglich wurde dieses aber erst nach der Einführung von (Schmalband-)ISDN (ISDN, Integrated Services Digital Network) in den frühen 80er-Jahren. Diese digitale Technologie, die erstmals in der Geschichte der Telefonie digitale Ende-zu-Ende-Verbindungen über Telefonnetze möglich machte, sollte neben der digitalen, leitungsvermittelnden Sprachübertragung auch einen speziellen Dienst zur paketorientierten Vermittlung von Daten erhalten. 1988 veröffentlichte die ITU-T (ITU-T, International Telecommunication Union – Telecommunication Standardization Bureau) die erste, allerdings milde formulierte, rudimentäre Spezifikation (ITU-T I.122) zu dem als Frame Relay bezeichneten ISDN-Dienst. Frame Relay sollte dabei über einen ISDN-B-Kanal mit 64 KBit/s zur Verfügung gestellt werden.

Aber der eigentliche Durchbruch gelang der Frame-Relay-Technologie, als sich vier Telekommunikationshersteller (Cisco, Digital Equipment, Nortel Networks und Stratacom) zu einem losen Gremium zusammenschlossen und weitere, mehr an der Praxis orientierte Standardisierungsvorschläge erarbeiteten. Diese „Group of Four“ war der Grundstock des Frame Relay Forums (FRF), das maßgeblich die weitere Standardisierung vorantrieb, die dann auch von offiziellen Organisationen übernommen wurde. So wurde die Technik auch sehr schnell von dem Medium ISDN entkoppelt und als universelle Datenvermittlungstechnologie weiter entwickelt. Im Jahr 1991 wurde der erste Frame-Relay-Dienst in Betrieb genommen. Heute ist Frame Relay, vor allem im Bereich von Zugangsgeschwindigkeiten im Bereich bis 2 MBit/s, immer noch verbreitet und findet sich im Portfolio von fast jedem Carrier und Service Provider wieder. Auch Frame Relay ist vom Prinzip her eine VPN-Technologie, denn auch hier werden virtuelle Verbindungen durch ein öffentliches Netz geschaltet – und dies bereits mit einigen Funktionen zur Garantie bestimmter Dienstgüter.

Eine weitere wichtige Entwicklung im WAN-Bereich war der erste Schritt in Richtung Multiservice-Netze mit der Entwicklung und Einführung von ATM (Asynchronous Transfer Mode), einer statistischen Multiplextechnologie für hohe Geschwindigkeiten, die ab 1990 Einzug in die Weitverkehrsnetze von Carriern und Service Providern hielt. ATM wurde speziell zur Optimierung von Weitverkehrsnetzen entwickelt und diente dazu, verschiedene Verkehrskategorien (Sprache, Video, verschiedene Arten von Daten) effizient über darunter liegende statische Multiplexhierarchien wie SDH zu übertragen. Trotzdem sich ATM in lokalen Netzwerken, wo es in den meisten Fällen von den verschiedenen Highspeed-Ethernet-Varianten vertrieben wurde, nie richtig durchsetzen konnte, ist es im WAN-Bereich nach wie vor eine der dominierenden Multiservice-Technologien. Selbst viele andere Netztechnologien sind ATM überlagert, so zum Beispiel Frame Relay Backbones oder Multi Protocol Label Switching (MPLS). Auch Technologien wie ADSL benutzen ATM.

Anfang der 90er-Jahre hielt noch eine weitere Entwicklung einen relativ unspektakulären, aber nichtsdestotrotz maßgeblichen Einzug in die großen öffentlichen Netze: Die so genannte Digitale Synchrone Hierarchie (SDH) löste die sich in vielen Bereichen als ineffizient und etwas umständlich erweisende PDH als Multiplextechnik ab. Zusammen mit SONET (Synchronous Optical Network), das vor allem in Nordamerika eingesetzt wird, aber weitgehend kompatibel ist, hat sich SDH weltweit als die Standardtechnologie für Weitverkehrsinfrastrukturen der großen Carrier etabliert. SDH und SONET ermöglichen den Aufbau von leistungsfähigen, stabilen Hochgeschwindigkeitsnetzen mit mehr als 99,999% Verfügbarkeit und Umschaltzeiten von garantiert weniger als 50 ms im Fehlerfall. Auch die Bitfehler sind extrem niedrig, ebenso die Laufzeitverzögerungen. Diese Netze bilden nach wie vor die Basis für fast alles, was heute an Informationen, egal welcher Art, über weite Strecken transportiert wird. Alle anderen Protokolle wie ATM, Frame Relay, IP oder Leitungsvermittlung (Sprache) benutzen bis auf ganz wenige Ausnahmen diese Technologien.

Allerdings wurde über diese ganze Zeit hinweg ein Bereich der Weitverkehrsnetze sehr stiefmütterlich behandelt: der Zugangsbereich, die so genannte „Last Mile“ zwischen dem Endteilnehmer (hier im Sinne von Haushalten oder kleinen Büros) und der Vermittlungsstelle. Denn in diesem Umfeld hatte sich bis Mitte der 90er-Jahre recht wenig verändert. Über die üblicherweise vorhandene Zweidraht-Kupferleitung zur nächstgelegenen Ortsvermittlungsstelle waren seit eh und je nur maximal etwa 56 KBit/s (mit Modemtechnologie) oder 128 KBit/s im Fall von ISDN (mit Kanalbündelung, sonst 64 KBit/s) möglich. Die Diskrepanz zwischen der Entwicklung der Bandbreite im Backbone- und der im Access-Bereich wurde also immer größer.

Aber durch politische Einflüsse, insbesondere die der in den meisten Ländern erfolgten Liberalisierung der TK-Märkte, wurde der Boden für neue, schnelle und vor allem vom Telefondienst entkoppelte Zugangsdienste bereitet. So wurden die heute aktuellen Highspeed-Zugangstechnologien wie DSL (Digital Subscriber Line), Breitbandkabelmodem, Satellitenverbindungen usw. entwickelt, die den Benutzern jetzt Geschwindigkeiten bis in Multi-Megabit/s-Bereich ermöglichen – bei gleichzeitiger Weiterbenutzung der bisherigen analogen oder digitalen Telefonie.

Man sieht also, die Entwicklungen der letzten 30 Jahre im Bereich der Übertragungstechnologie hatten es in sich und haben damit auch den auf ihnen basierenden Technologien deutliche Impulse geben. Aber man kann auch gleichzeitig erkennen, dass diese Entwicklungen immer noch weitergehen, insbesondere im Bereich der mobilen Kommunikation ist noch einiges zu erwarten, ebenso dürfte es im Bereich der Zugangstechnologien auch noch einigen Spielraum für Innovationen geben. Man darf also sehr gespannt sein, was die nächsten Jahre noch an Neuerungen und Verbesserungen bringen werden – ganz zu schweigen von den nächsten 30 Jahren.

Datennetze

Moderne Datennetzwerke beruhen, wie im vorigen Kapitel beschrieben, auf dem Prinzip des Zerlegens von Informationen in kleine Stücke und deren Übertragung über geeignete Netze. Die Verarbeitung erfolgt dabei hauptsächlich auf Schichten 1 und 2 des ISO OSIRM (OSIRM, Open Systems Interconnect Reference Model, s.u.). Nachdem wir nun gesehen haben, wie die Übertragungstechnologie in diesem Bereich in den letzten Jahren fortgeschritten ist, betrachten wir uns einmal die Entwicklung der Technologien auf den beiden nächsthöheren Schichten, auf denen hauptsächlich Endsysteme miteinander kommunizieren, die Netzwerk- und die Transportschicht.

Die ersten Verbindungen von Rechnern erfolgten grundsätzlich auf zwei gegensätzliche Arten, entweder herstellerspezifisch, also nur mit Produkten eines Herstellers bzw. von ihm genehmigten Lizenzprodukten, oder als offene und frei verfügbare Technologie, die den Aufbau von heterogenen Netzen ermöglicht.

Herstellerspezifische Netzwerktechnologien, die bekanntesten sind wohl die Produkte SNA von IBM, DECNet von Digital Equipment und IPX/SPX von Novell, wurden von diesen Unternehmen entwickelt, um deren spezifische Computer und Applikationen miteinander zu vernetzen. Daraus resultierte in den meisten Fällen eine starke Affinität der Netzwerktechnologie zu den Rechner- oder Softwareprodukten des jeweiligen Anbieters. SNA zum Beispiel wurde ursprünglich entwickelt, um ein für IBM-Hostarchitekturen maßgeschneidertes Netzwerk anbieten zu können, mit dem man Großrechner, Steuereinheiten und daran angeschlossene Drucker verbinden kann. Natürlich hat sich diese an eine bestimmte Architektur angepasste Technologie trotz etlicher Versuche einer Nachbesserung als ziemlich ungeeignet für andere Umgebungen, zum Beispiel PC-Ver-netzung oder heterogene Netze, erwiesen.

Das Gleiche gilt auch in verschiedenem Maß für alle anderen herstellerspezifischen Entwicklungen, auch wenn sie teilweise schon viel offener angelegt waren. So zum Beispiel auch IPX/SPX, das zwar in den Anfängen der PC-Netzwerke im lokalen Bereich eine ziemliche Dominanz erlangen konnte, aber, aufgrund seiner Schwächen im zunehmend an Bedeutung gelangenden WAN-Bereich, immer mehr von IP verdrängt wurde. Derartige Beispiele gibt es genug, allen ist dabei eigen, dass sie aufgrund ihrer Abhängigkeit von einem bestimmten Hersteller und ihrer Ausrichtung auf dessen Produkte niemals richtig zukunftsfähig waren und damit nach und nach, bis auf ein paar kleine Nischen, den Technologien für offene, heterogene Netze weichen mussten.

Offene Netze auf der anderen Seite entstanden hauptsächlich durch die Bemühungen von zwei Organisationen, der ISO (ISO, International Standardization Organisation) mit ihrem OSIRM und den darauf beruhenden Protokollen sowie den im Laufe der Zeit unterschiedlichen, zuständigen Dachorganisationen, welche die Entwicklung des Internet-Protokolls (IP) und aller damit verbundenen Protokolle koordinierten. Heute ist maßgeblich die IETF (Internet Engineering Task Force) mit ihren verschiedenen Arbeitsgruppen für die Standardisierung im IP- und Internet-Umfeld verantwortlich.

Leider ist das OSIRM geradezu ein Paradebeispiel für eine verpasste Chance. Und das, obwohl dieses Modell eigentlich richtungsweisend ist und sogar oft für andere Architekturen, die damit nicht das Geringste zu tun haben, z.B. das TCP/IP-Modell, fälschlicherweise als Referenz herangezogen wird. Aber das Problem beim OSI-Referenzmodell und den entsprechenden Protokollen war einfach die ISO selbst bzw. die Tatsache, dass sich eine derartige große und damit auch in mancher Hinsicht schwerfällige Organisation der Sache angenommen hatte: Es gab zu viele Interessengruppen, es sollten zu viele Features in die erste Version, es wurde zu viel Politik gemacht. Fazit: Das Ganze dauerte einfach zu lange. Nicht zuletzt deshalb schwenkten viele schließlich zu einem anderen Modell um, das zwar längst nicht alles konnte, auch nicht so sauber zwischen den verschiedenen Schichten trennte, das aber verfügbar war, funktionierte und sich deshalb rasant verbreitete: das eher pragmatisch angelegte TCP/IP-Modell und die ganze Flut der dafür spezifizierten Protokolle.

Man weiß nicht genau, wie viel Wahrheit darin steckt, aber Insider bezeichnen die frühere Sowjetunion als den ursächlichen Auslöser für die Entwicklung des weltgrößten Datennetzwerks, des Internets. Denn als die damalige Sowjetunion im Jahr 1957 eine gerade mal fußballgroße, dröge vor sich hin piepsende Metallkugel, genannt „Sputnik 1“, in eine Erdumlaufbahn brachte, löste dies bei den USA einen ziemlichen Schock und in der Folge nicht unbeträchtliche Entwicklungen aus. Was daraus entstand, ist weitgehend bekannt: Im Bereich der Raumfahrttechnologie holten die USA massiv auf, gipfelnd in der Landung eines Menschen auf dem Mond. Im militärischen Bereich, und der war und ist der Hauptantrieb für die Fortschritte im Bereich der Raumfahrt und vieler anderer Technologien, wurde man angesichts des Vorsprungs der Sowjets und der Tatsache, dass diese ebenfalls über Nuklearwaffen verfügten, ziemlich paranoid.

Und eine von vielen Maßnahmen, die in der damaligen Zeit im Klima des Kalten Krieges getroffen wurden, war die Einrichtung der Advanced Research Projects Agency, kurz ARPA, im Jahr 1959. Ziel der ARPA war es, neue Technologien entwickeln zu lassen, vielversprechende Ideen zu sammeln und nach Visionen und Visionären zu suchen – es war damals fast nichts verrückt genug. Die meisten der Projekte wurden in Zusammenarbeit mit Hochschulen und von verschiedenen Industrieunternehmen durchgeführt, so dass viele Entwicklungen zwar von der ARPA angestoßen, koordiniert und teilweise finanziert wurden, die Forschung selbst jedoch weitgehend in zivilen Einrichtungen erfolgte.

Und eines der Projekte der ARPA war es nun, eine Art selbstheilendes und damit ausfallsicheres Netzwerk zu entwickeln. Ein Netz, das auch beim Ausfall (damals gemeint: nuklearer Volltreffer) eines oder mehrerer Netzknoten sicherstellte, dass noch Daten über andere funktionsfähige Knoten zu ihrem Bestimmungsort transportiert werden können. Bereits zu dieser Zeit, es war Anfang der 60er-Jahre, wurde das Konzept einer paketorientierten Vermittlung („Packet Switching“) beschrieben. Allerdings gab es auch

im zivilen Bereich ähnliche Entwicklungen, zum Beispiel vom Massachusetts Institute of Technology (MIT) oder der RAND Corporation, die ebenfalls einen Einfluss auf das ARPA-Projekt eines unzerstörbaren Netzes hatten.

Ein erster Versuch verband im Jahr 1965 zwei Rechner an der Ost- und Westküste miteinander, aber ein echtes Datennetzwerk wurde erst daraus, als am 10. Dezember 1969 vier Universitätsrechner miteinander verbunden wurden: Die University of California Los Angeles, das Stanford Research Institute, die University of Utah und die University of California Santa Barbara nahmen das erste, echte paketvermittelnde Netzwerk der Welt, das nach seinen Wurzeln benannte ARPANET, in Betrieb. Und schon damals war es ein heterogenes Netz, denn alle vier Universitäten betrieben unterschiedliche Rechner. Das war die Geburtsstunde des Internets, auch wenn ein anderes Unternehmen, das zu dieser Zeit noch gar nicht existierte, die „Erfindung“ des Internets für sich in Anspruch nimmt.

Im Jahr 1972 waren bereits knapp 40 Rechner an das Netzwerk angeschlossen, es wurde das erste E-Mail-Programm geschrieben, und die ARPA bekannte offiziell Farbe und nannte sich fortan passender DARPA (Defense ARPA). Bereits ein Jahr später wurde das Internet international erweitert, als erste Länder schlossen sich Großbritannien und Norwegen an. Im Jahr 1983 erfolgte dann die längst überfällige Trennung des ARPANET in einen amerikanischen militärischen Teil (MILNET) und einen internationalen zivilen Teil (Internet). Zu dieser Zeit wurde auch die IP-Protokollfamilie zum bestimmenden Standard des Internets.

Im Jahr 1989, das Internet bestand mittlerweile aus über 10.000 Teilnehmern, stellte die ARPA ihre Aktivitäten ein, und das eigentliche ARPANET wurde schließlich 1990 offiziell außer Betrieb genommen. Was blieb, war das damals fast ausschließlich von wissenschaftlichen Einrichtungen benutzte Internet.

Bedingt durch neue Möglichkeiten, die sich durch die Entwicklungen im Bereich der Transporttechnologie ergaben, erlebte das Internet eine von immer höher werdenden Anforderungen an Bandbreiten getriebene, rasante Weiterentwicklung. Der ersten Verbindungen begnügten sich noch mit 56 KBit/s, erst im Jahr 1980 erweiterte man diese Verbindungen auf 1,5 MBit/s und zehn Jahre später, 1990, auf die ersten 45 MBit/s-Leitungen. Im Jahr 1994 fanden die ersten, schnellen 155-MBit/s-Glasfaserverbindungen auf Basis digitaler Multiplextechnologie (SONET, Synchronous Optical Network) Verwendung, und heutzutage hantiert man im Internet-Backbone mit 10-GBit/s-Verbindungen. Auch die geografische Ausdehnung des Internets ist imposant, man betrachte sich nur einmal die interkontinentalen Verbindungen. Allerdings ist auch das Internet ein Spiegelbild des sozialen und wirtschaftlichen Gefälles auf unserem Planeten, denn auch das geht ziemlich eindeutig aus der geografischen Bandbreitenverteilung hervor.

Und all die anderen Netzwerkprotokolle? Sie haben einfach nicht überlebt oder fristen noch ein tristes Nischendasein, bis auch sie von IP abgelöst werden. Selbst wenn sie in bestimmten Bereichen, z.B. IPX in lokalen Netzen, oder für bestimmte Systemumgebungen, für die sie speziell entwickelt wurden, z.B. SNA, durchaus Vorteile gegenüber IP aufweisen können, es führt einfach kein Weg an IP mehr vorbei. Und das nicht zuletzt auch aus wirtschaftlichen Gründen, denn anstelle eines ganzen Wusts unterschiedlicher Verfahren nur ein einziges Protokoll in seine Systeme implementieren zu müssen, bedeutet effizientere und schnellere Entwicklung und damit auch weniger Kosten für die Hersteller.

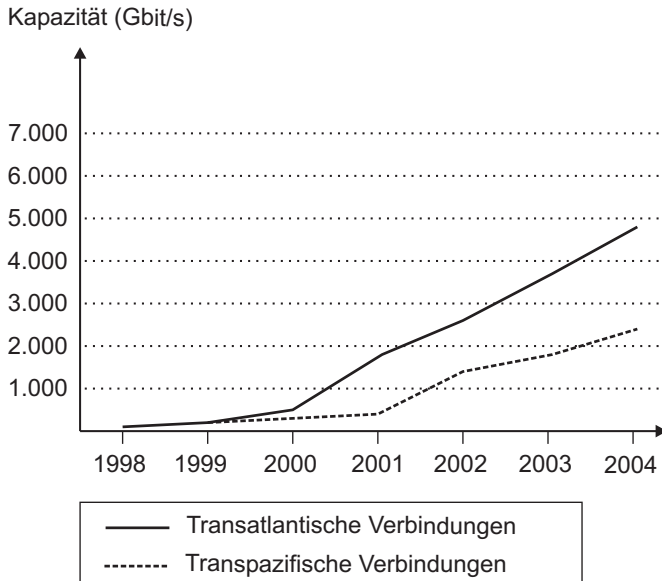


Abbildung 1.4: Der Datenverkehr im Internet steigt kontinuierlich, ein Ende ist nicht absehbar.

Im Jahr 1995 erblickte ein neuer Standard für ein neues IP-Protokoll das Licht der Welt, der RFC 1883, IP Version 6 (IPv6), der das „alte“ IP-Protokoll Version 4 (IPv4) ablösen sollte. Der Grund war hauptsächlich ein sehr viel größerer Adressierungsbereich und eine höhere Flexibilität des Protokolls. Aber noch etwas anderes wurde eher beiläufig in das Protokoll aufgenommen: zwei hochwertige Sicherheitsfunktionen auf IP-Ebene namens AH (Authentication Header) und ESP (Encapsulating Security Payload) und eine einfache, effiziente und robuste IP-in-IP-Tunneling-Methode. Und da wurden einige Leute sehr hellhörig und machten sich, sehr weitsichtig, wie man heute weiß, daran, die Sicherheits- und Tunneling-Funktionen von IPv6 auch in das alte IP-Protokoll aufzunehmen.

IP Version 6 hat, wie wir alle wissen, bis heute extreme Anlaufschwierigkeiten. Aber die IP-Sicherheitsfunktionen (IPSec, IP Security), die im Jahr 1998 in mehreren RFCs von Mitarbeitern von Northern Telecom (Nortel), Bay Networks, Cisco Systems, dem NIST (National Institute for Standards and Technology) und der NSA (National Security Agency) sowohl für IPv6 und IPv4 geschrieben wurden, haben das Gesicht heutiger Netze nachhaltig verändert. Denn mit IPSec konnte man sehr sicher Daten privater Netze über das Internet transportieren – auch mit privaten oder nicht registrierten, überlappenden Adressen. Damit wurden die IP-VPN, wie wir sie heute kennen, überhaupt erst möglich gemacht.

1.2.3 Netzwerkanwendungen

Hier haben sich, sowohl beeinflusst von der Netzwerk- und Übertragungstechnologie als auch selbst auf diese einflussnehmend, massive Entwicklungen ergeben. Ein weiterer von Netzwerken unabhängiger Faktor, der aber für die Netzwerkentwicklung extrem wichtig ist, ist der Siegeszug der Mikroprozessoren, der verteilten Systeme und der Per-

sonalcomputer. Als im Jahr 1974 die ersten vollwertigen 8-Bit-CPU's vorgestellt wurden, Unternehmen wie Apple die ersten Mikrocomputer entwickelten oder IBM 1982 seinen PC auf den Markt brachte, arbeiteten diese Systeme noch isoliert, das heißt ohne Verbindung zu anderen Computern. Daten wurden in der Regel per Diskette übertragen. Aber durch die einsetzende Flut von Mikrocomputern, die über die Unternehmen hereinbrach, wurde der Bedarf an Dingen wie einfacher Online-Datenaustausch, Datei- oder Druckdienste zunehmend größer, und die ersten lokalen PC-Netzwerke, insbesondere auf Basis von Novell Netware oder für größere Netze von Banyan, wurden installiert. Auch UNIX erwies sich als sehr kommunikationsfreudig, zuerst mit eigenen Verfahren, dann mehr und mehr mit dem IP-Protokoll.

Mit der Zeit, im LAN setzte sich gleichzeitig immer mehr das robuste und kostengünstige Ethernet gegen Token-Ring und Arnet durch, wurden die verschiedenen LAN-Inseln miteinander verbunden, und es entstanden richtige Unternehmensnetze, die auch zunehmend über WAN-Verbindungen verfügten. Das alles hat auch die Applikationslandschaft nachhaltig verändert. Einfache Datei- oder Druckservices reichten nicht mehr aus, und Informationssysteme, E-Mail, Groupware, verteilte Datenbanken, verteilte Applikationen und alle anderen Arten von netzbasierten Anwendungen fanden ihren Weg in die Unternehmen und übten ihrerseits wiederum einen starken Druck auf die Weiterentwicklung der Netzwerke aus. Manche Visionen, wie zum Beispiel das papierlose Büro (niemals wurde mehr Papier verbraucht als nach der Erfindung des PC), erfüllten sich nicht, andere hingegen wurden von der Realität sogar noch überholt.

Auch in der Philosophie hinsichtlich der Benutzung des Internets hat sich in den letzten zehn Jahren ein stetiger Wandel vollzogen. In den Anfängen waren hauptsächlich Universitäten sowie Forschungs- und Entwicklungsabteilungen verschiedener Unternehmen an das Internet angeschlossen. Es war zunächst ein reines Wissenschaftsnetz, dessen Betriebskosten auch hauptsächlich von diesen Einrichtungen getragen wurden. Für die Benutzer war es scheinbar, zumindest aus deren persönlicher Erfahrung, kostenlos. Auch die folgende Kommerzialisierung begann eher schleichend auf einer anfänglichen „For Free“-Basis. Verschiedene Unternehmen stellten über das Netz kostenlose Software-Upgrades zur Verfügung oder richteten Anwenderforen und Ähnliches ein.

Der E-Mail-Verkehr, anfangs rein textbasierend, kam ebenfalls langsam in Schwung. Nicht wenige Unternehmen nahmen das Internet damals auch nicht recht ernst, so glaubten Unternehmen wie zum Beispiel Microsoft und einige Online-Anbieter allen Ernstes, ein eigenes Internet-ähnliches Netzwerk, natürlich auf rein proprietärer Basis, aufbauen zu können. Was daraus wurde, ist bekannt, denn mittlerweile benutzen alle das Internet oder sind de facto Internet Service Provider (ISP) geworden, wenn auch mit unterschiedlichen Zielsetzungen, Inhaltsangeboten, Infrastrukturen und einem teilweise fragwürdigen Umgang mit Kundeninformationen.

Aber erst mit der Webtechnologie, dem grafischen Internetdienst „World Wide Web“ (WWW), im Jahr 1989 am europäischen Kernforschungszentrum CERN in Genf entwickelt, und allem, was sich daraus an weiteren Diensten und Applikationen entwickelte, wurde das Internet zu dem, was es heute ist.

Einen weiteren, enorm wichtigen Kick gab dem Internet eine Erweiterung der Firma Netscape im Jahr 1994, die ihren Navigator, damals ein sehr verbreiteter Browser, um Funktionen für sichere Webtransaktionen erweitern sollte. Netscape nannte diese Tech-

nologie damals SSL, Secure Socket Layer. Auf diesen Entwicklungen baut heute eine relativ neue Technologie für Remote Access und Extranets auf, die SSL-VPN.

Weitaus interessanter, vor allem auch hinsichtlich dieses Buches, ist das sich ändernde Benutzungsprofil des Internets. Statt wie in den Anfängen kostenfrei für Forschungszwecke, wird das Netz immer mehr selbst zum Geschäft. Und zur Geschäftsgrundlage. Denn viele Unternehmen basieren bereits vollständig auf dem Internet, andere verlagern zunehmend Teile ihres Geschäftes auf das Web.

Eine weiterer bemerkenswerter Punkt ist die Tatsache, dass immer mehr Unternehmen dazu übergehen, das Internet auch als Basis für ihre privaten Unternehmensnetze zu benutzen. Virtuelle private Netzwerke (VPN) benutzen dieses weltweit größte, öffentliche Weitverkehrsnetz für ihre private, vertrauliche und teilweise geschäftskritische Kommunikation. Somit wird das Internet zum ernsthaften Konkurrenten von Diensten, die auf traditionellen Weitverkehrsnetzen basieren, vor allem da VPNs deutlich billiger und flexibler zu betreiben sind als herkömmlich aufgebaute WANs.

Anteil von mobilen
Datendiensten im Internet

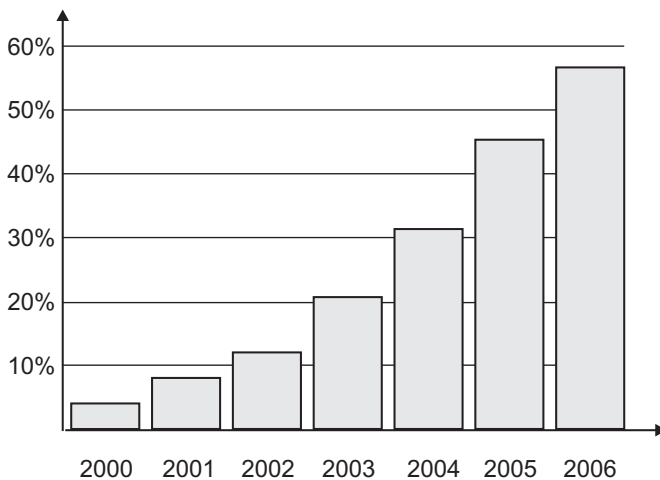


Abbildung 1.5: Der Anteil mobiler Datendienste im Internet ist in den letzten Jahren rasant angestiegen.

Das Schlagwort „Mobile Internet“ ist im Geschäftsleben auch zu Realität herangewachsen, teils durch die VPN-Technologie, die als öffentliche Netze natürlich nicht nur Fest-, sondern natürlich auch Mobilnetze benutzen kann, teils durch spezielle Internetanwendungen, die auf die niedrige Auflösung der kleinen Displays von Mobilteilen oder PDAs und die noch etwas holprige und langsame Datenübertragung über Mobilnetze optimiert sind. Hierin liegt ein hohes Potenzial, vorausgesetzt man bekommt die Technik in den Griff und die Preise sinken auf ein wettbewerbsfähiges Niveau.

Man sieht also, der Wandel, der sich im Umfeld der Weitverkehrsnetze vollzogen hat, ist schon sehr drastisch, vor allem angesichts dessen, was heute von allen möglichen Carriern und Service Providern angeboten wird. Was natürlich auf den ersten Blick ins Auge sticht, sind die höheren Bandbreiten bei gleichzeitig deutlich niedrigeren Preisen. Aber

es gibt noch viel mehr interessante Entwicklungen zu vermerken, die Konvergenz von Sprache und Daten oder LAN- und MAN-Technologien (MAN, Metropolitan Area Network), um nur zwei Beispiele aufzuführen.

So ist zu beobachten, wie in den Anfängen hauptsächlich Technologien für Sprachübertragung (Leitungsvermittlung) zum Transport für Datenpakete verwendet wurden. In naher Zukunft wird jedoch erwartet, der Trend ist ganz eindeutig erkennbar, dass Sprache zunehmend über reine Datennetze (Paketvermittlung) übertragen wird – Sprache nur noch einer von vielen verschiedenen Datentypen. So ist zum Beispiel die Festnetzkomponente von UMTS (vgl. Kapitel 10) in seinem Endausbau als reines Datennetzwerk auf Basis des IP-Protokolls Version 6 geplant.

Auch die bisherige Abgrenzung zwischen klassischen LAN-Technologien auf der einen und typischen WAN-Technologien auf der anderen Seite verschwimmt immer mehr, seit man in Metropolitan Area Networks zunehmend Ethernet einsetzt – die LAN-Technologie schlechthin.

1.3 Warum VPN?

Wenn man sich einmal heutige VPNs anschaut und sie mit dem vergleicht, was vor sechs, sieben Jahren eingesetzt wurde, wird man sich schnell bewusst, wie sich auch in diesem Bereich vieles verändert hat. Der erste Gedanke, dies alles nur mit gestiegenem Bandbreitenbedarf und Kostengründen zu erklären, ist jedoch nur ein kleiner Teil der Wahrheit. Denn es ist eine ganze Reihe von Faktoren, sowohl technologische als auch wirtschaftliche, betriebliche und soziale, die den Wandel zu VPN heutiger Ausprägung beeinflusst haben. Und dies sollte man sich vor allem dann klar machen, wenn man sein Netzwerk auch für die Zukunft planen will.

Insbesondere der ausschließliche Blick durch die Technologiebrille kann dabei schnell die Sicht auf das große Ganze verstellen, denn die Fortentwicklung von VPN ist von wesentlich mehr und verschiedenartigeren Faktoren abhängig als die Entwicklung im Bereich lokaler Netze. So sind neben der natürlich auch mitbestimmenden technologischen Entwicklung vor allem Änderungen in der Art und Weise, wie Unternehmen heute arbeiten, als auch im gesellschaftlichen, politischen und sozialen Umfeld ausschlaggebend.

1.3.1 Veränderung der Geschäftsprozesse

Nicht wenige Unternehmen haben die Art und Weise wie sie ihre Geschäfte betreiben, an die veränderte Kommunikationslandschaft angepasst. Neben dem Offensichtlichen, nämlich der Nutzung des Internets zum Zweck der Werbung und Verbreitung von Informationen, sowie dem starken Wachstum des B2C-Geschäftes (B2C, Business to Consumer), haben sich aber auch andere Dinge in den Unternehmen verändert. So ist heutzutage fast kein Rechner, der irgendwo in einem Unternehmen installiert wird, mehr ohne eine Netzwerkverbindung. Die Netze in den verschiedenen Standorten sind ebenfalls miteinander verbunden und damit auch alle Rechner und deren Benutzer im Unternehmen. Hinzu kommen Verbindungen ins Internet oder zu anderen, externen Personen oder anderen Unternehmen. Damit ist die Basis für B2B-Kommunikation geschaffen,

etwas, das etliche Unternehmen, wenn oft auch gar nicht unter diesem Modebegriff, schon längere Zeit erfolgreich praktizieren. Das alles hat entscheidenden Einfluss darauf, wie heute in vielen Firmen gearbeitet wird.

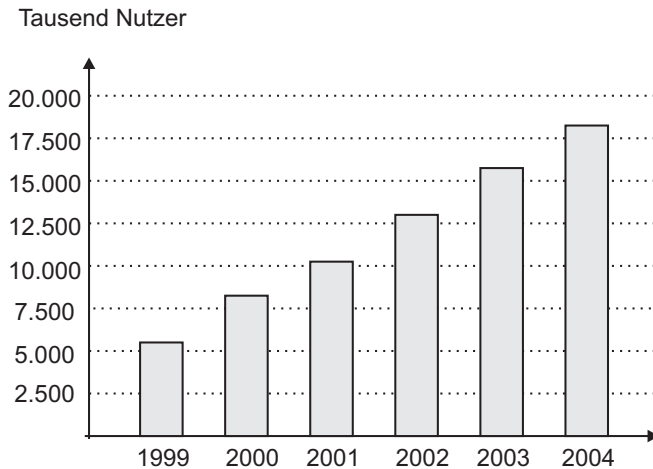


Abbildung 1.6: Die Zahl geschäftlicher Internetbenutzer ist in den letzten Jahren stark gewachsen.

Aber nicht nur das Wie, sondern auch das Was hat sich geändert, denn aufgrund der heutigen Infrastrukturen können Unternehmen auf eine Art ihr Geschäft betreiben, die noch vor zehn Jahren fast undenkbar war. Reine Online-Händler oder auch traditionelle Unternehmen, die ihr bisheriges Ladengeschäft um die Web-Komponente erweitert haben, bieten rund um die Uhr ihre Produkte und Dienstleistungen an. Darin liegt natürlich für etliche Unternehmen ein großes Ratiopotenzial. Denn man kann zum Beispiel auf der einen Seite das, auch nach seiner so genannten Reform immer noch verstaubte, deutsche Ladenschlussgesetz umgehen und trotzdem gleichzeitig die hierzulande recht hohen Personalkosten einsparen. Denn seine Information und Beratung besorgt der Kunde in der Regel selbst, ebenso die Abwicklung des Kaufs.

Aber auch die Kommunikation zwischen unterschiedlichen Unternehmen hat sich stark verändert. Während anfangs ein eher unverbindlicher Informationsaustausch erfolgte, sind heutzutage nicht wenige Geschäftsprozesse transparent über mehrere verbundene Unternehmen verteilt. Den Anfang machten die Banken schon vor längerer Zeit mit der Einführung des elektronischen Inter-Bank-Zahlungsverkehrs. Heute sind Extranets, also Netze zwischen verschiedenen Unternehmen, schon Stand der Technik. Als Beispiel seien hier die großen Verbundnetze für Banken, Reisebüros oder auch das riesige Super-VPN ANX (Automotive Network Exchange) im Automobilbereich genannt.

Das alles sind sowohl Folgen der modernen Vernetzung als auch gleichzeitig ihr Antriebsmotor. Denn wenn sich erst einmal ein System wie ANX prinzipiell durchgesetzt hat, dann werden sich immer mehr Unternehmen anschließen, es werden immer mehr Informationen übertragen und dadurch auch die Anforderungen an Bandbreite, Verfügbarkeit und Qualität des Netzwerks immer höher.

Viele Unternehmen, die großen schon lange, der Mittelstand mittlerweile auch mehr und mehr, stehen vor neuen Herausforderungen durch die Globalisierung und die damit verbundene unausweichliche Dezentralisierung ihrer Firmen und ihrer Geschäftsabläufe. Obgleich das Wort Globalisierung gerade in der jetzigen Zeit zu einem Schlagwort geworden und je nach Verwendung nicht nur positiv besetzt ist, beschreibt es die Situation immer noch am besten. Es gibt nur noch ganz wenige, im produzierenden Gewerbe fast gar keine Unternehmen mehr, die ausschließlich nur in einem Land tätig sind, nur für den Markt dieses Landes produzieren und nur Geschäftsbeziehungen mit Unternehmen und Kunden in eben diesem Land pflegen.

Das ist mittlerweile die große Ausnahme. Egal ob man ausländische Märkte bedient, ob man mit ausländischen Zulieferern zusammenarbeitet oder ob man aufgrund exorbitanter heimischer Personalkosten und steuerlicher Vorteile im Ausland produziert, man ist fast immer gezwungen, im Ausland Standorte oder Büros oder zumindest Personen zu haben, die einen Zugriff auf das Unternehmensnetz benötigen. Da dieser Zugriff ein komfortables Arbeiten und zeitnahe Informationsverfügbarkeit ermöglichen soll, werden bestimmte Anforderungen hinsichtlich Bandbreiten, Übertragungsqualität und vor allem auch Sicherheit gestellt, deren Befriedigung nicht unerhebliche Kosten erzeugen kann.

Veränderung der Wettbewerbssituation

Eine Folge der Globalisierung und der schnellen, weltweiten Verfügbarkeit von Informationen ist auch die veränderte Wettbewerbssituation, in der sich viele Unternehmen wiederfinden. Denn in vielen Branchen wurde früher auf dem lokalen, nationalen Markt für eben diesen produziert. Der Wettbewerb war damit überschaubar, und die Konkurrenz produzierte vor allem auch unter den gleichen wirtschaftlichen und sozialen Bedingungen. Heutzutage sieht das anders aus, kaum eine Firma, die nicht für ausländische, internationale Märkte produziert und damit nicht in einer völlig neuen Wettbewerbssituation steckt. Dieser Wettbewerb zwingt viele Firmen sowohl zu einer globaleren Präsenz als auch zu einer höheren Effizienz, um auch gegen Mitbewerber aus Ländern mit anderen sozialen und gesellschaftlichen Randbedingungen konkurrenzfähig zu bleiben.

1.3.4 Mobilität und Flexibilität

Mobilität und Flexibilität sind heutzutage keine bloßen schicken Schlagwörter mehr, sondern knallharte Anforderungen in vielen Situationen des modernen Geschäftslebens.

Im Zuge immer höherer Mobilität sind insbesondere drahtlose Kommunikationsformen gefragt, sowohl im Bereich mobiler Sprachkommunikation als auch mehr und mehr drahtlose Datenkommunikation. Der perfekt gestylte Yuppie, der abends auf der Fahrt zu seiner In-Disco schnell mit seinem Multifunktions-Mobilgerät in seiner Firma anruft und einen Millionen-Deal abschließt, um dann noch schnell ein paar Aktien an der New Yorker Börse zu verkaufen, ist natürlich eine Werbefiktion – allerdings nur in dieser überzogenen Art der Darstellung. Denn die dafür erforderliche Technik ist da und wird auch genutzt, zwar nicht so sehr von unserem Yuppie aus der Werbung, aber beispielsweise von einem Vertriebsingenieur, der beim Kunden dringend Informationen braucht

oder für den Servicetechniker im Bereitschaftsdienst, der sich von zu Hause aus bei einem Kunden in dessen Systeme einwählen kann, ohne deshalb seine Familie alleine lassen zu müssen.

Damit Mobilität aber auch gleichzeitig Produktivität gewährleisten kann, muss ein zuverlässiger und vor allem sicherer Zugriff mobiler Mitarbeiter auf die von ihnen benötigten Ressourcen möglich sein, egal ob von zu Hause, auf dem Flughafen oder beim Kunden.

1.3.5 Kostenoptimierung

Die in den letzten Jahren, vor allem in einigen europäischen Industrienationen, immer kritischer werdenden Standortkosten zwingen fast alle Unternehmen zu teilweise drastischen Maßnahmen, um ihre Betriebskosten zu senken. Es ist mittlerweile praktisch kein Unternehmensbereich mehr unantastbar, und in diesem Zuge werden auch die IT-Kosten immer kritischer hinterfragt. Zwangsläufig stößt man dabei, wenn man die Personalkosten vorerst einmal außer Acht lässt, bei vielen Unternehmen mit mehreren Standorten, vielleicht sogar noch international verteilt, auf einen stattlichen Posten: die Betriebskosten für das Weitverkehrsnetz, vor allem bestehend aus den laufenden Kosten für Leitungsgebühren, Verbindungszeiten oder übertragenen Datenvolumina. Hier versucht man, Ratiopotenzial freizusetzen.

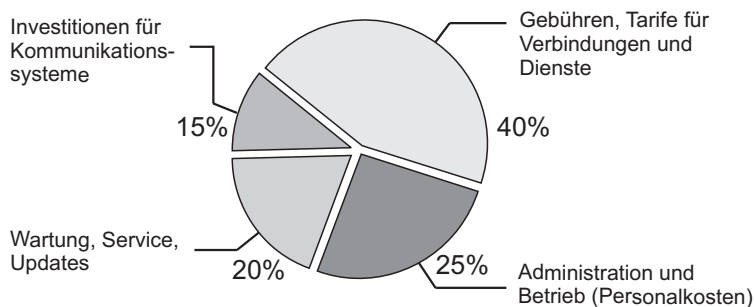


Abbildung 1.7: Das Modell Total Cost of Ownership betrachtet die Gesamtkosten im Gegensatz zu den reinen Anschaffungskosten.

Der in diesem Zusammenhang mittlerweile arg strapazierte Begriff der Total Cost of Ownership (TCO) hat trotz alldem bis heute nichts von seiner Aussagekraft verloren. Denn auch heute noch belegen entsprechende Studien, dass das Zur-Verfügung-Stellen notwendiger IT-Applikationen leicht € 10.000 pro Jahr und Benutzer übersteigen kann. Das Interessante an dieser Zahl ist jedoch weniger deren absolute Höhe als vielmehr die Tatsache, dass davon weniger als 15% für Hardware (Computer, Drucker usw.) anfallen und die restlichen 85% für Dinge wie Bereitstellung von Netzwerken und Kommunikationsinfrastruktur, Personalkosten usw., mit anderen Worten für die Betriebskosten aufgewendet werden.

Dies ist zunehmend für weitsichtig agierende Unternehmen auch der Grund, die geplanten Investitionen weniger darauf hin zu prüfen, ob sie in besonders günstige Anschaffungen münden, sondern vielmehr ob sie einen günstigeren Betrieb ermöglichen. Angesichts des Verhältnisses von 15% zu 85% ist es leicht nachvollziehbar, dass sich eine um

100% teurere Investition, wenn sie auch nur 10% Betriebskosten spart, in sehr kurzer Zeit amortisiert hat! Leider ist diese Weitsicht jedoch längst nicht in jedem Unternehmen vorhanden. Vielerorts verwenden die Einkäufer noch sehr viel Zeit und Mühe darauf, die letzten zwei Prozent bei den Anschaffungskosten herunterzuhandeln, was auf den ersten Blick natürlich nach einer Einsparung aussieht – nicht selten aber um den Preis deutlich erhöhter Betriebskosten!

Die Idee, die Gesamtkosten durch Senkungen im größten Anteil, den Betriebskosten, zu reduzieren, hat auch einigen Einfluss auf Weitverkehrsnetze, denn hier fallen nicht unbeachtliche Kosten an. Ein typisches WAN ist geradezu ein Paradebeispiel für den Einsatz des TCO-Modells, denn die Kosten für beispielsweise zehn WAN-Router eines kleinen Netzwerkes verschwinden angesichts der laufenden Kosten für die neun mindestens benötigten internationalen Festverbindungen, hier sind die Betriebskosten in der Regel sogar noch deutlich höher als 85% der Gesamtkosten. Hier bieten VPN ein sehr hohes Ratiopotenzial.

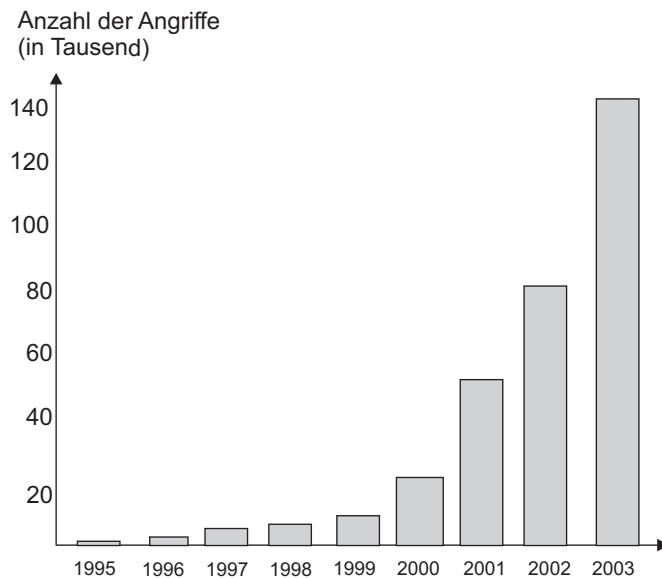


Abbildung 1.8: Seit dem Jahr 1999 ist die Anzahl der Angriffe aus dem Internet explosionsartig angestiegen.

1.3.6 Sicherheit

Wenn man verschiedene Standorte verbindet, Fernzugriff auf sein Unternehmensnetz erlaubt oder eine Verbindung zum Internet betreibt, dann verbindet man auf die eine oder andere Weise sein privates Netzwerk mit einem öffentlichen Netzwerk. Der Grad der Öffentlichkeit ist stark unterschiedlich, bei der Benutzung von digitalen Standardfestverbindungen zum Beispiel benutzt man ein öffentliches Telefonnetzwerk bei einem IP-VPN unter Umständen das Internet. Beides sind im engeren Sinn öffentliche Netze, jedoch sind die Möglichkeiten für potenzielle Angreifer sehr unterschiedlich. Das Bedrohungspotenzial reicht vom mehrtägigen Lahmlegen von Unternehmensnetzen oder kritischen Netzwerkverbindungen bis zum Ausspionieren von Betriebsgeheimnissen mit

nachfolgenden Milliarden Schäden, etwas, wovon einige europäische High-Tech-Firmen ein trauriges Lied singen können.

Aber auch weniger dramatische Szenarien sind durch die Hacker-Angriffe, insbesondere in den letzten Jahren, zunehmend ins Bewusstsein einer breiten Öffentlichkeit und leider manchmal erst dadurch auch in das der Verantwortlichen in unseren Unternehmen gerückt. Denial-of-Service-Angriffe, also die Sabotage von Kommunikationseinrichtungen, das Eindringen in fremde Netze, Ausspionieren fremder Daten oder das Einschleusen von Viren, sind nur einige Beispiele erfolgreicher Angriffe auf Netzwerke, die in letzter Zeit Schlagzeilen gemacht haben. Die Schäden belaufen sich in Einzelfällen in Millionenhöhen, in der Gesamtheit reden wir von etlichen Milliarden, die jedes Jahr durch Hacker verursacht werden – ganz zu schweigen von der Grauzone derer, die mit dem ihnen entstandenen Schaden nicht an die Öffentlichkeit können oder wollen.

Die einfachste und zuverlässigste Sicherheitsstrategie, nämlich eine vollständige Abschottung des Unternehmensnetzwerks, ist natürlich aufgrund dessen, was wir bisher in diesem Kapitel besprochen haben, nicht möglich. Fast alle heutige Unternehmen und Organisationen sind auf Weitverkehrsnetze, Internetzugriff, Internetpräsenz, B2B und mobilen Fernzugriff angewiesen. Es geht einfach nicht mehr ohne. Also müssen entsprechende Maßnahmen ergriffen werden, um diese Applikationen mit einem Maximum an Sicherheit und Zuverlässigkeit zu betreiben.

Als zentrale Punkte, die je nach Art und Sensibilität der zu übertragenden Daten von der Netzwerk-Sicherheitstechnologie abgedeckt werden müssen, sind folgende Bereiche herauszuheben:

- ▶ Datenvertraulichkeit
- ▶ Datenintegrität
- ▶ Datenauthentizität
- ▶ Verfügbarkeit der Übertragungseinrichtungen

Zur Sicherstellung dieser Anforderungen sind die entsprechenden Maßnahmen (z.B. Verschlüsselung, Authentifizierung, usw.) zu implementieren. VPN- und Sicherheitsprotokolle wie IPSec oder SSL bieten sehr gute Möglichkeiten, diese Sicherheitsbedürfnisse weitgehend zu befriedigen und VPN damit nicht zu einem Sicherheitsrisiko werden zu lassen.

1.4 Randbedingungen für den Einsatz

Der sinnvolle und mögliche Einsatz eines VPN hängt von verschiedenen Rahmenbedingungen ab, die von Land zu Land und von Benutzer zu Benutzer sehr unterschiedlich sein können.

Eine der Hauptmotivationen, der Kostenaspekt, ist sehr variabel, sowohl zeitlich als auch regional. In den meisten Ländern gibt es sehr unterschiedliche Tarife, sowohl im Bereich der öffentlichen Netze als auch für den Internetzugang. Da VPN oft international ausgelegt werden, muss man mit den unterschiedlichsten Tarifen und Zugangstechnologien kalkulieren, was schnell etwas unübersichtlich werden kann. Je weiter ein VPN international gespannt wird, desto komplexer werden die Kostenkalkulationen, die auch ständigen Änderungen durch die glücklicherweise meist fallenden Gebühren unterworfen sind.

Weiterhin gibt es auch die unterschiedlichsten Zugangstechnologien, die, sowohl von der Technik als auch von den Kosten her gesehen, regional wieder sehr unterschiedlich sein können. Hinzu kommt, dass nicht alle Provider immer alles anbieten, was technisch möglich wäre. Während in Deutschland ISDN sehr verbreitet ist, kennt in den USA diese Technik im Heimbereich praktisch niemand, sie wird noch nicht einmal angeboten, außer in ganz wenigen Fällen.

Weitere Rahmenbedingungen sind rechtlicher Natur. Bei Internet-VPN finden in der Regel bestimmte Verschlüsselungstechnologien Anwendung. Aber in bestimmten Regionen unserer Erde darf man solche Produkte überhaupt nicht einsetzen; in anderen Ländern sind nur Produkte mit so genannter schwacher Verschlüsselung erlaubt. Wiederum andere Länder beschränken den Export solcher Systeme. In Deutschland fällt die Ausfuhr starker Verschlüsselungssysteme beispielsweise unter das Kriegswaffen-Kontrollgesetz.

Auch die Zukunft von VPNs, die neben Daten auch Sprache übertragen können, ist noch nicht abschließend geklärt! Die heutige öffentliche Sprachkommunikation darf nach einer richterlichen Genehmigung von den Strafverfolgungsbehörden abgehört werden – das ist in allen Ländern gang und gäbe. Die Telekommunikationsunternehmen stellen die dafür notwendigen Schnittstellen zur Verfügung. Was wird aber, wenn Sprachdaten zunehmend über Internet-VPN übertragen werden und die IP-Pakete durch den Einsatz von starker Verschlüsselung vor jedem Abhören geschützt sind, da die Ver- und Entschlüsselung seitens des Endbenutzers und nicht des Carriers vorgenommen wird?

Man muss in diesem Zusammenhang kritisch auf die Tendenzen in Ländern wie in den USA oder in Großbritannien schauen, in denen das Abhören jedweder Kommunikation nach dem 11. September 2001 offensichtlich zum allerhöchsten Gut erhoben wurde. Ein vergleichbares deutsches Telekommunikations-Überwachungsgesetz wurde zwar bisher erfolgreich von den einschlägigen Interessenverbänden von Industrie und Handel verhindert, allerdings war dies nur deshalb möglich, weil es sich bisher fast ausschließlich um Daten- und nicht um Sprachkommunikation handelte. Angesichts der Tendenz, immer öfter auch Sprache im VPN zu nutzen, ist es mehr als fraglich, dass in Deutschland und anderen europäischen Ländern Privatsphäre und Unternehmensinterna noch sehr viel länger als schützenswerte Güter angesehen werden.

1.5 Der VPN-Markt

Der VPN-Markt ist mittlerweile ein großer und immer noch sehr dynamischer Markt geworden. In den USA begann vor etwa sieben Jahren ein regelrechter Boom im Bereich der Internet-VPN. Dort vermeldete eine Reihe kleiner Startup-Unternehmen große Erfolge, die sich auf reine IP-VPN-Technologie konzentrierten und die erste reine VPN-Hard- und -Software entwickelten. Auch die Großen der Netzwerkbranche reagierten wie üblich durch die Übernahme einiger dieser Firmen und die Integration von deren Produkten in das eigene Produktportfolio. Dies erschwert leider die Auswahl eines insbesondere kleinen Herstellers, da die Zukunft etlicher Firmen und ihrer Produkte nur sehr schwer vorherzusagen ist. So hatte zum Beispiel die Firma Neoteris, ein SSL-VPN-Spezialist, innerhalb kürzester Zeit praktisch zweimal den Besitzer gewechselt.

Dieses Problem wird noch dadurch verschärft, dass ausgerechnet die kleinen Unternehmen oft die interessantesten und besten Produkte haben. Nicht selten wurden solche Unternehmen von Mitarbeitern großer Firmen gegründet, die ihre Ideen an ihrem alten Arbeitsplatz nicht verwirklichen konnten. Aus Frustration über die meist unvermeidliche Bürokratie und die langen, manchmal schwer nachvollziehbaren Entscheidungswege in Großunternehmen verließen sie ihre Arbeitgeber und gründeten mit anderen Leidensgenossen eine eigene Firma. Geldgeber, die ausreichend Kapital in solche Unternehmen stecken, finden sich in den USA auch heute noch recht problemlos, und die rechtlichen Rahmenbedingungen sind dort ebenfalls sehr viel innovationsfreundlicher als zum Beispiel im regelungswütigen Deutschland. In solchen kleinen Firmen, in der Atmosphäre eines High-Tech-Startups und mit der hohen Motivation ihrer Mitarbeiter, die alle an ihre Ideen glauben, wurden in den letzten Jahren die wichtigsten Technologien und Systeme im Bereich der VPN entwickelt.

Auch in Deutschland begann sich im Jahre 1999 der Internet-VPN-Markt sehr positiv zu entwickeln. Aufgrund des rasanten Wachstums des Internets und des wachsenden Bedarfs von Unternehmen an internationaler, breitbandiger Weitverkehrskommunikation sagten die Analysten damals sogar ein noch viel stärkeres Marktwachstum voraus – und hatten sogar ausnahmsweise einmal Recht damit.

Wie fast alle Entwicklungen, die hauptsächlich in den USA entstanden, war auch diese Technologie erst ein bis zwei Jahre später nach Deutschland gekommen. Dies hing mit einer ganzen Reihe von ungünstigen Faktoren zusammen, vor allem mit der späten Liberalisierung des Telekommunikationsmarktes, einer gewissen Abwartehaltung der potenziellen Anwender und der ärgerlichen Tendenz einiger Carrier und Service Provider, ihre herkömmlichen WAN-Systeme und Dienstleistungen, in die sie einiges für Entwicklung und Anschaffung investiert hatten, erst einmal weiter zu vermarkten und das Thema VPN am besten gar nicht zu erwähnen. Die Fachliteratur hatte sich zum damaligen Zeitpunkt weitgehend zu dem Thema ausgeschwiegen, und auch in der Fachpresse wurde dem Thema lange Zeit eher eine Nischenposition zugewiesen. Selbst viele Hersteller hatten es, verständlicherweise, nicht eilig, sich mit dieser Technologie zu befassen, da sie mit traditionellen Routern und Remote-Access-Konzentratoren einen um ein Vielfaches höheren Umsatz machen konnten als mit der vergleichbaren VPN-Technologie. Groteskerweise haben sogar gewisse Hersteller, die für viel Geld kleinere VPN-Hersteller übernahmen, weiterhin versucht, ihre alten Routerprodukte – versehen mit dem Label „VPN“ – weiter zu vermarkten.

So fehlte denn aufgrund mangelnder Informationen auch lange Zeit der Druck vom Kunden, und das Thema wurde bei uns zu einem Spätstarter. Aber frei nach dem Motto „Lieber spät als nie“ boomt der Markt schon seit einiger Zeit auch in Deutschland, insbesondere im Bereich der Internet-VPN. Insbesondere der durch die wirtschaftliche Gesamtsituation hervorgerufene Zwang zur Kostensenkung, der mittlerweile fast vor keiner Branche mehr Halt macht, hat dem Thema VPN nochmals einen deutlichen Schub verpasst.

Die zukünftige Marktentwicklung für virtuelle private Netze wird denn auch von den Analysten immer noch als sehr positiv bewertet.

Hinzu kommt, dass die Carrier und Service Provider IP-VPN als Komplettlösung in ihr Angebotsportfolio aufgenommen haben und in diesem Bereich eine Reihe unterschiedlicher Lösungen offerieren. Diese reichen vom einfachen Internet-Zugriff bis hin zu Komplettlösungen inklusive Netzwerkmanagement. Es wird wohl in Zukunft immer mehr so aussehen, dass sich der Netzwerkplaner nicht mehr fragt, ob er die VPN-Technologie einsetzen soll, sondern wie.

1.6 Internet-VPN

Auslöser für diese Entwicklung gibt es gleich mehrere. Nachdem sich das Internet schon vor einigen Jahren endgültig von dem Begriff „free“ im Sinne von „umsonst“ verabschiedet hat und nachdem es aus dem universitären Bereich herausgelöst wurde und sich die Großen der Telekommunikation seiner angenommen haben, wurde und wird es in einem Maße ausgebaut, wie es vor einigen Jahren niemand vorausgesagt hätte. Diese Entwicklung findet gleichzeitig in mehreren Bereichen statt:

Die Zahl der Internetzugänge und damit gleichbedeutend die Zahl der Endgeräte nimmt rasant zu. Es ist zwar noch lange nicht so, dass jeder Rechner auf der Welt einen Internetanschluss hat, aber bedingt durch ständig sinkende Kosten und den steigenden Mehrwert des Netzes nimmt ihre Zahl deutlich zu.

Die Bandbreite des Internets wird der in Zukunft notwendigen Kapazität angepasst. Im Augenblick gibt es sogar ein regelrechtes Bandbreiten-Überangebot, vor allem im Backbone-Bereich. Man darf sich hier durch das manchmal langsame World Wide Web nicht täuschen lassen. Dessen mitunter etwas dürrtige Performance beruht meist auf völlig unterdimensionierten Serverfarmen ohne vernünftige Lastverteilung und auf einer zu schmalbandigen Anbindung an den Internet-Backbone. Mit der Netzgeschwindigkeit selbst hat dies meist nichts zu tun, aber gerade diese interessiert bei der Planung eines Internet-VPN.

Die Verfügbarkeit des Internets für den Endkunden wird deutlich höher. Dies liegt hauptsächlich daran, dass die Service Provider zunehmend durch so genannte SLA (Service Level Agreement, eine Vereinbarung zwischen Service Provider und Endkunde über Dienstqualitäten wie Bandbreite und Verfügbarkeit) gezwungen werden, für den Betrieb ihrer Datennetze ähnliche Verfügbarkeitskriterien wie im Sprachvermittlungsbereich anzulegen.

Hinzu kommen aktuelle Entwicklungen im Bereich CoS (Class of Service, Abstufung verschiedener Dienstqualitäten hinsichtlich Bandbreite, Verzögerung und Verfügbarkeit), mit denen die Service Provider die Schwäche des IP-Protokolls in diesem Bereich kompensieren sollen. Dies alles deutet darauf hin, dass das zukünftige Internet ein weltweit verfügbares IP-Netzwerk mit einer hohen, skalierbaren Bandbreite, in Grenzen deterministischen Verzögerungszeiten, einer hohen Verfügbarkeit und einer Vielzahl unterschiedlicher Zugangstechnologien sein wird.

Obwohl sich das alles auf den ersten Blick sehr positiv anhört, muss man, zumindest heute und in näherer Zukunft, noch einige Einschränkungen machen. Dies hängt mit der Struktur und der augenblicklich eingesetzten Technologie des Internets zusammen. Das Internet basiert auf dem IP-Protokollstandard und den ganzen anderen damit zusam-

menhängenden RFC (Request for Comments, Standards im Internetbereich), die teilweise schon einige Jahre auf dem Buckel haben und unter anderen technischen, gesellschaftlichen und kommerziellen Rahmenbedingungen entwickelt wurden als heute vorliegen. Obwohl, jawohl Sie lesen richtig, das IP-Protokoll ursprünglich für den Transport von Sprache entwickelt wurde, hat sich Ausrichtung im Laufe der Zeit immer mehr in Richtung Datentransport verändert. Im Protokoll-Header vorgesehene Felder zur Signalisierung von Informationen über benötigte Dienstqualitäten wurden jahrelang praktisch ignoriert.

Aber heutige Applikationen benötigen immer öfter derartige Features, welche die IP-Protokollfamilie in ihrer jetzigen Form noch nicht bietet oder die noch nicht flächendeckend im Internet eingesetzt werden. So ist es zum Beispiel heute technisch noch nicht möglich, im Internet eine bestimmte Bandbreite oder Verzögerungszeit zu garantieren. Obwohl dies einige Service Provider in ihren SLA tun, besitzen sie überhaupt noch nicht die technischen Möglichkeiten dazu, sie hoffen meist einfach, dass es schon klappen wird oder dass man sie nicht erwischt. Es gibt zwar Ausnahmen, allerdings setzen diese voraus, dass man sich ausschließlich im Netz des betreffenden Anbieters bewegt oder in Netzen anderer Provider, die mit diesem sowohl ein entsprechendes Service Level Agreement haben als auch geeignete Systeme betreiben.

Außerdem ist die Verfügbarkeit des Internets immer noch ein äußerst kritischer Punkt, da etliche kleinere und mittlere Vermittlungsknoten immer noch mit Routern betrieben werden, die ursprünglich für den Einsatz in Unternehmensnetzen entwickelt wurden und deren Ausfallsicherheit in keiner Weise den im Carrier-Umfeld sonst üblichen Maßstäben entspricht. Im Kernbereich des Internets, also in redundanten Hochgeschwindigkeits-Glasfasernetzen, wird zwar in der Regel so genannte „Five-Nine“-Technologie, die eine Verfügbarkeit von mehr als 99,999% garantiert, eingesetzt. Auf der IP-Routing-Ebene und mit zunehmender „Nähe“ zum Endkunden findet zunehmend herkömmliche Router-Technologie Verwendung, die nicht unbedingt solch hohe Verfügbarkeiten garantieren kann.

Wenn man sich die Struktur des Internets einmal versinnbildlicht, kann man es grundsätzlich in drei unterschiedliche Bereiche gliedern, die vom Backbone- bis zum Access-Bereich abgestuft sind. Im Access-Bereich sind IP-Endgeräte zu finden, also Endgeräte wie PCs, UNIX-Workstations, VPN-Gateways und Access-Router. Der Internet-Backbone dient ausschließlich dazu, mehrere Netze miteinander zu verbinden.

Aufgrund dieser Unterscheidung gibt es auch drei Klassen von ISP (Internet Service Provider, eine Organisation, die Dienste wie den Zugriff auf das Internet, E-Mail usw. zur Verfügung stellt), und zwar solche, die nur einen Teil des Backbones betreiben, solche, die nur die Access-Technologie bereitstellen, und solche, die beides machen, über deren Infrastruktur sowohl ein Teil des Internet-Backbone-Verkehrs transportiert als auch Zugriffsdienste für Endanwender bereitgestellt werden.

Nach alldem sieht die Zukunft von virtuellen privaten Netzwerken sehr rosig aus. Die wichtigsten Rahmenbedingungen stimmen, die Technologie ist verfügbar, ebenso eine Auswahl der verschiedensten öffentlichen Netze, und die potenziellen Kunden haben die Idee der VPN angenommen. Nun stellt sich die Frage, welche Art von VPN man einsetzen soll. Was ist am besten geeignet? Was ist zukunftssicher? Was ist am kostengünstigsten?

Diese Fragen lassen sich einfach beantworten, denn es gibt praktisch keine Alternative mehr zu einem IP-VPN. Das Internet-Protokoll, so alt es auch sein mag, ist das Netzwerkprotokoll der Zukunft. Es wird alle anderen, noch übrig gebliebenen Netzwerkprotokolle verdrängen. Man muss sich nur einmal die aktuellen Projekte der IETF (Internet Engineering Task Force, dem Gremium zur Entwicklung und Verabschiedung von Internet-Standards) anschauen, dann weiß man, wohin der Internet-Zug fährt. Es wird in Zukunft viel besser als heute möglich sein, über das Internet unternehmenskritische Applikationen zu betreiben, Geschäfte abzuwickeln, vertrauliche Informationen zu übertragen und auch zu telefonieren.

Wer jetzt sein VPN aufbaut und gewisse Anforderungen hat, die augenblicklich nur über ATM oder Frame Relay zu erfüllen sind, hat trotzdem noch alle Optionen. Denn der Einsatz solcher Netze verbietet nicht den Einsatz von IP als Netzwerkprotokoll. Wenn man sein Netzwerk-Design dementsprechend gestaltet, ist eine spätere Migration kein großer Aufwand mehr und zieht vor allem keinen Bruch in der privaten Netzwerkstruktur nach sich. Einige moderne VPN-Router bieten sogar die Möglichkeit, IP-VPN mit Frame-Relay oder ATM zu integrieren.

Die zukünftige Entwicklung im Internet wird vor allem im Bereich der Erhöhung der Bandbreite im Access-Bereich und der Verfügbarkeit sowie der durchgehenden Implementierung von Bandbreitenmanagement und verschiedenen Dienstqualitäten stattfinden.

Auch im Bereich der Sicherheitstechnologie ist die Richtung momentan klar auf Steigerung der Performance ausgerichtet, neue Algorithmen sind momentan nicht das Hauptthema der einschlägigen Entwicklungslabors. Auch hier gibt es vielversprechende Entwicklungen, so wurde kurz vor Fertigstellung dieser Auflage erstmals die 10-GBit/s-Grenze bei der Datenverschlüsselung überschritten: So präsentierte der Netzwerkhersteller Nortel auf einer Konferenz im September 2005 eine erste Implementierung einer Verschlüsselung mit einer Geschwindigkeit von 10 GBit/s (AES-Algorithmus mit 256-Bit-Schlüssel) und einer maximalen Latenzzeit von 400 ns in seinen optischen Systemen. Dabei wurden vier Standorte in Amsterdam, Ottawa, Chicago und San Diego mit verschlüsselten 10-GBit/s-Glasfaserverbindungen vernetzt.

Auch in den Labors der einschlägigen Chiphersteller wird eifrig entwickelt, um die aktuellen Verschlüsselungsalgorithmen durch spezielle Hardware-Implementierungen abzubilden. Bei dem Maß, in dem die Kryptografie gerade Einzug in Systeme und Netze hält, ist der erste PC-Chipsatz mit integrierter Verschlüsselungshardware vermutlich nur eine Frage der Zeit. Für Netzwerkadapter gibt es so etwas ja schon seit einiger Zeit.

Im Zuge dieser Entwicklungen wird auch die Paketverzögerung im Internet immer geringer. Diese hängt von der Anzahl der verschiedenen Netzknoten und von deren jeweiliger Verarbeitungsgeschwindigkeit ab. Je schneller diese Knoten sind und je weniger Knoten insgesamt durchlaufen werden, desto besser wird dieser Wert. Vor allem die neuen rein optischen Verstärker eliminieren die Verzögerungen, die durch die zweifache Medienkonvertierung in herkömmlichen opto-elektronischen Systemen nicht vermeidbar sind.

An der Verfügbarkeit des Internets wurde ebenfalls fleißig gearbeitet. Da zunehmend Hersteller am weiteren Auf- und Ausbau beteiligt sind, die teilweise schon seit über hundert Jahren im Telefongeschäft tätig sind oder sich auf Carrier-Technologie spezialisiert haben und deren Hauptkriterium daher schon immer Ausfallsicherheit war, sind auch

hier positive Entwicklungen festzustellen. Dass das Internet insgesamt einmal eine Ende-zu-Ende-Verfügbarkeit von 99,999% haben wird, ist natürlich noch Zukunftsmusik. Dies liegt daran, dass es genau genommen „das Internet“ nicht gibt. Es ist vielmehr ein Zusammenschluss von vielen Zugangs- und Backbone-Netzen verschiedenster Betreiber, in denen die unterschiedlichsten Technologien eingesetzt werden. Aber zumindest im Backbone-Bereich und zunehmend auch im Access-Bereich ist die Verfügbarkeit des Netzes ein Thema geworden, das die Kunden auch in den Verträgen mit den Providern festlegen wollen.

Aber nicht nur in diesem Bereich, sondern auch im Zugangsbereich, der so genannten „Last Mile“, gibt es neue Entwicklungen. Auch hier können Konkurrenten der Telekom deren Leitungen und Räumlichkeiten (Co-Lokation) nutzen, um dem Kunden vollständig bedienen zu können. Ebenso muss das Breitbandkabelnetz (Fernsehen, Rundfunk) der Telekom von dieser aufgrund einer Entscheidung des europäischen Gerichtshofs vollständig verkauft werden. Durch geeignete Technologien und geringfügige Änderungen an den Übertragungskomponenten ist zusätzlich Datenverkehr und Telephonie über das Breitbandkabelnetz möglich, und einige Provider sind auch schon dabei das, leider erst zum Teil verkaufte, Kabelnetz technisch für diese neuen Dienste umzurüsten.

DSL hat in Deutschland mittlerweile eine der höchsten Durchdringungsraten der Welt erreicht und bringt, im Vergleich zu ISDN, wirklich hohe Geschwindigkeiten in kleine Büros oder zu Heimarbeitsplätzen.

Im Mobilbereich wurde sehr viel in neue Infrastrukturen investiert, und in den nächsten Jahren werden noch weitere zig Milliarden Euro folgen. Während sich die traditionellen GSM-Netze und ihre Festnetzkomponenten schon in der so genannten Phase 2,5 befinden und damit auch spezielle paketorientierte Datendienste bereitstellen, steht mit UMTS eine neuere und im Endausbau deutlich schnellere Mobiltechnologie zur Verfügung. Obwohl UMTS noch in den Anfängen steckt und die peinliche Versteigerung der Lizenzen in Deutschland nur die erste von vielen folgenden negativen Randerscheinungen war, wird es die bisherige GSM-Technologie langfristig ablösen – eine der UMTS-Lizenzbestimmungen ist ja die Forderung nach 50% Flächendeckung im Jahr 2005. Die meisten Mobilbetreiber haben zudem durchsickern lassen, dass sie ihre GSM-Lizenzen nicht mehr zu verlängern gedenken, wenn sie UMTS erst einmal flächendeckend in Betrieb haben.

So werden auch im Zugangsbereich, egal ob kabelgebunden oder drahtlos, die neuen schnelleren Technologien den VPNs einen weiteren Schub geben.

1.7 VPN-Typen

Bei VPN unterscheidet man, abhängig vom Einsatzgebiet, zwischen verschiedenen Arten, wobei diese durchaus auch miteinander kombiniert werden können. Die unterschiedlichen Arten von VPN lehnen sich dabei stark an ihre korrespondierenden klassischen Netzwerkstrukturen an.

1.7.1 Remote Access VPN

Ein Remote-Access-VPN ermöglicht es, von entfernten Systemen aus auf ein Unternehmensnetz zuzugreifen. Auf herkömmliche Weise erreicht man dies durch den Einsatz von RAC im Unternehmensnetz. Ein RAC (Remote Access Concentrator) ist ein System, das an öffentliche Telefonnetze angeschlossen wird und die analoge oder digitale Einwahl in diese Netze ermöglicht. Standard-RAC terminieren Verbindungen unterschiedlicher Natur wie zum Beispiel ISDN oder analoge Rufe über Modems. Sie werden in der Regel, je nach Anzahl der maximal gleichzeitig zu terminierenden Verbindungen, über einen oder mehrere Primärmultiplexanschlüsse mit dem Telefonnetz verbunden. Diese Primärmultiplexanschlüsse (PMX), in Deutschland als S2M-Anschluss bezeichnet, sind eine digitale, strukturierte 4-Draht-Verbindung mit einer Bandbreite von 2,048 MBit/s. Sie besteht aus 30 Nutzdatenkanälen, einem Signalisierungskanal und einem Steuerungskanal, kann also gleichzeitig maximal 30 Wählverbindungen betreiben.

Die RAC müssen technisch dafür ausgelegt sein, alle vorkommenden analogen und digitalen Protokolle verarbeiten zu können. In heutigen Systemen müssen Modemprotokolle wie V.90, V.34, V34+ und manchmal auch noch X.2 oder k56flex implementiert werden. Im digitalen Bereich müssen sowohl synchrones ISDN als auch asynchrone Varianten wie V.120 und, vor allem wenn eine Einwahl aus einem GSM-Netzwerk unterstützt werden soll, auch das V.110-Protokoll implementiert werden. In der Regel sollen auch noch alle Dienste unter einer einzigen Rufnummer erreichbar sein und erst im RAC unterschieden werden.

Dies macht heutige Remote-Access-Konzentratoren technisch sehr komplex und damit auch meist sehr teuer. Weiterhin müssen solche Geräte auch skalierbar sein, da die Anzahl der benötigten Ports in der Regel stetig wächst. Skalierbarkeit bedeutet in diesem Kontext nicht nur, dass eine ausreichend große Anzahl von Einwahlports verfügbar sein muss, sondern auch, dass die interne Verarbeitungskapazität und die Anbindung an das Intranet ebenfalls mitwachsen müssen, andernfalls entsteht zunehmend ein Performance-Engpass. Somit sieht man sich mit einer Reihe von Anforderungen und den damit verbundenen Kosten konfrontiert, die einen Remote-Access-Dienst üblicherweise sehr kostenintensiv machen:

Man muss eine relativ teure Technologie zum Terminieren der verschiedenartigen Verbindungen aus dem öffentlichen Telefonnetz beschaffen und warten.

Die Technologie muss ständig an neue technische Gegebenheiten und wachsende Kapazitäten angepasst werden.

Man zahlt eine vergleichsweise hohe Grundgebühr für die benötigten Primärmultiplexanschlüsse.

Die Verbindungsgebühren sind ebenfalls hoch, insbesondere wenn häufig Verbindungen im Fernbereich oder gar internationale Verbindungen benötigt werden.

Andere Dienste wie DSL (Digital Subscriber Line), Kabelmodems, die Daten über Breitband-Fernsehkabel übertragen, oder Wireless LAN Hotspots sind mit dem RAC nicht zu verarbeiten und erfordern zusätzlichen technischen Aufwand.

Ein Remote-Access-VPN befasst sich mit genau diesen kritischen Faktoren, die einen Remote-Access-Dienst sehr teuer und aufwändig machen können. Sein Ziel ist es, die Hardware zum Terminieren der Verbindungen kostengünstig und einfach zu halten und die Verbindungsgebühren zu minimieren.

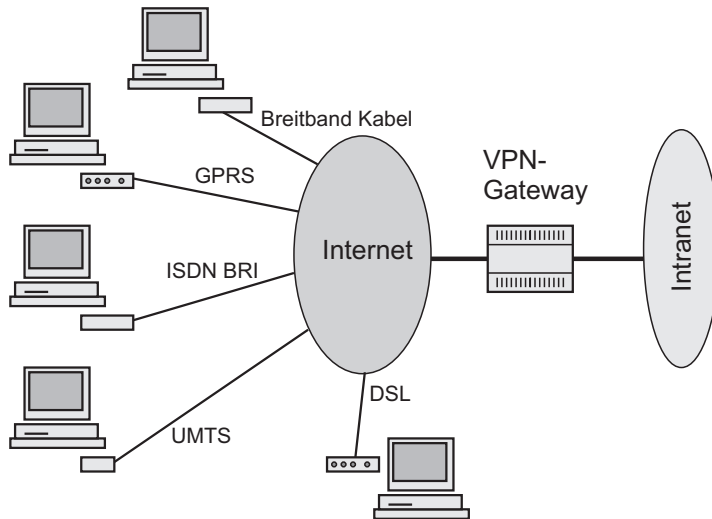


Abbildung 1.9: Ein typisches Remote-Access-VPN ist unabhängig von der lokal eingesetzten Zugangstechnologie.

Ein Remote-Access-VPN besteht aus einem VPN-Gateway oder VPN-Router, der die virtuellen Remote-Access-Verbindungen terminiert, und Software-Clients, die auf den entfernten Rechnern installiert werden, um die Verbindungen aufzubauen. In seltenen Fällen werden die Verbindungen auch erst im RAC des Internet Service Providers initiiert, so dass in diesem Fall auf dem Endgerät keine spezielle Client-Software notwendig ist. Der VPN-Router terminiert also nur logische Verbindungen, so genannte Tunnel. Die Anbindung zu einem Internet Service Provider erfolgt meist per LAN-Interface an einem Access-Router, den der Provider zur Verfügung stellt und betreibt, oder in Ausnahmefällen über die WAN-Schnittstelle eines Kunden-Routers direkt zum Provider.

Die Clients können sich mit beliebigen Übertragungstechnologien mit einem Service Provider verbinden. Ob man Modems, ISDN, xDSL oder Kabelmodems einsetzt, hängt lediglich davon ab, welche Zugangstechnologie die lokalen Service Provider in den jeweiligen Regionen anbieten. Der Endkunde selbst braucht kein zentrales Equipment zur Terminierung dieser Verbindungen mehr bereitzuhalten, dies erledigen die Service Provider. Der Kunde terminiert nur so genannte Tunnel, die vom Provider zu seinem VPN-Gateway aufgebaut werden. Auch eine steigende Anzahl von Benutzern ist damit zum Problem der ISP geworden, die andererseits aber auch ein vitales Interesse daran haben, dass genügend Ports zur Verfügung stehen. Denn wer sich nicht bei einem Service Provider einwählen kann, der beschert ihm auch keinen Umsatz.

Bei den Gebühren ergeben sich mit einem Remote-Access-VPN erhebliche Kostenvorteile, insbesondere bei einer großen Flächendeckung oder bei internationalem Einsatz. Man baut nämlich keine Telefonverbindung vom Endgerät zu einem zentralen RAC im Unternehmensnetz mehr auf, sondern nur noch irgendeine, am jeweiligen Ort verfügbare Verbindung zum Internet auf. Dies kann mit beliebigen Technologien geschehen, mit ISDN, analogen Telefonverbindungen, Wireless LAN Hotspots, UMTS, GPRS, LAN oder wie auch immer.

Bei Kostenvergleichen, die auf Anschlussgebühren, zeitabhängigen Tarifen, Entfernungszonen und Verbindungszeiten basieren, kann man Einsparungen im Bereich von 70 bis 80% erzielen. Bei den Gesamtsummen, die viele Unternehmen pro Jahr für normalen Remote Access ausgeben, ist dies ein erhebliches Einsparpotenzial. Die Vorteile durch die höhere Produktivität mobiler Mitarbeit wurde hierbei noch gar nicht mit berücksichtigt.

Ein weiterer wichtiger Punkt, den man beachten sollte, ist der deutlich geringere Port-Preis eines VPN-Routers im Vergleich zu einem RAC. Ab einer Portzahl von einigen tausend Ports kostet ein VPN-Port bei einigen Herstellern weniger als 5% eines Remote-Access-Ports! Der Grund für diesen krassen Preisunterschied liegt darin, dass VPN-Router nicht eine Vielzahl von Technologien, wie Telefonsignalisierung, Modemprotokolle, Analogverarbeitung usw., implementieren müssen und daher von der Hardwarearchitektur vergleichsweise einfach aufzubauen sind. Es gibt Hersteller, die sogar Standard-PC-Architekturen mit Windows oder UNIX als VPN-Router benutzen, also überhaupt keine Hardware-Entwicklungskosten haben.

Beim Einsatz einer VPN-Technologie, bei der die virtuelle Verbindung auf dem Client initiiert wird, ist man auch völlig unabhängig vom Internet Service Provider. Man kann ihn jederzeit wechseln oder auch problemlos mehrere ISP gleichzeitig benutzen. Denn der Service Provider ist dann in keiner Weise mehr in die Funktion des VPN involviert, er terminiert lediglich Telefonanrufe und Festverbindungen und überträgt IP-Pakete zwischen Endgeräten und VPN-Routern

Branch Office VPN

Branch-Office-VPNs ersetzen die herkömmlichen WAN-Verbindungen, mit denen man verschiedene Standorte oder Netzwerke in diesen Standorten miteinander verbindet. Der Begriff Branch-Office-VPN hat sich mittlerweile weitgehend für diesen VPN-Typ durchgesetzt, gelegentlich spricht man auch von Site-to-Site-VPN. Warum besteht überhaupt eine Notwendigkeit, bisherige Weitverkehrsnetze, die mit verbreiteten Technologien wie Standardfestverbindungen, Frame Relay oder ATM eingerichtet wurden, durch ein VPN zu ersetzen? Auch hier gibt es einen Hauptgrund, nämlich die hohen Kosten durch die relativ hohen Verbindungsgebühren. Ganz besonders dramatisch sind die Kosten, wenn die zu verbindenden Standorte sehr weit voneinander entfernt oder gar im Ausland liegen. Je nach Anzahl, Entfernung, benötigter Bandbreite und zu übertragender Datenmenge kommen da schnell sehr hohe Kosten zusammen.

Tendenzen wie die Globalisierung, internationale Fusionen und Kooperationen sowie die ganzen neuen, so genannten E-Technologien (E-Business, E-Commerce, E-Learning usw.) führen dazu, dass die Kosten für die benötigte Datenkommunikation immer weiter in die Höhe schnellen. Der steigende Kostendruck zwingt auch im Weitverkehrsbereich immer mehr Unternehmen dazu, ihre eingesetzte WAN-Technologie neu zu überdenken und gegebenenfalls zu ändern, um konkurrenzfähig bleiben zu können.

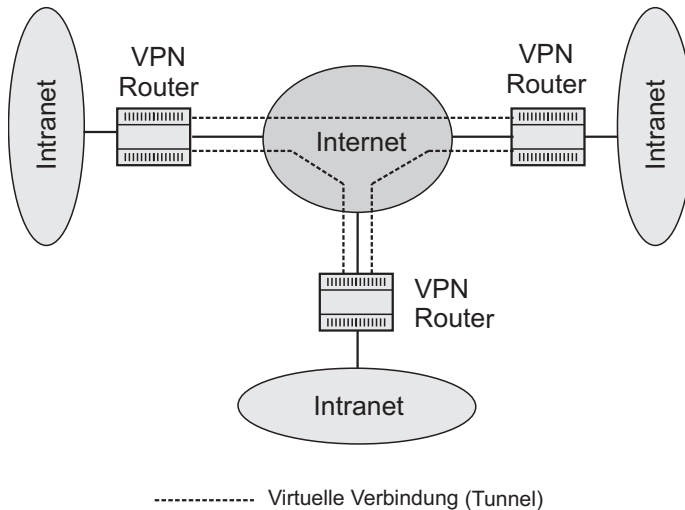


Abbildung 1.10: Ein Branch-Office-VPN verbindet verschiedene Standorte über virtuelle Verbindungen.

Als Ausweg aus dem Kostendilemma bieten sich zunehmend Branch-Office-VPNs an (siehe Abbildung 1.10). Der Faktor, um den die Übertragungskosten reduziert werden können, hängt im Wesentlichen von folgenden Faktoren ab: der Entfernung zwischen den Systemen, der übertragenen Datenmenge und der eingesetzten Technologie und Geschwindigkeit.

Eingreifen kann eine VPN-Lösung hauptsächlich beim ersten Faktor, nämlich bei der Entfernung zwischen den Standorten. Diese bleibt geografisch gesehen zwar nach wie vor bestehen, aber die kostenintensiven, direkten Verbindungen (Standardfestverbindung, Frame-Relay-PVC usw.) zwischen den privaten Netzen werden durch zwei sehr kurze Verbindungen zwischen den Standorten und den Zugangsknoten eines Internet Service Providers ersetzt. Die verbleibende Strecke zwischen diesen beiden POPs (Point of Presence, Zugangsknoten zum Internet) wird vom Internet überbrückt, über das die Datenpakete sehr kostengünstig übertragen werden können.

Das Einsparpotenzial ist hier nicht ganz so hoch wie bei Remote-Access-VPN, es ist jedoch durchaus möglich, die Gebühren um bis zu 50% zu reduzieren, im internationalen Bereich sogar um noch mehr.

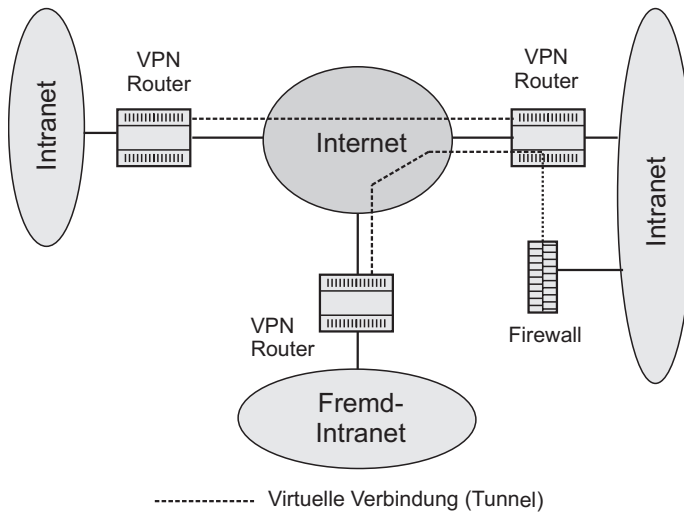


Abbildung 1.11: Extranet-VPNs benötigen spezielle Maßnahmen, um unternehmensfremden Personen Zugriff auf das Unternehmensnetz zu gewähren.

Extranet-VPN

Ein Extranet-VPN sieht von seiner Struktur her ähnlich aus wie ein Remote-Access- oder Branch-Office-VPN oder eine Kombination von beiden (siehe Abbildung 1.11). Der fundamentale Unterschied zu den beiden zuvor besprochenen VPNs liegt in der Natur der Teilnehmer. Ein VPN bildet ein rein privates Netzwerk ab, auf das nur Angehörige der eigenen Firma oder Organisation Zugriff haben und das nur eigene Standorte miteinander verbindet. Ein Extranet-VPN hingegen öffnet das private Netzwerk auch für externe Personen oder Organisationen und gewährt diesen (einen meist limitierten und kontrollierten) Zugriff auf Ressourcen im Unternehmensnetzwerk.

Extranet-VPN werden also auf der Basis der normalen VPN-Technologie aufgebaut, aber die Datenpakete stammen nicht von eigenen Mitarbeitern und müssen deshalb gesondert behandelt werden. Dies kann entweder das VPN-Gateway selbst tun, oder man übergibt die Pakete einem speziell dafür ausgelegten System, meist einer Firewall. In dem Beispiel in Abbildung 1.11 werden die Verbindungen zu eigenen Mitarbeitern im Intranet terminiert, die Verbindungen von der Partnerfirma jedoch auf einer Firewall. Für den eigentlichen Datentransport wird das VPN benutzt. Die Firewall ist für die besondere Behandlung der Extranet-Datenpakete zuständig, also für die dynamische, zustandsabhängige Filterung, für die Zugriffsbeschränkung, das Auditing und das Logging.

In kleineren Netzen kann man aus Kostengründen auch Firewalls und VPN-Router miteinander auf einer Plattform kombinieren. Allerdings haben viele gute VPN-Router auf ihrem öffentlichen Interface einen so genannten gehärteten Netzwerk-Stack, der von Hause aus nur ganz wenige Protokolle verarbeiten kann und damit sehr gut vor Angriffen geschützt ist. Beim gleichzeitigen Einsatz als Firewall muss dieser durch einen normalen Stack ersetzt werden, was unter Umständen ein Sicherheitsthema werden kann.

1.8 VPN-Dienste

Beim Einsatz von VPN hat man die Wahl, sein VPN selbst aufzubauen oder in verschieden starkem Maße so genannte VPN-Dienste von Service Providern oder Carriern zu benutzen. Das kann man in verschiedenen Abstufungen tun, je nachdem, wie stark man sich an einen Provider binden will.

IP-VPN arbeiten mit virtuellen IP-Verbindungen. Die Endpunkte dieser Verbindungen (Tunnel) sind die IP-Schnittstellen von Routern, VPN-Gateways oder VPN-Client-Systemen. Dies ermöglicht sowohl einen VPN-Betrieb vollständig in Eigenregie, also ganz ohne Mitwirkung eines Providers, als auch ein vollständiges Outsourcing und alle möglichen Zwischenstufen.

1.8.1 Eigenbetrieb

In diesem Fall ist der Service Provider in keiner Weise in den Betrieb des VPN involviert. Er stellt dem Kunden lediglich Internetzugänge zur Verfügung. Der Kunde beschafft und betreibt seine Access-Router und VPN-Gateways selbst. Das User- und Gruppenmanagement, ebenso wie Systemkonfigurationen, liegen im Verantwortungsbereich des Kunden.

Die Nachteile dieser Lösung sind der relativ hohe Aufwand für den Kunden und die vielen verschiedenen Schnittstellen. Denn hier sind mindestens drei Organisationen beteiligt: der Endkunde (Intranet und VPN), ein Carrier (Last-Mile-Übertragung) und ein Service Provider (Access-Router und Internetzugang). Bei Problemen kann es dabei zum allseits bekannten Hin- und Herschieben der Verantwortung kommen.

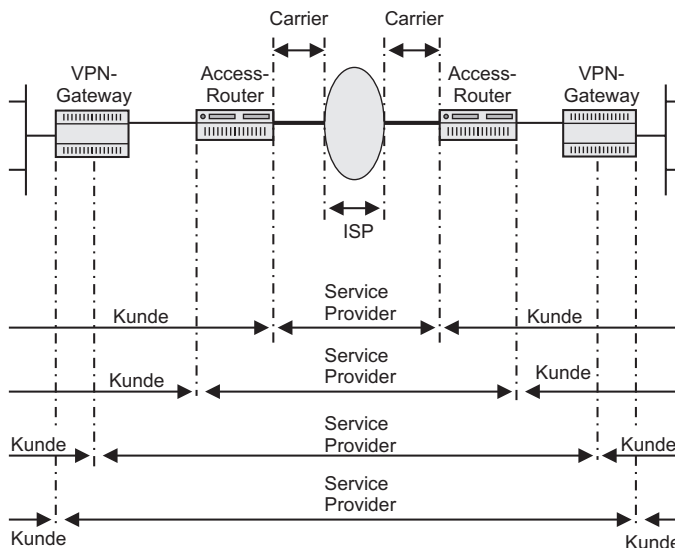


Abbildung 1.12: Die Verteilung der Verantwortlichkeiten zwischen Carrier, Service Provider und Kunde bei verschiedenen Outsourcing-Szenarien

Der Vorteil aus der Sicht des Kunden ist der, dass keine Bindung an einen Provider nötig ist und jederzeit zu einem anderen gewechselt werden kann.

1.8.2 Access-Equipment-Outsourcing

In diesem Fall wird vom Service Provider ein etwas größeres Leistungspaket gekauft, denn er ist in dieser Variante für die vollständige Kommunikation zwischen seiner Infrastruktur und einem fest definierten Punkt, meist dem LAN-Interface seines Access-Routers, verantwortlich.

Der Kunde beschafft und betreibt lediglich seine VPN-Gateways, die über einen LAN-Anschluss mit den Access-Routern des Service Providers verbunden werden.

Somit ist der Nachteil der vielen Schnittstellen für den Kunden entschärft, denn er hat nur noch einen Ansprechpartner, seinen Service Provider, falls die Weitverkehrsverbindung nicht funktioniert.

Seine Vorteile bleiben weitgehend bestehen, ein Wechsel ist auch hier nicht so kompliziert. In diesem Modell ist der Service Provider in keiner Weise in den VPN-Betrieb involviert.

1.8.3 VPN- und Access-Equipment-Outsourcing

Das ändert sich jedoch mit dieser Variante. Hier werden neben den Access-Routern und der Verbindungsinfrastruktur auch die VPN-Systeme vom Provider gestellt. Der Zugriff auf die VPN-Systeme durch den Service Provider beschränkt sich jedoch auf die reine Systemkonfiguration. Das VPN-Management, also die Konfiguration von Tunneln, Sicherheitseinstellungen, Gruppen, Benutzern usw. obliegt ausschließlich dem Kunden.

Dies erfordert aber unbedingt VPN-Gateways, die ein spezielles Split-Management ermöglichen, das diese beiden Funktionen sauber trennen kann.

Big Brother

Dies hat auch vor allem rechtliche Gründe. Denn in Deutschland und den meisten anderen Ländern auch sind Carrier und Service Provider verpflichtet, in ihren Kommunikationsnetzen den Ermittlungsbehörden im Bedarfsfall, der allerdings eine richterliche Genehmigung erfordert, eine technische Möglichkeit zum Abhören (Lauschangriff) zur Verfügung zu stellen. In Internet-VPNs wird jedoch aus Sicherheitsgründen fast ausschließlich IP-Security (IPSec) mit starker Verschlüsselung eingesetzt. Wie Sie in Kapitel 5 und 6 erfahren, ist IPSec jedoch so zu implementieren, dass weder eine Schlüsselhinterlegung noch eine Schlüsselrückgewinnung möglich ist. Demnach kann ein Service Provider oder Carrier Abhörmaßnahmen nur dadurch ermöglichen, dass er die IPSec-Verschlüsselung ganz abschaltet, indem er die Sicherheitsstrategie im VPN-Gateway entsprechend ändert.

Da ein Abhören aber nur dann Sinn macht, wenn der Abgehörte nichts darüber weiß, darf der Service Provider seinen Kunden auch nicht darüber informieren – und kommt damit in Teufels Küche, denn die Pakete, die dann ja alle unverschlüsselt durch das Internet transportiert werden, können dadurch auch von anderen, nicht berechtigten Personen abgehört werden.

Das Split-Management ist ein möglicher Ausweg aus diesem Dilemma. Denn Endanwender, also Firmen oder Privatpersonen, dürfen, zumindest in Deutschland, nach Belieben verschlüsseln. Auf diese Weise kann der Service Provider oder Carrier dem Kunden ein VPN-Gateway zur Verfügung stellen und teilweise auch managen; der

Kunde hat jedoch ein anderes Management-Interface und kann seine Benutzer, Gruppen und Tunnel mit der geeigneten Verschlüsselung konfigurieren. Der Knackpunkt dabei ist der, dass der Endbenutzer die Rechte auf seinem System vergibt und dem Service Provider eben nur eingeschränkten Zugriff gewährt.

1.8.4 Vollständiges Outsourcing

Dieses Modell ist für den Kunden mit dem geringsten Aufwand verknüpft, denn der vollständige Betrieb des VPN wird vom Service Provider durchgeführt – inklusive der Benutzerverwaltung.

Der Nachteil ist der, dass hier eine sehr starke Abhängigkeit vom Provider besteht und ein Wechsel zu einem anderen Provider mit sehr hohem Aufwand verbunden ist. Außerdem behalten viele Kunden aus Sicherheitsgründen die Verwaltung ihrer Benutzer und die Einstellungen für die Verschlüsselungsstärke lieber in eigener Hand.

1.9 Intranet-VPN

Das VPN wird hierbei durch Erweiterungen der in lokalen Netzwerken (LAN) eingesetzten Switching-Technologien erzeugt. Die virtuellen Netzwerke werden auf der OSI-Schicht 2 abgebildet. Das LAN-Switching wurde ursprünglich eingeführt, um den Durchsatz in Shared-Media-Netzen wie dem Ethernet zu erhöhen, indem diese Netze in verschiedene, so genannte Collision Domains aufgeteilt werden. Die Switches verbinden diese Collision Domains, um die Pakete weiterzuleiten, die an Stationen in der gleichen Collision Domain in dem gleichen oder in einem anderen Switch adressiert sind.

Die in den LAN-Switches eingesetzte Technologie geht nun noch einen Schritt weiter und erlaubt es dem Netzwerkadministrator, eine logische Gruppe von Geräten im gesamten lokalen Netzwerk zu definieren, die dann ein virtuelles LAN (VLAN) bilden. Dieses VLAN kann sich auch über mehrere Switches hinweg erstrecken.

In der VLAN-Technologie werden keine speziellen Sicherheitstechnologien wie Verschlüsselung, Integritätssicherung oder Paketauthentifizierung eingesetzt. Es erfolgt lediglich eine Trennung des Transports der Datenpakete auf der OSI-Schicht 2, und statt auf Sicherheit wurde bei dieser Technologie mehr auf Performance abgezielt.

Die Erzeugung eines solchen virtuellen LAN kann je nach gewünschtem Einsatz auf verschiedene Weise erfolgen, woraus auch die Bezeichnung der jeweiligen Art des VLAN abgeleitet wird. Viele moderne Switch-Architekturen beherrschen alle Spielarten der VLAN-Technologie und können sogar routen. Ein so genannter Layer-3-Switch trifft seine Weiterleitungsentscheidungen extrem schnell und routet mit „Wire Speed“, mittlerweile auch im Gigabit-Bereich.

VLAN werden in letzter Zeit auch häufig benutzt, um Sprach- und Datennetze auf Layer 2 zu separieren.

Neben VLANs werden auch gelegentlich Verfahren auf der Schicht 3 benutzt, insbesondere IP-in-IP-Tunneling, unter Umständen sogar mit Verschlüsselung in besonders sensiblen Bereichen.

2 Anforderungen an VPN

Die Einsatzgebiete für virtuelle private Netzwerke sind sehr vielfältig. Je nach den gestellten Anforderungen an Sicherheit, Quality of Service sowie anderen Rahmenbedingungen kann man, entsprechend dem Angebot der Service Provider, die komplette Weitverkehrsinfrastruktur, die eigene Business-to-Business-Kommunikation (B2B) und den Remote Access als virtuelles privates Netzwerk aufbauen.

Bei der Auswahl der geeigneten Technologie muss man sehr genau untersuchen, welche Anforderungen an das VPN gestellt werden. In der Regel resultieren diese aus Sicherheitsbedürfnissen, gefolgt von Kostenaspekten, der Verfügbarkeit und – abhängig von den eingesetzten Applikationen – den benötigten Bandbreiten und tolerierbaren Verzögerungszeiten.

2.1 Sicherheit

Im Bereich der Datensicherheit gibt es eine ganze Reihe von Anforderungen, die sich in verschiedene Bereiche gliedern:

- Datenvertraulichkeit
- Schlüsselmanagement
- Paketauthentifizierung
- Datenintegrität
- Benutzerauthentifizierung
- Benutzerautorisierung
- Schutz vor Sabotage
- Schutz vor unerlaubtem Eindringen

2.1.1 Datenvertraulichkeit

Es muss sichergestellt werden, dass Unbefugte die Daten auf ihrem Weg durch das Internet nicht lesen können.

Vielfach wird auch gefordert, dass das interne Netzwerk mit seinen Verkehrsbeziehungen (Quell- und Zieladressen, Protokoll- und Portnummern) ebenfalls nicht ausgespäht werden kann. Dies wird im Allgemeinen durch die Verschlüsselung der Paketdaten erreicht. Falls die Verkehrsbeziehungen ebenfalls geschützt werden sollen, muss das originale Paket vollständig in den Datenbereich eines neuen Pakets eingepackt werden. Dies nennt man Tunneling.

Als Verschlüsselungsverfahren sollten unbedingt standardisierte, wohl bekannte und starke Verfahren eingesetzt werden. Meist gibt die eingesetzte VPN-Technologie aus

Gründen der Interoperabilität die Verfahren auch schon fest vor. In der Praxis werden wegen ihrer hohen Geschwindigkeit ausschließlich so genannte symmetrische Verfahren eingesetzt, bei denen Sender und Empfänger den gleichen Schlüssel zum Ver- und Entschlüsseln der Daten benötigen.

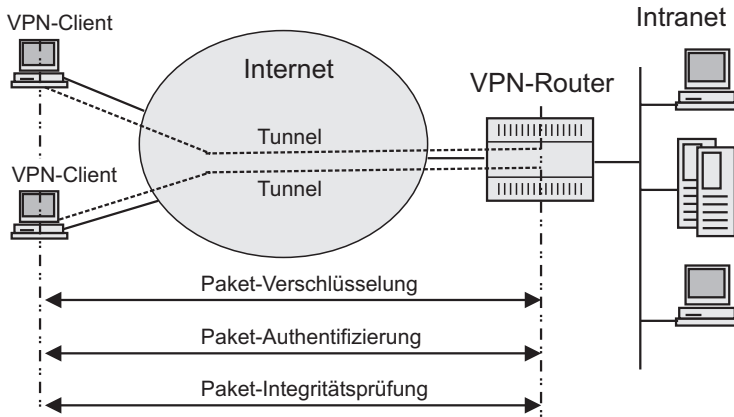


Abbildung 2.1: Beim Transport vertraulicher Informationen über das Internet müssen verschiedene Sicherheitsmaßnahmen getroffen werden.

Schlüsselmanagement

Um auf sicherem und vor allem automatischem Wege eine Verteilung von symmetrischen Schlüsseln zu ermöglichen, benötigt man ein zuverlässiges Schlüsselmanagement. Dessen Aufgabe besteht im Erzeugen aller benötigten Schlüssel zur Verschlüsselung, Integritätsprüfung und Authentifizierung und in deren Verteilung zu den richtigen Gegenstellen in einem VPN.

Gute Schlüssel, vor allem solche zur Datenverschlüsselung, haben eine relativ kurze Lebensdauer von meist einer Session oder wenigen Stunden und müssen deshalb sehr oft erzeugt und verteilt werden. Manuelle Verfahren scheiden aus diesem Grund, vor allem auch bei größeren Installationen, aus. Da Out-of-Band-Verfahren, also die Verteilung der Schlüssel über ein anderes Kommunikationsmedium, praktisch den doppelten Aufwand bei der Auslegung eines Netzes erfordern, scheiden diese ebenfalls in den meisten Fällen aus.

Die heute bekannten Verfahren zum Schlüsselmanagement basieren meist auf so genannten asymmetrischen Verfahren, bei denen zum Ver- und Entschlüsseln jeweils unterschiedliche Schlüssel verwendet werden, von denen einer, der öffentliche Schlüssel (Public Key), allgemein bekannt sein darf. Diese Verfahren nennt man daher auch Public-Key-Verfahren.

2.1.3 Paketauthentifizierung

Es muss garantiert werden, dass ankommende Pakete auch tatsächlich von dem authentischen Sender kommen und nicht von Dritten mit gefälschten Absenderadressen und neu berechneten Prüfsummen geschickt wurden.

Ähnlich wie bei einer Benutzer-Authentifizierung muss tatsächlich jedes ankommende Paket authentifiziert werden, was man durch symmetrische Schlüssel oder so genannte Pre-Shared Secrets, vertrauliche Daten, die nur dem Sender und dem Empfänger bekannt sind, erreicht. Aus Gründen der Geschwindigkeit und der Einfachheit kombiniert man dies meist mit Verfahren zur Prüfung der Datenintegrität.

Der Empfänger muss zuverlässig erkennen können, ob ein ankommendes Paket während des Transports verändert wurde.

Normale Prüfsummenverfahren reichen hierfür nicht aus, da auch ein Angreifer nach Änderung eines Datenpakets auch dessen Prüfsumme neu berechnen kann. Spezielle Verfahren auf Basis von symmetrischen Verschlüsselungsverfahren berechnen die Paketprüfsumme und fügen sie in das Paket mit ein. Die Schlüssel sind nur dem Sender und dem Empfänger bekannt. Ein Angreifer, der ein Paket ändern will, kann die Prüfsumme nicht korrekt berechnen.

Dies ist ganz wichtig bei Remote-Access-VPN. Ein Benutzer, der über ein VPN-Gateway Zugriff auf das Intranet verlangt, muss seine Identität möglichst zuverlässig nachweisen.

Der Grad dieser Zuverlässigkeit ist von Fall zu Fall verschieden, so dass ein guter VPN-Router eine Reihe unterschiedlich starker Authentifizierungsverfahren unterstützen muss. Die Abstufung reicht dabei von einfachen Passwortverfahren bis hin zur Verwendung von Tokenkarten oder digitalen Zertifikaten. In manchen Bereichen, zum Beispiel bei den Tokenkarten, gibt es keine verbindlichen Standards, so dass man hier immer proprietär ist. Für PKI (Public Key Infrastructure, eine Infrastruktur zum Management von öffentlichen Schlüsseln und digitalen Zertifikaten) gibt es eine Arbeitsgruppe innerhalb der IETF, die auch schon eine Reihe von Standards zur Verwendung von digitalen Zertifikaten in VPN hervorgebracht hat.

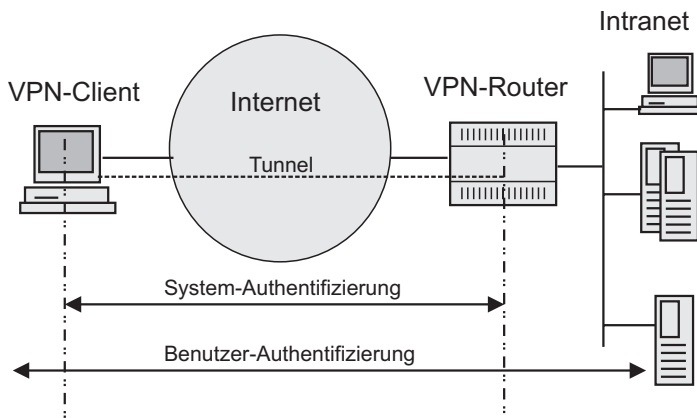


Abbildung 2.2: Neben dem System muss vor allem auch der jeweilige Benutzer authentifiziert werden.

Wird das VPN auch als Extranet verwendet, dann greifen auch externe Personen oder Organisationen, denen nur begrenzte Zugriffsrechte gewährt werden dürfen, auf Ressourcen des Unternehmensnetzes zu.

Die Autorisierung ist aber nur begrenzt ein Thema für Zugriffssysteme wie Router oder VPN-Gateways. Denn diese Systeme sind nicht in der Lage, auf Benutzer- oder Gruppenebene rollenbasierende Zugriffe auf Verzeichnisse, Datenbanktabellen oder Drucker zu steuern. Diese Funktionen können nur von den Systemen wahrgenommen werden, die diese Ressourcen auch verwalten, also von Betriebssystemen und Datenbankmanagementsystemen.

Zugriffssysteme müssen aber die Fähigkeit besitzen, auf der Ebene von Netzwerkprotokollen eine Filterung vorzunehmen. Im Fall des IP-Protokolls heißt dies, dass aufgrund von Host- oder Netzwerkadressen, Protokollnummern, Portnummern usw. Entscheidungen getroffen werden können, ob ein Paket weitergeleitet oder verworfen wird. Aber dies ist keine Filterung auf Benutzerebene mehr, sondern auf Netzwerkebene, da Benutzer normalerweise nicht an feste IP-Adressen gebunden sind.

Falls die Filterungen auch aufgrund von Informationen in höheren Ebenen des OSI-Referenzmodells vorgenommen werden sollen, spricht man von Firewalls. Diese wurden bisher meist als Übergangspunkt vom Intranet in das Internet verwendet, sind aber auch als Schutzsystem im Intranet-Extranet-Übergang einsetzbar. Firewalls bieten eine Filterung von Netzwerkpaketen bis hin zu Dateninhalten, Gateways auf Applikationsebene und Integrationsmöglichkeiten von externen Applikationen wie z.B. Virenschutzprogramme. Manche Router oder VPN-Gateways bieten als Option die Möglichkeit, Firewalls zu integrieren, jedoch ist es aus Gründen der Sicherheit und Performance oft besser, beide Funktionalitäten voneinander zu trennen.

Das VPN-Gateway soll vor Angriffen sicher sein, die darauf abzielen, seine Funktionalität zu beeinträchtigen oder es funktionsunfähig zu machen.

Es gibt eine Reihe mehr oder weniger subtiler Arten von Angriffen auf Internetsysteme, denen auch VPN-Gateways ausgesetzt sein können. Vor der plumpen Art einer solchen DoS-Attacke (Denial of Service, dabei wird verhindert, dass ein System seine Dienste erbringen kann), dem Überlasten einer Übertragungsstrecke oder einer Netzwerkschnittstelle, kann man sich nicht schützen, allerdings sind solche Angriffe mittlerweile sehr schnell zurückzuverfolgen, und der Angreifer bekommt in der Folge sehr viel Zeit, um über seine Untaten nachzudenken. Die Strafverfolgungsbehörden haben sich mittlerweile auf der ganzen Welt recht gut auf diese Art von Kriminalität eingestellt.

Subtilere Angriffe verschleiern, teilweise recht effizient, ihren eigentlichen Ursprung und schicken wenige, unverdächtig scheinende Pakete zu den Zielsystemen. Sie versuchen sie damit zu umfangreichen Aktivitäten zu bewegen, aufgrund derer ihnen für ihre eigentlichen Dienstleistungen immer weniger Ressourcen zur Verfügung stehen. So genannte DDoS-Angriffe (Distributed DoS) gehen noch einen Schritt weiter und legen ihren Angriff zweistufig an. Im ersten Schritt wird auf einer großen Anzahl von Rech-

nern unbemerkt ein Programm aufgespielt, das die eigentliche Attacke durchführen soll. Als Nächstes fangen alle diese Rechner zu einem bestimmten Zeitpunkt an, gleichzeitig mit unverdächtig scheinenden Paketen das Zielsystem regelrecht zu bombardieren. Die Zurückverfolgung des Angriffs ist sehr schwer, da er von Hunderten oder Tausenden weltweit verteilter Systeme auszugehen scheint, diese aber selbst auch nur „Opfer“ sind. Der wirkliche Urheber hat inzwischen genug Zeit gehabt, seine Spuren zu verwischen.

Ein VPN-Gateway muss verhindern, dass Unbefugte die Möglichkeit haben, über seine öffentlichen Schnittstellen in das Unternehmensnetzwerk zu gelangen.

Denn dies ist der direkte Weg zu den Informationen, die ein Angreifer ausspionieren oder manipulieren will. Der Hacker begnügt sich nicht mit dem, was über Netzwerkverbindungen übertragen wird, sondern er will Zugang zu den Systemen, auf denen die Daten gespeichert und verarbeitet werden. Ganz beliebte Systeme sind Authentifizierungssysteme, auf denen Passwörter und andere kritische Informationen gespeichert werden. Wer solch ein System „geknackt“ hat, dem steht ein ganzes Unternehmensnetzwerk offen.

Da VPN-Gateways in vielen Fällen die einzige Schnittstelle zwischen einem Intranet und dem Internet sind, auf das Millionen Unbekannte Zugriff haben, müssen hier besondere Maßnahmen zum Zugriffsschutz getroffen werden. Man unterscheidet dabei:

- Physische Sicherheit
- Schnittstellensicherheit
- Betriebssicherheit

Physische Sicherheit

Die Systeme müssen in sicheren Umgebungen betrieben werden. Die Sicherheit des Betriebsraums ist wesentlich, Security-Gateways dürfen nicht in Büros oder normalen EDV-Räumen betrieben werden, sondern nur in entsprechend sicheren Zonen. Sehr gute Systeme sind durch spezielle Maßnahmen gegen unerlaubtes Öffnen des Geräts abgesichert und erzeugen im Zugriffsfall einen entsprechenden Alarm im Netzwerkmanagementsystem.

Schnittstellensicherheit

Die Interfaces, die mit dem Internet verbunden sind, sollten spezielle Mechanismen zum Schutz gegen die verschiedenen Arten von Angriffen aufweisen. Als besonders geeignet erweisen sich so genannte gehärtete IP-Stacks, die eines großen Teils ihrer normalen Funktionalität beraubt und damit gegen eine ganze Reihe von Angriffen immun sind, da die meisten Angriffspunkte überhaupt nicht mehr vorhanden sind.

Als schlecht im Sinne von unsicher sind im Allgemeinen VPN-Systeme einzustufen, die als Programm oder Prozess auf nicht sicheren Betriebssystemen laufen. Trotz anders lautender Behauptungen kann die Sicherheit eines Programms nicht größer als die Sicherheit des Betriebssystems sein, auf dem es läuft. Insbesondere die IP-Stacks einiger Betriebssysteme sind beliebte Ziele von Angriffen verschiedenster Art.

Betriebssicherheit

Hier gilt es, dafür Sorge zu tragen, dass sich durch den Betrieb des VPN-Gateways keine Hintertüren für Angreifer öffnen. Da verschiedenen Studien zufolge der weitaus größte Teil von Angriffen von innen heraus erfolgt und nicht von Seiten des Internets, sind hier spezielle Maßnahmen nötig. Insbesondere sollte die Administration der Systeme im LAN verschlüsselt erfolgen, also zum Beispiel mit SSL, SSHv2 oder noch besser mit IPSec. Überhaupt sollte man sich ganz genau vergewissern, welche Zugriffsmöglichkeiten im privaten Netz gegeben sind und wie damit umzugehen ist. Ein cleverer Angreifer – und viele davon sind sehr clever – wird, sobald er sieht, dass ein VPN beispielsweise IPSec mit starker Verschlüsselung benutzt, sofort alle Gedanken an einen Angriff auf die IPSec-Verbindung selbst aufgeben und nach anderen Schwachstellen im System suchen. Und eben diese gilt es zu vermeiden.

2.2 Verfügbarkeit

Ein virtuelles privates Netzwerk soll traditionelle Weitverkehrs- oder Remote-Access-Lösungen ergänzen oder sogar ganz ersetzen. Dies bedeutet aber auch, dass ein VPN eine Verfügbarkeit bieten muss, die nicht unter der von herkömmlichen WAN-Infrastrukturen liegt. Die „alten“ Lösungen, die auf Standardfestverbindungen, Frame Relay und ISDN basieren, weisen in der Regel eine hohe Verfügbarkeit auf.

2.2.1 Die Verfügbarkeit von Wähl- und Festverbindungen

Traditioneller Remote Access wird über Wählverbindungen über das analoge oder digitale Fernsprechnetz, bei Bedarf auch über das Mobilfunknetz, aufgebaut und in einem Remote-Access-Konzentrator terminiert. Diese Netze bieten in der Regel Verfügbarkeitszeiten von 99,999%. Mit anderen Worten, das Netzwerk zwischen der ISDN-Karte im Notebook eines Außendienstmitarbeiters und dem Primärmultiplexanschluss seines Unternehmens fällt pro Jahr maximal 5,2 Minuten aus – wenn überhaupt.

Für viele Remote-Access-Infrastrukturen ist die so genannte „Niemals besetzt“-Eigenschaft eines Telefonnetzwerks ebenso wichtig. Haben Sie schon jemals am soeben abgehobenen Hörer eines am öffentlichen Fernsprechnetz angeschlossenen, funktionsfähigen Telefonapparats kein Freizeichen gehört? Vermutlich nicht. Wenn nun mehr Zugangskanäle in ein Unternehmensnetzwerk gelegt werden, als es potenzielle Benutzer gibt, kann man sich damit eine Remote-Access-Lösung mit tatsächlich garantiertem Zugriff aufbauen, da man selbst Einfluss auf die Auslegung seiner Systeme hat. Dies ist in vielen Fällen ein entscheidendes Kriterium, zum Beispiel beim Einsatz von Applikationen für Buchungs- oder Reservierungssysteme.

Das Gleiche gilt für die meisten Festverbindungen und anderen WAN-Services. In ihren Verträgen garantieren die Provider entsprechende Verfügbarkeiten. Die für Standardfestverbindungen eingesetzten Fernvermittlungssysteme der großen Carrier bieten Verfügbarkeiten ähnlich der von ISDN, und auch die Frame-Relay- und ATM-Technologie ist mittlerweile mit so genanntem „Carrier-Grade“-Equipment aufgebaut, das auch ein mittleres Erdbeben unbeschadet übersteht.

Die Verfügbarkeit von Internet-VPN

Diese Verfügbarkeiten muss ein virtuelles privates Netzwerk ebenfalls bieten können, soll es ein traditionell aufgebautes Netzwerk ergänzen oder gar ersetzen. Denn üblicherweise investiert man nicht in eine Technologie, die qualitativ schlechter ist als die aktuelle. Es sei denn, die neue Technologie bietet enorme Kostenvorteile. Dann ist man schon eher geneigt, einen Kompromiss zu machen oder sich zu überlegen, welche Verfügbarkeit denn wirklich benötigt wird.

Es gilt also, überhaupt erst einmal zu ermitteln, welche Verfügbarkeit tatsächlich benötigt wird. In sehr vielen Fällen arbeiten Unternehmen beispielsweise mit Festverbindungen nicht wegen ihrer 99,99% Verfügbarkeit, sondern einfach, weil es zum Zeitpunkt der Entscheidung keine wirklichen Alternativen gab. Um die tatsächlich notwendige Mindestverfügbarkeit zu ermitteln, muss man sich Klarheit verschaffen, welche Applikationen in welchem Maße das VPN benutzen werden und welche Implikationen Verbindungsabbrüche verursachen können. Es ist auch wichtig zu wissen, welcher Art der Datenverkehr ist und welches Zeitprofil er aufweist. Wird das VPN zum Beispiel nur während der Bürostunden benutzt, z.B. für Online-Anwendungen, kann man getrost eine nur halb so gute Verfügbarkeit akzeptieren, ohne eine Qualitätseinbuße zu haben. Statistisch gesehen fällt mehr als die Hälfte der Ausfallzeit in Zeiten, in denen das VPN ohnehin nicht benutzt wird.

Die Verfügbarkeit eines Internet-VPN kann man nicht allgemein beurteilen. Man muss dabei drei Fälle unterscheiden:

1. Die Verbindungen, auf denen das VPN basiert, gehen zu einem einzigen Provider.
2. Die Verbindungen, auf denen das VPN basiert, gehen zu zwei oder mehreren Providern, die miteinander kooperieren und entsprechende Durchleitungsverträge und/oder Service Level Agreements (SLA) abgeschlossen haben.
3. Die Verbindungen, auf denen das VPN basiert, gehen zu verschiedenen Providern, die keine Verträge miteinander abgeschlossen haben. Es ist nicht vorher bestimmbar, welchen Weg die Pakete nehmen und wie sie dort behandelt werden.

In den beiden ersten Fällen kann man mit dem oder den Service Providern geeignete Service Level Agreements abschließen, in denen neben anderen Eckwerten auch die Mindestverfügbarkeitszeit vertraglich geregelt ist. Hier werden mittlerweile in den meisten Fällen bereits Verfügbarkeitszeiten garantiert, die deutlich über denen der meisten lokalen Kundennetzwerke liegen. Im dritten Fall ist dies nicht möglich; man kann zwar mit jedem einzelnen Provider ein SLA vereinbaren, dieses gilt aber nur für seine eigenen Netze, und es ist keine durchgehende Verfügbarkeit garantiert.

Letztendlich mündet das Ganze wieder in eine Kostenkalkulation: Kosten die theoretisch möglichen Ausfälle mehr, als ich durch die neue VPN-Technologie einspare?

2.3 Quality of Service

Quality of Service, kurz QoS, die Dienstgütequalität spielt eine zunehmend wichtige Rolle in heutigen Netzen. Denn der momentane und in der Zukunft noch viel stärkere Trend geht in Richtung konvergente Netze, also ein einziges Netz für alle Arten der Kommunikation, sei es Sprache, Daten, interaktive Videokonferenzen oder Streaming. Solche Netze müssen mit den unterschiedlichsten Verkehrskategorien zurechtkommen. Stoßweiser Datenverkehr (Bursty Traffic) mit extrem unterschiedlichen Paketgrößen, relativ kontinuierliche Ströme aus kleinen Paketen für Sprache, die aber ohne große Verzögerungen übertragen werden, und viele andere Arten von Datenströmen teilen sich dabei ein Medium.

Im LAN, mit den auf zig GBit/s hochgerüsteten Infrastrukturen, mit virtuellem LAN, Ethernet-Priorisierung und Hardware-Queues, hat man das Thema bislang relativ gut in den Griff bekommen. Im WAN-Bereich war es bisher meist so, dass man verschiedene Netze für Sprache und Daten hatte, wobei insbesondere die Sprache über SDH-Infrastrukturen mit garantierter Bandbreite und minimaler, konstanter Verzögerung übertragen wurde. Selbst bei Daten hat man teilweise schon QoS eingesetzt, zum Beispiel bei Frame Relay oder ATM.

Ein VPN sollte diese Dienstgütequalität natürlich nicht, oder wenn, dann in einem für den Anwender erträglichen Maß, unterschreiten. Um zu verstehen, was ein VPN an Dienstgütequalität bereitstellen muss, sehen wir uns im Folgenden einmal die gewachsenen Anforderungen durch traditionelle LAN- und WAN-Technologien an.

Die Antwort, ob und in welchem Maß QoS im VPN möglich ist, finden Sie in Kapitel 9, Quality of Service in VPN.

2.3.1 QoS in Sprachnetzen

In Sprachnetzen braucht man sich über Sprachqualität keine Gedanken zu machen. Sie ist, zumindest in voll-digitalen (ISDN, SHD) Netzen, einfach da. Die Sprachqualität ist sehr hoch, es gibt keine Verzögerung, keine Verzögerungsvarianz, kein übermäßiges Rauschen, keinen Bandbreitenengpass und praktisch keine Ausfälle. Die Netze haben in der Regel, zumindest hierzulande, eine Verfügbarkeit deutlich jenseits der 99,999%. Die Technologie sorgt dafür, dass, wenn eine Verbindung zustande gekommen ist, die ganze Zeit die volle Bandbreite (64 KBit/s) für das Gespräch zur Verfügung steht.

Natürlich rede ich hier vom Festnetz. In GSM-Netzen trifft hiervon praktisch gar nichts mehr zu: Verbindungsabbrüche, Echos, Rauschen, miese Sprachqualität, „Netzabschnitt besetzt“ und all dies ist man hier gewöhnt. Das ist der Preis der Mobilität.

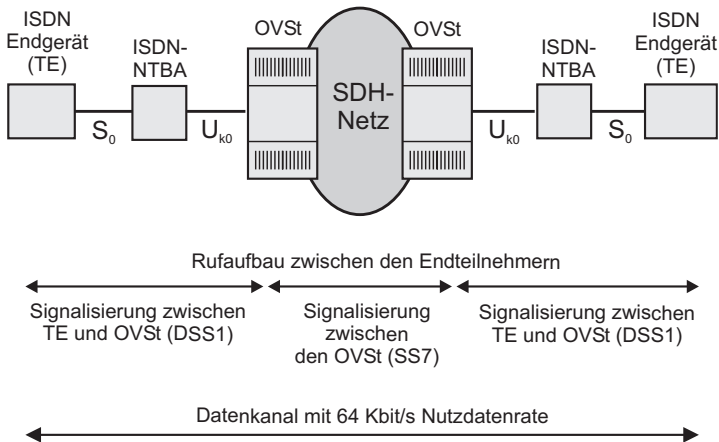


Abbildung 2.3: In digitalen Sprachnetzen verfügt eine Verbindung über garantierte Bandbreiten.

2.3.2 QoS im LAN

Im LAN gibt es nur noch einziges Protokoll, nämlich Ethernet. Das hat zwei Seiten, eine gute und eine schlechte. Die gute Seite ist die, dass es für den Anwender relativ überschaubar geworden ist, was im LAN eingesetzt wird. Die schlechte Seite ist die, dass Ethernet und QoS eigentlich zwei verschiedene Welten sind. Ethernet ist ein Shared Medium, auf das im Prinzip jeder zu jeder Zeit zugreifen darf und anschließend prüft, ob die Übertragung des von ihm auf das Medium gelegten Frames erfolgreich war. Wenn gleichzeitig jemand anderes auf die gleiche Idee kam, ging es für beide schief, und beide versuchen es nach einiger Zeit noch einmal. Damit bekommt man bei einer gewissen Anzahl von Nutzern weder Bandbreite (in der Regel maximal 30% der theoretisch möglichen) noch kann man irgendwelche QoS-Kriterien wie Verzögerung oder Mindestbandbreiten garantieren. Aber Ethernet hat sich weiterentwickelt.

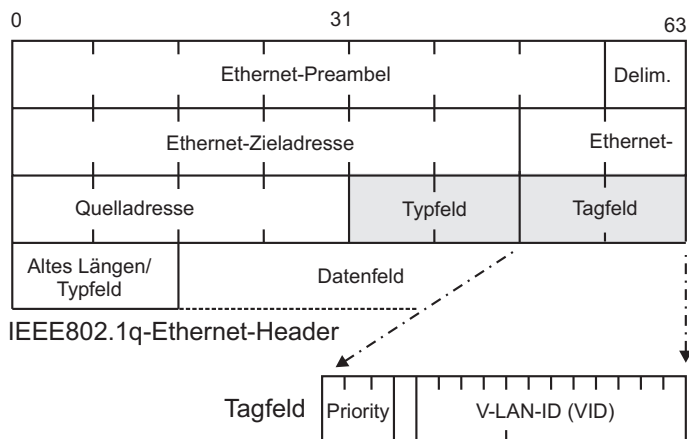


Abbildung 2.4: Neben VLAN-Informationen verfügt der erweiterte Ethernet-Header auch über ein Feld für QoS (Priority).

Aus dem ursprünglichen „gelben Kabel“ mit 5 MBit/s halbduplex sind 10 GBit/s voll-duplex geworden, durch Switching und virtuelle LANs (VLAN) hat man Kollisionen und Broadcasts reduzieren können. Somit haben heutzutage fast alle Benutzer im LAN einen exklusiven Switch-Port mit 100 MBit/s zur Verfügung.

Aber nicht nur im Bereich Geschwindigkeit hat sich bei Ethernet einiges getan. Neue Standards wie der IEEE 802.1/p bieten Möglichkeiten, Ethernet-Frames zu klassifizieren und in Switchen oder auf Ethernet-Ports mit unterschiedlicher Priorität weiterzuleiten. Zu diesem Zweck wurde der Ethernet-Header um 4 Byte, den so genannten Tag, erweitert. Dieser Tag hat mehrere Funktionen: Er beinhaltet vor allem eine VLAN-Nummer (vid, VLAN-ID) nach IEEE 802.1Q und eine achtstufige Prioritätsmarkierung (user_priority) nach IEEE 802.1p. Weiterhin werden noch der Typ des Tags (z.Zt. fest 0x8100) und die Art der Bitordnung angegeben.

Die Auswertung der Prioritätsbits obliegt der Netzwerk-Policy und hängt nicht zuletzt von der eingesetzten Hardware ab, insbesondere wie viele Warteschlangen pro Port unterstützt werden.

Und wie sieht es im WAN aus? Bescheiden, muss die korrekte Antwort heißen, denn während sich ein einziger Benutzer im LAN eines exklusiven 100 MBit/s-Ports erfreuen kann, wird der Standort, in dem er und seine 20 Kollegen arbeiten, oft mit 2 MBit/s, teilweise sogar mit noch deutlich weniger Bandbreite, mit anderen Lokationen verbunden. Das ist eine ziemliche Diskrepanz.

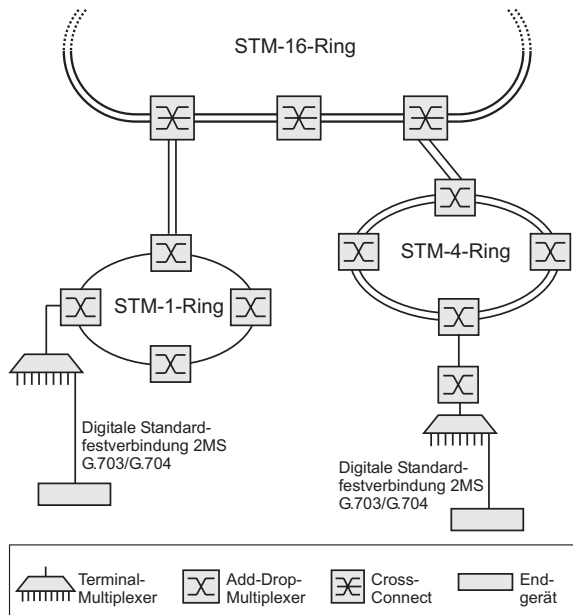


Abbildung 2.5: Digitale Standardfestverbindungen bekommen fest geschaltete Verbindungen innerhalb einer SDH-Infrastruktur.

Allerdings bieten einige WAN-Protokolle etwas Linderung an, denn sie stellen einige QoS-Möglichkeiten zur Verfügung. Sehen wir uns einmal die drei verbreitetsten Technologien an:

Digitale Standardfestverbindungen

Bei digitalen Festverbindungen (z.B. digitale StFV 2 MBit/s) liegen die Dinge recht einfach. Hier werden in der Regel bittransparente Verbindungen über SDH-Netzwerke geschaltet mit fester Bitrate, z.B. 2,048 MBit/s mit minimaler Verzögerung und praktisch keinem Jitter. Die Verfügbarkeit ist ebenfalls sehr hoch.

Digitale Standardfestverbindungen sind gewissermaßen ein Abfallprodukt der digitalen Telefonnetze. Je nach Land und Carrier werden verschiedene Geschwindigkeiten angeboten, üblich sind 64 KBit/s, 128 KBit/s und 2 MBit/s, teilweise auch 34 MBit/s. Manche Anbieter bieten auch maßgeschneiderte Geschwindigkeiten, meist in Abstufungen von 64 KBit/s an ($n \times 64$ KBit/s) oder einigen Zwischenwerten (256, 512, usw.) bis hin zu 34-MBit/s-Zugängen, die in 2×2 MBit/s Schritten angepasst werden können.

Typ	Nutzbare Geschwindigkeit	Fehler-rate	Mittlere Verfüg.	Schnittstellen (Elektrisch und Rahmenstruktur)
Digital 64S	64 KBit/s	10^{-7}	98,5%	ITU-T Empf. I.430 (SO-FVM)
Digital 64S2	128 KBit/s	10^{-7}	98,5%	ITU-T Empf. I.430 (SO-FVM)
Digital 2MU	2,048 MBit/s	10^{-7}	98,5%	ITU-T G.703
Digital 2MS	1,984 MBit/s	10^{-7}	98,5%	ITU-T G.703 und G.704
Digital 34M	34,368 MBit/s	10^{-7}	98,5%	ITU-T G.703
Digital 155M	149,760 MBit/s	10^{-7}	98,5%	ITU-T G.703, G.708 und G.709 (SDH STM-1 mit VC4)

Tabelle 2.1: Einige Festverbindungen der Deutschen Telekom

Digitale Standardfestverbindungen haben keine eingebauten QoS-Funktionen, sie übertragen bittransparent mit einer festen Datenrate.

Frame Relay

Frame Relay (FR) basiert auf der nicht mehr zeitgemäßen, relativ langsamen und durch ihren hohen Verarbeitungsaufwand ineffizienten X.25-Technologie. Diese wurde zu einer Zeit entwickelt, als die Medien zur Datenübertragung noch sehr fehleranfällig waren und die Applikationen keine besonders großen Anforderungen an Bandbreiten und andere Qualitätskriterien stellten. Frame Relay wird üblicherweise für niedrige bis mittlere Geschwindigkeiten bis zu 45 MBit/s eingesetzt, falls bestimmte Anforderungen an QoS erfüllt werden müssen.

Frame Relay enthält eine elementare Link-Layer-Funktionalität, um die Pakete, die hier als Frames bezeichnet werden und Nutzdaten bis zu jeweils 4096 Byte aufnehmen können, zwischen zwei Endpunkten zu transportieren. Dies geschieht im Allgemeinen analog zum X.25-Protokoll mit dauerhaft konfigurierten, virtuellen Verbindungen (PVC, Permanent Virtual Circuit).

Mittlerweile sind auch virtuelle Wählverbindungen (SVC, Switched Virtual Circuit) im Frame-Relay-Protokoll spezifiziert, jedoch werden sie relativ selten angeboten, da die Netz- und Systembelastung für die Signalisierung zum Auf- und Abbau der Verbindungen im Vergleich zu permanenten Verbindungen sehr hoch ist.

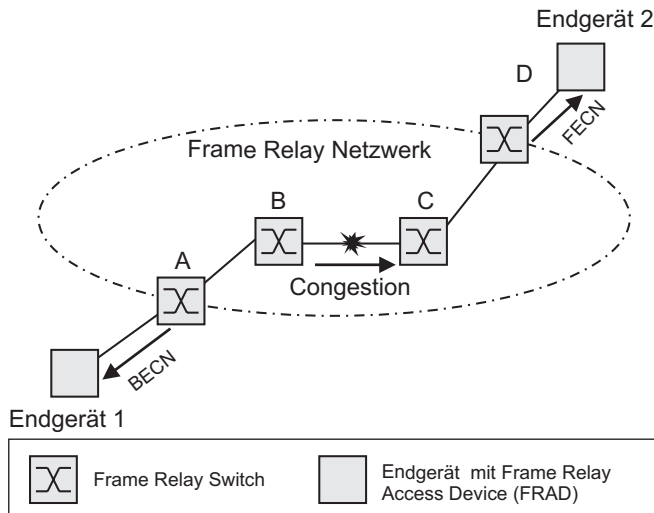


Abbildung 2.6: Ein Frame-Relay-Netz informiert die Endsysteme über Überlastsituationen innerhalb einer virtuellen Verbindung.

Frame Relay bietet bereits einige rudimentäre Funktionen zur Flusssteuerung bzw. zu dem Umgang mit Überlastsituationen (Congestion). Mit bestimmten Markierungen (Flags) im Frame-Header können Frame-Relay-Switches in beide Richtungen, also sowohl Sender (Backward Explicit Congestion Notification, BECN) als auch Empfänger (Forward Explicit Congestion Notification, FECN), signalisieren, dass sie sich in einer Überlastsituation befinden.

Wie in Abbildung 2.6 zu sehen, signalisiert ein Frame-Relay-Netz den beiden Endgeräten die aktuelle Überlast zwischen den Switches B und C. Frames für das Endgerät 2 treffen bei diesem mit gesetztem FECN-Bit ein. In den Frames auf dem Rückweg zum Endgerät 1 wird das BECN-Bit gesetzt. Falls jedoch in dieser Richtung kein Datenverkehr existiert, kann das Frame-Relay-Netzwerk auch spezielle Management-Frames erzeugen, die nichts weiter tun, als dem Endgerät die Überlastsituation mitzuteilen. Bestimmte Reaktionen der Endgeräte auf Überlast sind in den Standards des Frame-Relay-Forums zum FR-Protokoll nicht verbindlich vorgeschrieben, dies wird den höheren Übertragungs- oder Verbindungsprotokollen wie TCP oder SNA überlassen.

Zusätzlich können abgehende Frames noch eine Prioritätsmarkierung erhalten (Discard Eligible, DE), die Auskunft darüber gibt, ob der Frame verworfen werden kann, um drohende Überlastsituationen zu vermeiden. Das DE-Bit kann auch, in Verbindung mit speziellen Parametern zur Konfiguration der PVCs oder SVCs, zur Vereinbarung und Umsetzung von Dienstqualität in Service Level Agreements benutzt werden. Das funktioniert, indem der Provider bestimmte Werte wie CIR (Committed Information Rate, die garantierte Übertragungsgeschwindigkeit), EIR (Excess Information Rate, die erlaubte Überschreitung der CIR) und die Bandbreite der Zugangsleitung, oft eine digitale Standardfestverbindung, vereinbart.

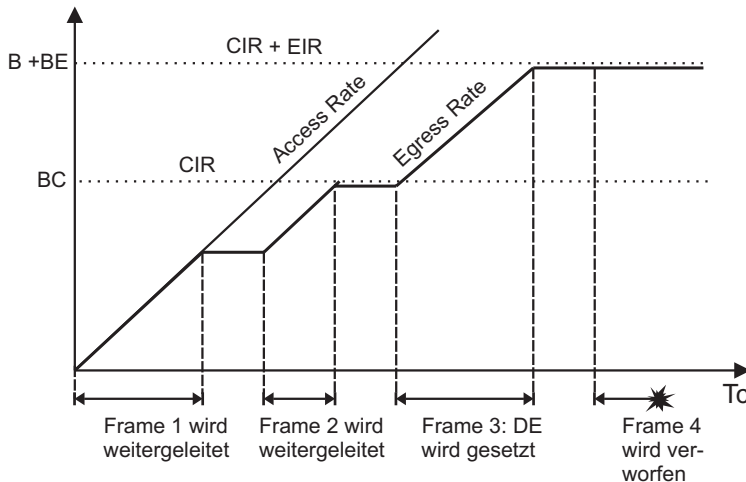


Abbildung 2.7: Die Flusssteuerung von Frame Relay

In Abbildung 2.7 ist das Zusammenspiel dieser Parameter für ausgehende Frames illustriert. Die Access Rate ist die Geschwindigkeit, mit der die Frames dem Frame-Relay-Netzwerk angeboten werden. Die Egress Rate ist die Geschwindigkeit, mit der die Frames weitergeleitet werden. Wird die CIR nicht überschritten (Frames 1 und 2), so werden die Frames unverändert weitergeleitet. Überschreitet die Access Rate (Frame 3) die CIR, ist aber noch innerhalb der EIR, so wird im Frame das DE-Bit gesetzt, der Frame wird also im Fall von Lastproblemen bevorzugt verworfen. Wird jedoch auch die EIR überschritten, so wird der Frame sofort verworfen.

ATM

ATM, der Asynchronous Transfer Mode, bietet von den drei hier betrachteten Funktionen noch die fortschrittlichsten QoS-Funktionen. ATM wurde, wie Sie vermutlich wissen, primär zum Transport von Sprache und Video entwickelt, aber auch zum Transport anderer Daten. Aus diesem Grund wurde von Anfang an auch auf das Thema Quality of Service geachtet. Die Technik wurde auf hohe Geschwindigkeit und geringe Verzögerungen getrimmt. ATM definiert fünf unterschiedliche Service-Klassen, die in Tabelle 2.2 beschrieben sind. Diese unterschiedlichen Klassen sind für verschiedene Arten von Applikationen wie Video, interaktive Sprachübertragung (Telefonie) oder Datenpaketübertragung geeignet.

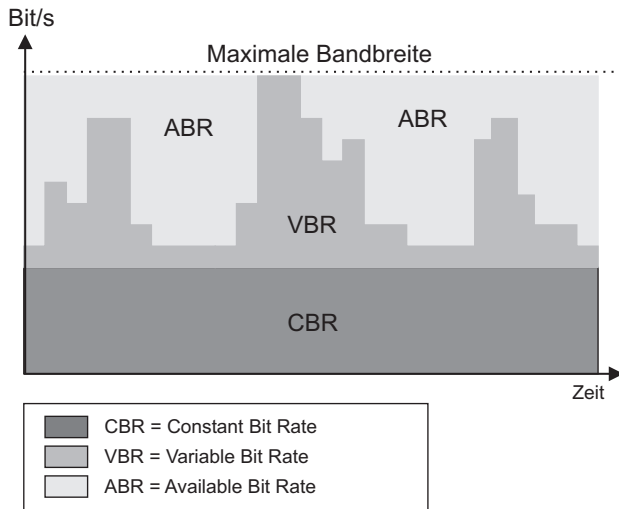


Abbildung 2.8: Die Verteilung verschiedener ATM-Dienste innerhalb der verfügbaren Datenrate

ATM bietet, trotz einer gewissen Komplexität (die ohne LAN-Emulation aber noch erträglich ist), eine Reihe von Vorteilen gegenüber anderen Protokollen. Einer dieser Vorteile ist die Tatsache, dass auf ATM-Ebene bereits fünf verschiedene Service-Klassen definiert sind:

ABR (Available Bit Rate) ist eine Klasse, die Best-Effort-Transport mit Rückkopplung über Ressource-Management(RM)-Zellen zum Sender zur Verfügung stellt. Falls die Quelle auf die RM-Zellen adäquat reagiert, kann mit ABR eine sehr niedrige Zellenverlustrate (CLR, Cell Loss Ratio) garantiert werden.

nrt-VBR steht für „non real-time Variable Bit Rate“. Vom Verhalten her entspricht diese Klasse in etwa Frame Relay und bietet einige Garantien hinsichtlich Delay, Zellenverlust und Latenz.

rt-VBR ist im Gegensatz zu nrt-VBR für Echtzeitanwendungen gedacht, beispielsweise komprimierte Sprache oder Video. Diese Klasse kann Werte wie maximale Verzögerung oder CLR garantieren.

UBR, Unspecified Bit Rate, ist die in ATM-Netzen am meisten eingesetzte Service-Klasse, leider garantiert sie fast gar nichts an Qualitätsmerkmalen. Sie kommt dem normalen „Best-Effort“-Verhalten von IP sehr nahe, lediglich die PCR (Peak Cell Rate) kann festgelegt werden.

CBR (Constant Bit Rate) ist ebenfalls für Echtzeitanwendungen geeignet und bietet minimale Verzögerung und Jitter für unkomprimierte Sprach- und Videoanwendungen. CBR ist mit normalen Telefonnetzen vergleichbar, diese Klasse garantiert QoS, indem in ihr Werte wie beispielsweise PCR, CDV (Cell Delay Variation, Jitter) oder CLR festgelegt werden können.

Mit Hilfe dieser Service-Klassen kann man PVCs oder SVCs passend an seine Anforderungen durch bestimmte Applikationen konfigurieren bzw. durch einen Provider konfigurieren lassen.

Dienstklasse	Parameter	Beispielanwendung
CBR	PCR (Peak Cell Rate), CTD (Cell Transfer Delay), CDV (Cell Delay Variation), CLR (Cell Loss Ration)	Sprachvermittlung
rt-VBR	PCR, CTD, CDV, CLR, SCR (Sustained Cell Rate)	Voice/Video over Packet
nrt-VBR	PCR, SCR, CLR	Frame Relay Transport
ABR	PCR, MCR (Minimum Cell Rate)	Daten (IP)
UBR	Keine	Daten (IP)

Tabelle 2.2: Die ATM-Dienstklassen und die dazugehörigen Parameter

Obwohl ATM zwischen verschiedenen ATM-Switches extrem kleine Zellen (53 Byte) überträgt, ist es verbindungsorientiert und basiert auf virtuellen Verbindungen (VC, Virtual Circuit). Es werden entweder feste virtuelle Verbindungen (PVC, Permanent Virtual Circuit) durch ein ATM-Netz konfiguriert, oder sie werden für die Dauer einer Verbindung geschaltet (SVC, Switched Virtual Circuit) und anschließend wieder deaktiviert. Im Internet, in dem ATM vor allem in den Backbones der Service Provider und Carrier eingesetzt wird, werden vor allem PVCs eingesetzt. ATM bietet die Möglichkeit, die Konfiguration eines ATM-Netzwerks zu vereinfachen, indem man verschiedene VCs zu einem virtuellen Pfad (Virtual Path, VP) zusammenfasst.

2.3.4 QoS in IP

Die Überschrift ist nicht ganz korrekt, denn auf IP-Ebene kann es eigentlich gar keine ausführenden QoS-Funktionen geben. Das tut es auch nicht, aber IP besitzt Möglichkeiten, QoS-Informationen, die es von den Applikationen bekommen hat, an Layer-2-Protokolle wie Ethernet, Frame Relay oder ATM zu signalisieren.

Heute hat sich in IP die Philosophie des klassenbasierenden QoS durchgesetzt. Hierfür gibt es etliche RFCs, die sich mit dem Thema Differentiated Services (DiffServ) beschäftigen. Die Idee hinter DiffServ ist einfach die, dass IP-Pakete in eine überschaubare Anzahl verschiedener Qualitätsklassen eingeteilt und entsprechend ihrer Klasse und nicht verbindungsabhängig behandelt werden. Damit hat man einen auch in großen Netzen skalierbaren QoS-Mechanismus gefunden. Aber auch DiffServ schafft keine zusätzliche Bandbreite, sondern verteilt nur die vorhandene – oder wie es ein Kollege von Juniper Networks einmal formuliert hat: „Auch DiffServ kann die Geschwindigkeit des Lichtes nicht erhöhen.“

DiffServ benutzt das TOS-Feld im IP-Header als so genannten DSCP (DiffServ Code Point), um entsprechende Informationen über die Zugehörigkeit eines Pakets zu einer bestimmten Qualitätsklasse aufzunehmen. Weiterhin sind in DiffServ noch verschiedene Kategorien (PHB, Per Hop Behaviour) festgelegt, die bestimmen, wie Router, Switches oder Gateways mit Paketen bestimmter Klassen zu verfahren haben.

Weitere Details zu DiffServ finden Sie in Kapitel 9 (QoS in VPN).

Warum stellt sich diese Frage eigentlich, wenn es geeignete Netzwerktechnologien wie ATM oder Frame Relay gibt, die entsprechende QoS-Mechanismen haben? Ganz einfach weil ein VPN gerade diese Strukturen, die nebenbei auch noch klassische Layer-2-VPNs sind, nicht einsetzt, sondern mit möglichst günstigen Access-Technologien den nächsten POP eines Internet Services Providers erreichen und dann über das Internet kostengünstig kommunizieren will.

Und genau das sind die beiden Knackpunkte: erstens die Access-Technologie und zweitens das Internet selbst, das die Pakete nach Best-Effort-Manier überträgt – ohne sich in irgendeiner Weise um QoS zu kümmern. Und Access-Technologien wie xDSL, Festverbindungen oder drahtlose Medien kennen ebenfalls keinen QoS.

Was kann man denn tun, wenn man trotzdem ein konvergentes Netz hat und ein VPN plant oder umgekehrt? Nun, man sollte einfach strukturiert an die Sache herangehen und nicht in voreilendem Gehorsam gleich sagen: „Das geht bestimmt nicht, also versuche ich es gar nicht.“

Man muss einfach zuerst seine Anforderungen klar definieren. Man hat eine gewisse Netzstruktur und darin bestimmte Anwendungen, die bestimmte Qualitätskriterien im Netz bedingen, oder es kommen Anwendungen, z.B. VoIP, hinzu, die neue Anforderungen stellen. Daraus kann man die Mindestanforderungen ableiten, die ein Netz bzw. in unserem Fall auch ein VPN erfüllen muss, damit die Anwendungen funktionieren.

Und manche Anforderungen muss man auch manchmal kritisch hinterfragen. Gerade beim Thema konvergente Netze höre ich zuweilen, dass die Sprachqualität nicht schlechter werden dürfe als das, was man vom Festnetz oder von der TK-Anlage gewohnt sei, ansonsten gäbe es keine Akzeptanz seitens der Benutzer. Dann lautet meine augenzwinkernde Frage: Telefonieren Ihre mobilen Mitarbeiter denn nicht mit dem Handy? Doch, und welche Qualitätseinbußen gegenüber dem Festnetz werden da ertragen!

Wenn man nun weiß, was man mindestens will, muss man schauen, was an Technologie zur Verfügung steht, sowohl an Equipment als auch an Infrastruktur. Die Carrier und Provider schlafen auch nicht und wissen ganz genau, dass sie angesichts der Entwicklungen in Enterprise-Netzen auch bestimmte QoS-Services anbieten müssen.

Falls man, was ich schon oft bei weltweiten VPNs erlebt habe, in bestimmten Regionen die Internetstruktur so verheerend ist, dass Verzögerungen von mehreren Sekunden das statistische Mittel sind, dann gibt es auch noch Alternativen. Eine ist zum Beispiel die Kombination mit IP-VPN und Layer-2-VPN auf Basis von Frame Relay in einem VPN-Router.

Ein wichtiges Kriterium ist auch die Auswahl der VPN-Technologie. Wer zum Beispiel DiffServ einsetzen möchte, der kann von Protokollen wie SSL, TLS, L2TP oder L2TP/IPSec gleich die Finger lassen, denn diese Protokolle verhindern zuverlässig das Übertragen des DSCP oder generell die Signalisierung von QoS-Informationen an das äußere IP-Paket. Einzig IPSec ist hierzu in der Lage, sofern im Inneren eines IPSec-Pakets nicht L2TP das Ganze bereits im Vorfeld blockiert hat.

Skalierbarkeit und Migrationsfähigkeit

Wenn Sie heute über den Einsatz eines VPN nachdenken, dann sollten Sie auch bereits an übermorgen denken. Langfristige Voraussagen sind aber ziemlich schwierig und treffen manchmal auch nicht zu, also strebt man eine möglichst hohe Offenheit seines Systems an – Offenheit sowohl in Hinblick auf die Einhaltung von Standards, um gegebenenfalls einen Hersteller wechseln zu können, als auch in Hinblick auf die Modularität und Erweiterbarkeit der Systemkomponenten.

„Think big – start small“

Mit diesem Leitsatz liegt man genau richtig, denn für VPN gelten die gleichen Maßstäbe hinsichtlich Skalierbarkeit und Migrationsfähigkeit wie für andere Netzwerktechnologien auch. Während der Planungsphase sollte man sich bereits Gedanken machen, wie das Netz in der Zukunft aussehen kann. Denn sehr viele Rahmenbedingungen sind stetiger Änderung unterworfen, wie zum Beispiel die Zahl der Benutzer, benötigte Bandbreiten und Qualitätsmerkmale durch neue Applikationen, rechtliche Gesichtspunkte und neue Geschäftsfelder oder Akquisitionen.

Beim Thema Skalierbarkeit muss man aber unbedingt zwischen Leistung und Sicherheit differenzieren. Denn die Sicherheitsstrategie eines Unternehmens richtet sich nach völlig anderen Kriterien. Ein VPN-Gateway in einer kleinen Zweigstelle benötigt beispielsweise keinen so hohen Datendurchsatz wie seine Gegenstelle in der Unternehmenszentrale, auf der Hunderte von Verbindungen konzentriert werden, jedoch muss es den gleichen Sicherheitsanforderungen genügen. Hier gilt es, sehr genau zu prüfen, ob Endgeräte für kleine Außenstellen, die vom Systemdesign her für eine kleine Übertragungsbandbreite ausgelegt sind, diese auch dann noch bieten, wenn eine starke Verschlüsselung eingesetzt wird.

Ein Faktor, der sich in den meisten Netzwerken laufend ändert, ist die steigende Zahl von mobilen Anwendern und Heimbüros. Hier sollte, wie es sich in der Praxis gezeigt hat, von vornherein mit einer entsprechend großen Zahl kalkuliert werden.

Die Skalierbarkeit im Bereich der Systemleistung ist ein ganz wesentliches Entscheidungskriterium bei der Planung eines VPN. Meist sind Standorte verschiedener Größe, Heimbüros und mobile Mitarbeiter mit der notwendigen Technologie auszustatten. Je nach Einsatzgebiet sind unterschiedliche Datendurchsätze und Anschlusstechnologien notwendig, vom redundanten VPN-Router bis hinunter zur VPN-Client-Software für PCs. Idealerweise sollen diese verschiedenen VPN-Systeme aber auch eine möglichst einheitliche Konfigurations- und Managementoberfläche bieten.

Integration in existierende Netze

Selten bekommt ein Netzwerkplaner die Gelegenheit, ein VPN auf der grünen Wiese zu planen. Er hat viel häufiger die Aufgabe, ein VPN in bestehende Infrastrukturen zu integrieren. Diese Infrastrukturen sind bestehende lokale Netze, Weitverkehrsnetze und Remote-Dienste sowie die dazugehörigen Management-, Accounting- und Überwachungssysteme. Das zunehmende Verlangen vieler Unternehmen nach Kostentranspa-

renz auch im Netzwerkbereich führt zu nutzer- oder kostenstellenbezogenen Abrechnungssystemen, die vor allem in Verbindung mit WAN- und Remote-Access-Diensten eingesetzt werden, da diese durch die zusätzlich an die Carrier zu entrichtenden Gebühren sehr kostenintensiv sind.

2.5.1 Management

Insbesondere sind bei der Integration der VPN-Komponenten die bereits vorhandenen Managementsysteme zur Konfiguration und Überwachung zu unterstützen, die auch zu diesem Zweck für traditionelles Netzwerkequipment eingesetzt werden.

Die wichtigsten Funktionalitäten sind die zur Konfiguration, zur Überwachung und zur Abrechnung der Netzwerkdienste.

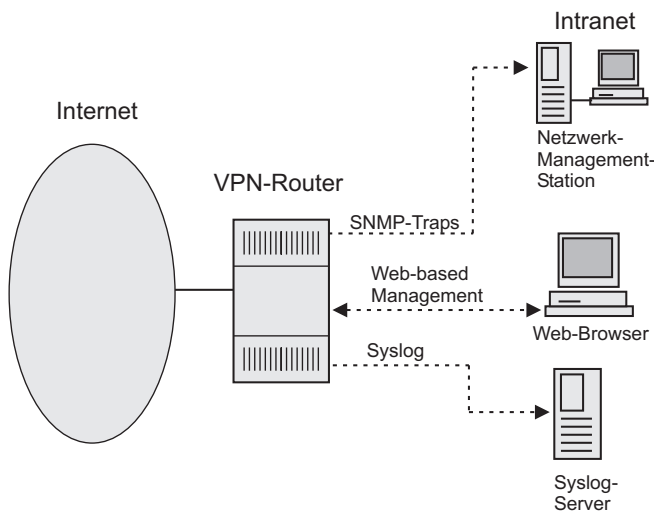


Abbildung 2.9: VPN-Komponenten müssen sich in existierende Managementsysteme integrieren lassen.

Konfiguration

Die Konfiguration von Netzwerkkomponenten erfolgt oft über SNMP (Simple Network Management Protocol). Dieses Protokoll wurde entwickelt, um eine problemlose Interoperabilität zwischen Netzwerkmanagementprodukten unterschiedlicher Hersteller zu ermöglichen. Eine Netzwerkmanagementstation, meist eine Grafik-Workstation, dient als zentrales Element zur Steuerung und Überwachung, während auf den Netzwerkkomponenten selbst so genannte SNMP-Agents residieren. Diese Agents kommunizieren über das Netzwerk mit der Managementstation, um Konfigurationsparameter zu setzen oder abzufragen. Eine weitere wichtige Funktionalität der Agents ist ihre Fähigkeit, bei kritischen Zuständen auf dem Netzwerkgerät einen so genannten SNMP-Trap zu erzeugen, der unmittelbar zur Managementstation geschickt wird, um diese über das aufgetretene Problem zu informieren.

Die Kommunikation der verschiedenen SNM-Komponenten erfolgt über UDP-Pakete, in denen verschiedene Kommandotypen zum Setzen und Lesen von MIB-Objekten oder -Variablen eingekapselt werden. Eine MIB (Management Information Base) ist eine Datenstruktur, welche die Summe aller managebaren Objekte der Netzwerkkomponenten darstellt. Leider ist die Sicherheit, wenn man es überhaupt so nennen kann, von SNMPv1 und SNMPv2 auf einem nicht allzu hohen Stand, so dass die Kommunikation mit einfachsten Methoden aufgezeichnet oder unbefugt verändert werden kann. Aus diesem Grund werden sicherheitskritische Systeme im Allgemeinen nicht über SNMP konfiguriert. Auch die Überwachung mit SNMP beschränkt man auf nicht sicherheitskritische Parameter wie Zähler für übertragene Bytes oder Prüfsummenfehler und Ähnliches.

Bei sicherheitskritischen Netzwerkkomponenten benutzt man andere Protokolle zu deren Konfiguration. Neben herstellerspezifischen, oft nicht offen gelegten Protokollen haben sich hier das SSL-Protokoll (Secure Socket Layer), Secure Shell (ssh) und IP-Security (IPSec) durchgesetzt. Diese Protokolle sind standardisiert und mittlerweile weit verbreitet.

Die Anforderungen an das Management, die aus dem hier Gesagten resultieren, sind genau genommen ein Kompromiss aus dem Wunsch nach reibungsloser Integration in bestehende SNMP-basierte Managementsysteme und der Forderung nach ausreichender Sicherheit. Es kann also auf einem VPN-Gerät durchaus SNMP unterstützt werden, jedoch mit der Einschränkung, dass die Konfiguration des Geräts abweichend vom Protokoll auf keinen Fall mit SNMP-Set-Befehlen durchgeführt werden darf, sondern über ein sicheres Protokoll erfolgen muss.

In letzter Zeit hat sich das so genannte webbasierte Management immer weiter verbreitet. Hierbei greift man über einen Webbrowser direkt auf die Netzwerkgeräte zu, um sie zu konfigurieren oder abzufragen. Der Vorteil dieser Methode besteht darin, mit dem Webbrowser eine grafische Benutzeroberfläche zu haben, mit der mittlerweile fast jeder umgehen kann und die praktisch auf jedem Arbeitsplatzrechner installiert ist. Ein weiterer Pluspunkt ist, dass neue Funktionen auf dem Gerät auch gleich in der grafischen Oberfläche verfügbar sind. In traditionellen SNMP-Umgebungen mit speziellen Netzwerkmanagement-Applikationen müssen die Funktionen dem Programm erst bekannt gemacht werden, bevor man sie benutzen kann. Vor allem mit grafischen Darstellungen, insbesondere in heterogenen Umgebungen, tun sich leider die meisten SNMP-Managementsysteme und die Gerätehersteller etwas schwer. Das Web-Management kann auch in Teilen kompatibel zu SNMP sein, indem der Browser intern die MIB dieses Geräts modifiziert oder abfragt und indem SNMP-Traps verwendet werden, um Alarmierungen abzusetzen.

Allerdings ist das HTTP (Hyper Text Transfer Protocol), genau wie SNMP, alles andere als sicher: Man kann es relativ problemlos mitlesen oder verändern. Die Lösung dieses Sicherheitsproblems ist auch hier die Verwendung des SSL-Protokolls, das speziell zur sicheren Browser-Server-Kommunikation entwickelt wurde. Bei Einsatz und Kombination der richtigen, sicheren Übertragungsprotokolle steht also einem zeitgemäßen webbasierten Management der VPN-Komponenten nichts mehr im Wege.

Überwachung

Auch für die Überwachung von traditionellen Netzwerkkomponenten hat sich neben dem SNMP-Protokoll zunehmend das webbasierte Management durchgesetzt. Unter Überwachung versteht man in diesem Zusammenhang die gezielte, durch eine Managementstation oder einen Browser ausgelöste Abfrage von Systemparametern und das Versenden so genannter Traps, die von einem Netzwerksystem aufgrund eines kritischen Systemzustands ausgelöst wurden.

Das gezielte Abfragen von MIB-Variablen in den Netzwerkkomponenten muss sowohl durch SNMP-Get-Befehle als auch durch einen Browser unterstützt werden. Auch hier gilt es, Vorsicht walten zu lassen, falls man damit auch sicherheitsrelevante Parameter abfragt, denn jedermann, der Zugriff auf die Netzwerkverbindung zwischen VPN-Komponente und Managementstation hat, kann SNMP und HTTP mitlesen! Der Einsatz von sicheren Übertragungsprotokollen ist auch hier dringend angeraten.

Die Erzeugung von SNMP-Traps zur Signalisierung von Alarmzuständen muss in jedem Fall unterstützt werden. Hier werden keine sicherheitsrelevanten Parameter ungeschützt übertragen. Es wird lediglich der Managementstation mitgeteilt, welches Problem wo aufgetreten ist. Spezielle sichere Übertragungsprotokolle sind hierfür nicht notwendig.

Aber ein anderer Punkt verdient in diesem Zusammenhang Beachtung: SNMP definiert im Standard nur eine kleine Anzahl von Traps. Will man aber Traps erzeugen, die bei Eindringversuchen in das Netzwerk oder bei anderen einsatzspezifischen Vorfällen oder Situationen generiert werden, muss das VPN-System eine derart erweiterte SNMP-Funktionalität bieten. Generell sollte es möglich sein, für alle kritischen Zustände – und damit sind nicht nur Hardware-Probleme gemeint – die Generierung eines Traps konfigurieren zu können.

Accounting

Während in lokalen Netzwerken das Accounting, also die auf Benutzer, Organisationseinheiten oder Kostenstellen heruntergebrochene Abrechnung von Netzwerkdiensten, fast niemals eingesetzt wird, ist dies im WAN-Bereich immer öfter und beim Remote Access fast immer zu finden. Dies hat eine Reihe von Gründen. Im LAN-Bereich fallen keine zusätzlichen Übertragungskosten an, man kann eine Weiterverrechnung auf Basis von Netzwerkports, Plattenplatz, Geschwindigkeiten usw. durchführen. Des Weiteren bieten die meisten LAN-Komponenten zurzeit nur sehr rudimentäre oder gar keine Accounting-Möglichkeiten, so dass nicht selten die notwendigen technischen Möglichkeiten gar nicht gegeben sind.

Im Bereich der öffentlichen Netze addieren sich zu den Kosten für das Equipment und den Betrieb die teilweise recht hohen Gebühren, die an die Carrier auf zeit- oder volumenabhängiger Basis zu entrichten sind. Diese Kosten will man in der Regel nicht in einem festen Verhältnis auf die Kostenstellen umlegen, da nicht jede Organisationseinheit die Dienste in gleichem Maße in Anspruch nimmt. Es sollen vielmehr die tatsächlich verursachten Kosten ermittelt und weiter verrechnet werden. Somit findet man insbesondere im Remote-Access-Bereich Accounting-Funktionalitäten, die eine zeit- und volumenabhängige Abrechnung der Dienste ermöglichen. Auch im Bereich der traditionellen WAN-Router findet man zunehmend Abrechnungsfunktionen auf IP-Ebene.

Löst man diese herkömmlichen Strukturen nun durch die VPN-Technologie ab, müssen diese Accounting-Systeme ebenfalls unterstützt werden, auch im parallelen Betrieb während der Migrationsphase. RADIUS (Remote Authentication Dial In User Service, ein Standard, der Protokolle und Funktionen zur Authentifizierung, Autorisierung und zum Accounting von Remote-Access-Benutzern beschreibt) hat sich im Laufe der Jahre zu einem weit verbreiteten Standard mit hoher Interoperabilität zwischen verschiedenen Herstellern entwickelt. Praktisch alle Remote-Access-Systeme, sowohl im Enterprise- als auch im Carrier-Umfeld, arbeiten mit RADIUS zur Benutzerauthentifizierung und zum Accounting. Um die Datensätze auszuwerten, die von RADIUS-Servern erzeugt werden, gibt es mittlerweile eine Fülle von Programmen. Das Format eignet sich auch zum direkten Importieren in verbreitete Applikationen wie Datenbanken oder Tabellenkalkulationsprogramme.

Aus diesen Gründen liegt es auf der Hand, dass VPN-Systeme RADIUS-Accounting unterstützen müssen. Dadurch können sie einfach und schnell in eine bestehende Infrastruktur integriert werden.

2.5.2 Sicherheit

Sicherheitsstrategie

Ein VPN muss sich reibungslos in die Sicherheitsstrategie einer Organisation integrieren lassen. Die Sicherheitsstrategie, auch als Security Policy bezeichnet, ist eine unternehmensweite Definition von Sicherheitsanforderungen, die eine ganze Reihe von Bereichen abdecken müssen, wie zum Beispiel die Behandlung von Kennwörtern, physikalischer Zugangsschutz, Zugriffsregelung auf kritische Ressourcen usw. Sie beschreibt dabei sowohl die Anforderungen als auch die Zuständigkeitsbereiche für deren Umsetzung. Eine gute Sicherheitsstrategie definiert dabei nicht, welche Technologien eingesetzt werden oder bestimmte Schlüssellängen, dies ist Aufgabe der für die Umsetzung verantwortlichen Organisationseinheiten. Die Security Policy legt lediglich fest, welche Daten über welchen Zeitraum vor welchen potenziellen Angriffen und durch wen zu schützen sind. Dies klingt einfach und ist es auch. Was komplex ist, ist ihre konsistente Umsetzung.

Eine Sicherheitsstrategie legt zum Beispiel fest, dass die Konstruktionsdaten eines Unternehmens mindestens zwanzig Jahre auch vor Zugriffen durch fremde Geheimdienste sicher sein müssen, und benennt optional noch die dafür verantwortlichen Organisationseinheiten. Das war es auch schon, denn die Umsetzung obliegt den betroffenen Organisationseinheiten, die mit diesen Daten umgehen. Im Netzwerkbereich leiten sich daraus die entsprechenden Kriterien ab, um die Übertragungen sowohl im Intranet als auch im Weitverkehrsnetz oder VPN durch starke Verschlüsselung zu schützen.

Authentifizierung

Üblicherweise sind für Benutzer, die sich per Remote Access in das Unternehmensnetzwerk einwählen, in der Sicherheitsstrategie bestimmte Anforderungen hinsichtlich der Authentifizierung festgelegt. Im lokalen Netzwerk kann man sich oftmals ohne besondere Authentifizierung anschließen, denn hier sieht die Security Policy des Unternehmens in der Regel einen physikalischen Zugangsschutz vor, meist per Magnetkarte am

Eingang der Gebäude. Dieser Zugangsschutz entfällt beim Remote Access natürlich und muss durch besondere Verfahren zur Identitätsfeststellung nachgebildet werden. Dies trifft auf den traditionellen Remote Access und Remote-Access-VPN gleichermaßen zu.

Firewalls

In den meisten Organisationen wird heute ein Internetanschluss betrieben. Da man aus Sicherheits- und anderen Gründen (z.B. wegen der Verwendung privater, nicht registrierter IP-Adressen) sein Intranet nicht direkt mit dem Internet verbinden kann, wird diese Funktion in der Regel von so genannten Firewalls übernommen.

Eine Firewall kontrolliert den gesamten Datenverkehr zwischen Systemen im Internet und dem Intranet. Sie soll einerseits verhindern, dass sich Unbefugte Zugang zum Intranet verschaffen, und andererseits den Datenverkehr kontrollieren, den Systeme im Intranet in das Internet leiten. Hierbei werden die Datenpakete sogar bis auf die Ebene der Dateninhalte geprüft, um zum Beispiel Webseiten mit moralisch fragwürdigem Inhalt zu sperren. Der wichtige Punkt hierbei ist, dass eine Firewall den Verkehr in das Internet kontrolliert.

Ein Internet-VPN greift hingegen auf kein einziges Internetsystem zu, sondern es benutzt das Internet lediglich als Transportmedium für IP-Pakete. Dies ist eine völlig andere Situation, da die IP-Pakete ausschließlich im privaten Bereich versendet und empfangen werden. Querverbindungen in das Internet sind bei einigen sehr guten, dedizierten VPN-Systemen überhaupt nicht möglich.

Aus diesem Grund macht es auch wenig Sinn (und kann von Fall zu Fall sogar Probleme bereiten), wenn man beide Funktionalitäten auf einem System zusammenführt, egal ob man VPN-Funktionen in eine Firewall integriert oder umgekehrt eine Firewall in einen VPN-Konzentrator. Vielmehr ist zu fordern, dass der VPN-Router als Ablösung klassischer Netzwerkkomponenten wie Router oder Remote-Access-Konzentratoren und die Firewall gleichzeitig nebeneinander zu betreiben sind.

PKI

Technologien wie E-Business oder E-Commerce und neu geschaffene rechtliche Rahmenbedingungen für digitale Unterschriften haben zusammen mit einem gestiegenen Sicherheitsbedürfnis bei der Benutzung öffentlicher Netze eine Reihe von Applikationen hervorgebracht, die mit elektronischen, digitalen Schlüsseln Daten verschlüsseln oder Dokumente signieren. Bei großen Organisationen fällt dabei eine ganze Reihe verschiedener Schlüssel an, die erzeugt, verwaltet, regelmäßig erneuert und vor allem eindeutig bestimmten Personen zugeordnet werden müssen.

Eine Public-Key-Infrastruktur (PKI) leitet ihren Namen von der so genannten Public-Key-Kryptografie (vgl. Kapitel 4) ab, bei der mit zwei verschiedenen Schlüsseln, einem öffentlichen und einem privaten, gearbeitet wird. Mit einem Schlüssel (Public Key, dem öffentlichen Schlüssel) wird verschlüsselt und mit dem anderen (Private Key, dem privaten, geheimen Schlüssel) wieder entschlüsselt. Die beiden Schlüssel bilden ein Paar; der öffentliche Schlüssel ist jedermann zugänglich und fest an eine Person gebunden, während der private von der Person geheim gehalten wird. Etwas, das mit dem einen Schlüssel verschlüsselt wurde, kann nur mit dem korrespondierenden anderen Schlüssel wieder entschlüsselt werden.

Die Bindung einer Person an einen öffentlichen Schlüssel erfolgt über ein so genanntes digitales Zertifikat. Die Erzeugung und Verwaltung dieser Zertifikate, die Registrierung der Benutzer und das Erzeugen und Speichern von Schlüsselpaaren zum Zweck der Datenverschlüsselung auf Rechnern ist die Hauptfunktion einer PKI. Diese kann von einem Unternehmen selbst betrieben werden, oder man benutzt eine öffentliche PKI.

Falls nun digitale Zertifikate zum digitalen Signieren eingesetzt werden, liegt es auf der Hand, diesen Mechanismus auch zur Authentifizierung von Benutzern oder VPN-Systemen einzusetzen. Hier sollten die VPN-Geräte und die Client-Software eine entsprechende Unterstützung bieten.

2.6 Koexistenz zu traditionellen WAN

Üblicherweise findet nicht immer sofort eine Ablösung von traditionellen Netzen durch virtuelle private Netze statt. Vielmehr müssen beide für eine gewisse Migrationsphase gleichzeitig zu betreiben sein. Typische Fälle sind der gleichzeitige Einsatz von VPN und Frame-Relay- oder ATM-Netzen, da es oft der Fall ist, dass die beiden letztgenannten Technologien nicht an jedem benötigten Standort zur Verfügung stehen.

Auch normaler Remote Access per Einwahl über das Telefonnetz wird sehr oft parallel zum VPN-Remote-Access betrieben. Hier sollen aus Kostengründen zwar möglichst viele Verbindungen über das VPN laufen, jedoch nutzt man den Einwählteil oft als Backup oder für bestimmte kritische Applikationen, die eine garantierte Einwahl und bestimmte Qualitätsmerkmale benötigen.

Um das Ratiopotenzial eines VPN voll ausschöpfen zu können, dürfen Einsparungen, die auf der einen Seite durch geringere Gebühren und günstigeres Equipment erzielt wurden, nicht durch andere, durch die VPN-Technologie anfallende, neue Kosten zunichte gemacht werden. Solche Kosten sind oft versteckt und auf den ersten Blick vielleicht gar nicht mit dem Einsatz eines VPN in Zusammenhang zu bringen.

Einer der negativen Kostenfaktoren, vielleicht der schwerwiegendste, sind erhöhte Managementkosten. Diese können erzeugt werden, wenn sich die VPN-Systeme nicht in das vorhandene Netzwerkmanagement integrieren lassen und ein paralleles Management aufgebaut wird.

Auch Zusatzkosten durch möglicherweise erforderliche IP-Adressänderungen in einem großen Netzwerk sollten vermieden werden, indem man die VPN-Technologie in bestehende Strukturen integriert und nicht umgekehrt bestehende Strukturen an das VPN anpasst.

2.7 Adressmanagement

Das IP-Adressmanagement ist in vielen Unternehmen zunehmend zu einem kritischen Faktor geworden. Dies hat historische Gründe, die mit der Entwicklung und Einführung des IP-Protokolls zu tun haben.

IP war viele Jahre ein Protokoll, das fast ausschließlich im Internet eingesetzt wurde. Die Folge war, dass alle beteiligten Geräte offizielle IP-Adressen und etliche Organisationen und Firmen offizielle IP-Netze zugeteilt bekamen. Als TCP/IP praktisch Bestandteil von UNIX wurde, es als kostenloser Zusatz für Windows 3.x verfügbar war und Betriebssysteme wie VM oder OS/400 von IBM ebenfalls damit ausgerüstet wurden, gab es ein paar Entwicklungen, an denen viele Unternehmen heute noch zu knabbern haben. Denn in vielen Firmen begannen sich regelrechte „IP-Inseln“ zu bilden, oft mit selbst vergebenen, nicht registrierten Adressbereichen.

Zuerst wollte man überhaupt keinen Internetzugriff, im Zeitalter des World Wide Web wurde dieser Wunsch von den Anwendern jedoch immer dringlicher vorgetragen. Als Ausweg boten sich Firewalls mit NAT (Network Address Translation) an. Im Intranet der Unternehmen wurden bis dato die IP-Adressen manuell vergeben und statisch auf den Systemen eingetragen. Dokumentiert wurde meist nichts, und wenn doch, dann waren die Daten oft nicht aktuell und unvollständig. Im Verlauf dieser und anderer Entwicklungen wurde der Ruf nach einem einfachen und automatischen Management der IP-Adressen immer lauter.

Das Resultat war das DHCP (Dynamic Host Configuration Protocol), ein Verfahren, mit dem IP-Schnittstellen automatisch und dynamisch von einem speziellen Server konfiguriert werden. DHCP ist eine Client-Server-Applikation, bei der die Clients beim Start eines Rechners mit einer IP-Schnittstelle von einem DHCP-Server eine IP-Adresse und weitere optionale Konfigurationsparameter anfordern. Es hat mittlerweile eine recht breite Verwendung gefunden. Neben Implementierungen in PC-Betriebssystemen gibt es auch professionelle, hoch skalierbare Systeme, die mit relationalen Datenbanken arbeiten und einen dynamischen DNS-Server (Domain Name System) integriert haben, der die sich ändernden IP-Adressen immer dem korrekten Domainnamen zuordnet.

Viele heute eingesetzte Remote-Access-Konzentratoren verwenden DHCP, um den Clientrechnern eine IP-Adresse dynamisch zuzuweisen. Diese Funktionalität muss ein VPN-Gateway ebenfalls unterstützen, um in das unternehmensweite IP-Adressmanagement integrierbar zu sein. Idealerweise ist sowohl ein gerichtetes DHCP möglich, bei dem sich die Systeme an dedizierte DHCP-Server wenden, als auch ein ungerichtetes, bei dem per Netzwerk-Broadcast ein passender Server gesucht wird.

Die Anforderungen im Bereich der IP-Adressverwaltung beschränken sich jedoch keinesfalls nur auf DHCP, es müssen auch andere Mechanismen zur Zuteilung von IP-Adressen unterstützt werden. Hierbei resultieren die Anforderungen sowohl aus der aktuellen Managementumgebung als auch aus zukünftig einzusetzenden Systemen. Insbesondere arbeiten viele Remote-Access-Konzentratoren mit RADIUS-Servern, auf denen neben der Authentifizierung und dem Accounting auch IP-Adressen userbezogen oder aus einem so genannten Pool heraus vergeben werden. Auch eine gerätespezifische Vergabe von IP-Adressen auf Basis von Benutzernamen oder aus einem Pool sollte von einem modernen VPN-Router unterstützt werden.

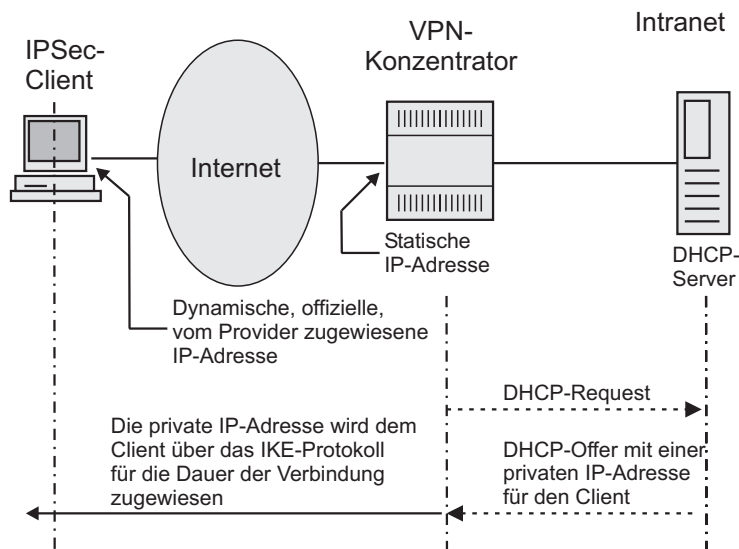


Abbildung 2.10: Vorhandene IP-Managementsysteme wie DHCP sollten auch für VPN benutzbar sein.

Directory Services (Verzeichnisdienste) erfreuen sich in der Fachwelt einer immer größeren Beliebtheit. Nachdem in der Vergangenheit einige proprietäre Verfahren wie Novell NDS oder Banyan Streetwork und Standards wie X.500 miteinander konkurriert hatten, konzentriert sich die Entwicklung momentan auf das Lightweight Directory Access Protocol (LDAP). Es wird mittlerweile von allen Größen der Netzwerkindustrie sowohl im Bereich der Betriebssysteme als auch im Bereich der Netzwerkkomponenten und Managementsysteme eingesetzt. LDAP ist sehr flexibel und bietet einen fest spezifizierten Satz von Funktionen und Formaten zum Erzeugen, Modifizieren, Abfragen und Löschen von Einträgen in der Verzeichnisdatenbank. Die Struktur eines Verzeichnisses ist frei wählbar und kann somit maßgerecht an unterschiedliche Bedürfnisse angepasst werden.

Ein VPN-System, das auch noch in der Zukunft einsetzbar sein soll, muss daher ebenfalls LDAP unterstützen können, um in unternehmensweite Verzeichnisdienste integrierbar zu sein.

2.8 Interoperabilität

Interoperabilität ist eine wichtige und kritische Anforderung an heutige VPN-Router und VPN-Gateways. Dies hat gleich mehrere Gründe. Je nach Auswahl eines geeigneten Tunneling-Modells (vgl. Kapitel 3) können mehrere Partner an einem VPN beteiligt sein (zum Beispiel ein Internet Service Provider und der Endkunde), die unter Umständen Equipment verschiedener Hersteller einsetzen.

Ein anderes Beispiel für die Notwendigkeit der Interoperabilität verschiedener VPN-Systeme ist die Dynamik im Bereich von Firmenzusammenschlüssen oder Kooperationen, die in der heutigen Zeit zu beobachten ist. Hierbei können in den unterschiedlichen Unternehmen Anforderungen der Art entstehen, dass die eingesetzten VPN-Geräte mit-

einander kommunizieren müssen. Das Automotive Network Exchange (ANX) ist beispielsweise ein VPN, das für Automobilhersteller und Zulieferer mit unterschiedlichsten Geräten auf der Basis des IPSec-Protokolls betrieben wird.

Dies erfordert, dass das jeweils verwendete Equipment mit den Gegenstellen interoperabel ist. Die Lösung besteht darin, ausschließlich solche Protokolle einzusetzen, die standardisiert und allgemein akzeptiert sind. Leider hat die Vergangenheit gezeigt, dass auch standardkonforme Implementierungen bestimmter Datenkommunikationsprotokolle keineswegs ein Garant für eine reibungslose Kommunikation sind. Dies liegt an der Tatsache, dass viele Protokolle sehr flexibel sein müssen und gegebenenfalls so konfiguriert werden können, dass eine Kommunikation mit anders konfigurierten Gegenstellen unmöglich ist.

Wichtig ist daher, dass es eine oder mehrere mögliche Konfigurationen der Protokolle gibt, bei denen eine Interoperabilität gewährleistet oder sogar nachgewiesen ist. Insbesondere im Bereich des IPSec-Protokolls ist dies zu fordern. ICSA-Labs ist eine sich selbst als unabhängig bezeichnende Firma, die sich mit Internetsicherheit beschäftigt. Die ICSA Labs führen Tests durch, welche die Funktionalität und Interoperabilität von IPSec-Implementierungen ermitteln sollen. Geräte oder Programme, die erfolgreich getestet wurden, dürfen das ICSA-IPSec-Logo tragen und sind, was viel wichtiger ist, mit einer sehr hohen Wahrscheinlichkeit zu anderen zertifizierten Systemen interoperabel.

3 Tunneling

Tunneling ist die Basis praktisch aller VPNs. Mit Hilfe dieser Technologie kann man Pakete eines Netzwerkprotokolls in Pakete eines anderen Netzwerkprotokolls einkapseln und über dieses Netzwerk übertragen. Auf diese Weise kann man zum Beispiel IPX-Pakete durch ein IP-Netzwerk transportieren. Eine andere, speziell für IP-Netze interessante Anwendung ist das Verstecken von privaten, nicht registrierten Netzwerk- und Hostadressen, indem man IP in IP tunnelt. Auf diese Weise kann man seine privaten Netze über das Internet miteinander verbinden. Man kapselt hierzu die IP-Pakete mit privaten Adressen in Pakete mit offiziell registrierten IP-Adressen ein und transportiert diese durch das Internet zur Gegenstelle, welche die originalen Pakete wieder auspackt.

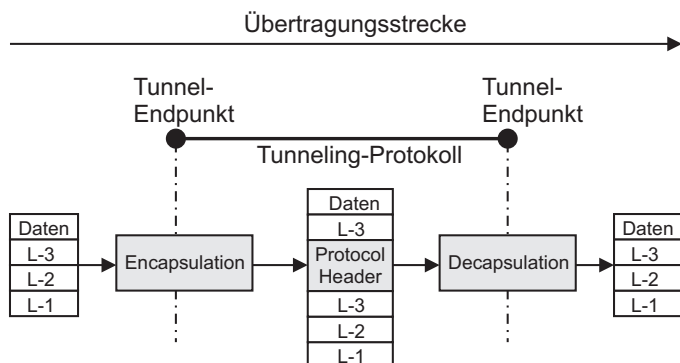


Abbildung 3.1: Das Prinzip aller Tunneling-Protokolle

Es gibt eine ganze Reihe von Tunneling-Protokollen, von denen jedoch drei eine besondere Rolle spielen: das Layer 2 Tunneling Protocol (L2TP), das IP-Security Protocol (IPSec) im Tunnel-Modus und Multi Protocol Label Switching (MPLS). Diese drei Protokolle sind standardisiert, und sie verdrängen zunehmend alle anderen Tunneling-Protokolle in Neuinstallationen. SSL, obwohl immer öfter in VPN eingesetzt, ist kein Tunneling-Protokoll. Hier müssen zu SSL externe Komponenten die Einkapselung von UDP-, TCP- oder IP-Paketen vornehmen.

In Abbildung 3.1 ist das Prinzip des Tunnelings veranschaulicht. Die Netzwerkpakete (Layer 3) werden in andere Netzwerkpakete mit einem neuen Layer-3-Header eingekapselt (Encapsulation), und es wird ein Tunnel-Header eingefügt. An diesem Header, der zwischen dem neuen Layer-3-Header und der Nutzlast angeordnet sein muss, erkennt der Empfänger, dass es sich um ein Paket des betreffenden Tunneling-Protokolls handelt. Er wertet den Header aus, entpackt das originale Paket (Decapsulation) und transportiert es weiter. Zwischen dem Sender und dem Empfänger, den so genannten Tunnelendpunkten, die durch die Netzwerkadressen des neuen Layer-3-Headers festgelegt sind, besteht eine Tunnelverbindung.

In manchen Fällen ist es nicht notwendig, im IP-Protokoll einen speziellen Tunnel-Header einzufügen, denn gemäß dem Standard benutzt man das Protokollfeld des IP-Headers, um anzuzeigen, welches Protokoll auf den Header folgt. Üblicherweise ist dies TCP (mit Protokollnummer 6) oder UDP (mit Protokollnummer 17). Wenn jedoch die Nummer 4 (IP) in diesem Feld enthalten ist, weiß die Gegenstelle, dass ein IP-Paket folgt, also ein vollständiges Paket eingekapselt wurde. Auf diese Weise kann man IP-in-IP-Tunneling extrem effizient ohne einen zusätzlichen Header verwenden.

Die Tunnelendpunkte können in der Regel sowohl normalen Netzwerkverkehr als auch Tunnelpakete verarbeiten. Wie ein Paket zu behandeln ist, erkennt der Empfänger am Tunnel-Header oder an der IP-Protokollnummer. Der Sender nimmt eine Einkapselung aufgrund der Verarbeitungsstrategie einer höheren Protokollschicht vor. Dabei kann es sich um eine Sicherheitsstrategie handeln oder um interne Tabellen mit Kennzeichnungen, aus denen hervorgeht, welche Pakete normal zu transportieren sind und welche getunnelt werden müssen. Es gibt allerdings eine Ausnahme: Security-Gateways werden aus Sicherheitsgründen manchmal so konfiguriert, dass sie ausschließlich Tunneling-Protokolle terminieren und keine anderen Protokolle wie zum Beispiel SNMP, FTP usw.

3.1 Tunneling-Modelle

In Abhängigkeit davon, wo die Tunnel beginnen und enden, kann man drei verschiedene, so genannte Tunneling-Modelle unterscheiden: das Ende-zu-Ende-Modell, das Provider-Enterprise-Modell und das Intra-Provider-Modell. In Abbildung 3.2 sehen Sie die drei Modelle im Vergleich.

3.1.1 Das Intra-Provider-Modell

Das Intra-Provider-Modell zeichnet sich dadurch aus, dass die Tunnel bei den Service Providern beginnen und enden. Gegenüber den beiden anderen Modellen hat diese Variante einen entscheidenden Vorteil: Auf der Seite des Kunden muss keine spezielle Hard- oder Software installiert werden, er kann seine vorhandenen Systeme weiter benutzen und ist nicht in das Tunneling involviert. Die notwendigen Gateways werden von den Providern betrieben und stehen normalerweise auch in deren Räumen. Dieses Modell wird vorwiegend für Remote-Access-VPN verwendet, kann jedoch auch für Branch-Office-VPN eingesetzt werden. Das in einem der folgenden Abschnitte beschriebene MPLS ist ein typischer Vertreter dieser Kategorie und wird sehr oft zur Verbindung verschiedener Standorte eingesetzt.

3.1.2 Das Provider-Enterprise-Modell

Beim Provider-Enterprise-Modell sind sowohl die Service Provider als auch die Endkunden in das Tunneling involviert. Die Tunnel beginnen im POP (Point of Presence) des Service Providers und enden im Gateway des Kunden. Dieses Modell wird primär für Remote-Access-VPN eingesetzt, kann aber auch in Branch-Office-VPN Verwendung finden. In diesem Modell muss der Kunde eine spezielle Hard- oder Software als VPN-Gateway einsetzen. Die Client-Software kann ein beliebiges Einwählprogramm (Microsoft DFÜ-Netzwerk usw.) sein, da der Tunnel erst im POP des Providers beginnt.

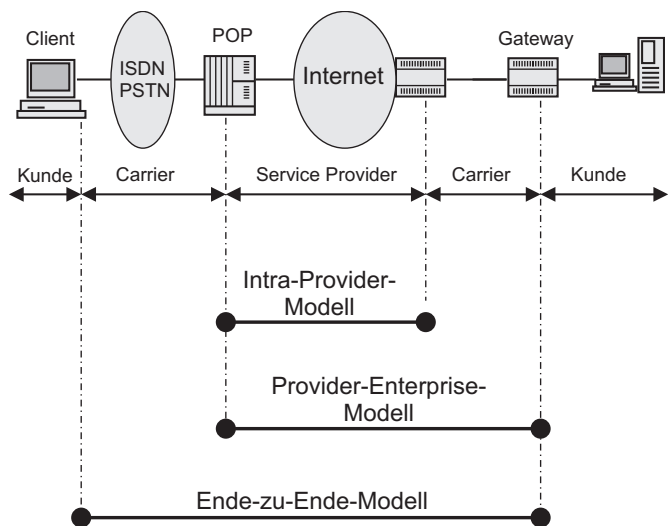


Abbildung 3.2: Die drei grundsätzlichen Tunneling-Modelle

3.1.3 Das Ende-zu-Ende-Modell

Im Ende-zu-Ende-Modell werden die Tunnel ausschließlich vom Kunden aufgebaut. Die Carrier und/oder Service Provider sind dabei gar nicht in das Tunneling involviert. Sie transportieren im Fall von IP-VPN ausschließlich IP-Pakete. Dass in diesen Paketen andere Pakete eingekapselt sind, können sie zwar erkennen (wenn sie das wollen), aber es hat keinen Einfluss auf den Transport der IP-Pakete.

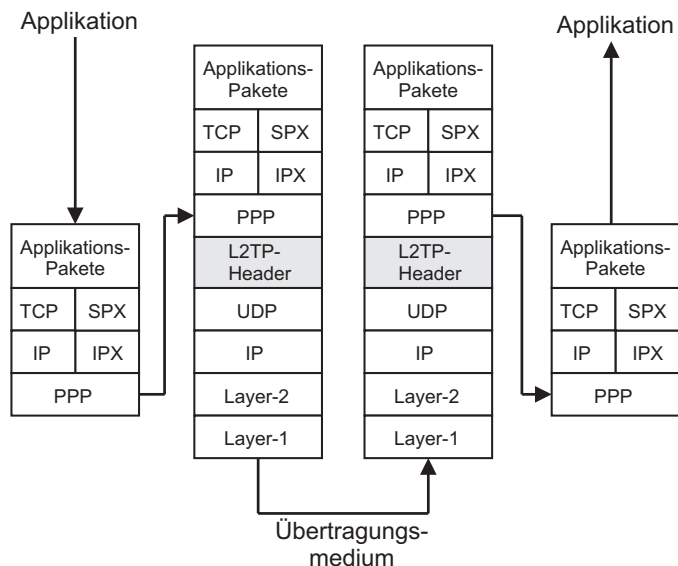


Abbildung 3.3: Das Prinzip des Layer-2-Tunnelings, in dem Pakete der Schicht 2 eingekapselt werden

Üblicherweise stehen die Gateways zum Netzwerk eines Providers in den Räumen der Endkunden, sind deren Eigentum und werden auch von ihnen betrieben. Jedoch kann dieses Modell auch als so genannter Carrier Managed Service von einem Service Provider oder Carrier betrieben werden. Die Remote-Access-Clients wählen sich in die POPs der Service Provider ein, und eine spezielle VPN-Clientsoftware im Endgerät des Kunden baut daraufhin einen Tunnel zu einem VPN-Gateway im Kundennetzwerk auf.

Theoretisch kann man fast alle Tunneling-Protokolle in einem Ende-zu-Ende-Modell einsetzen, jedoch sind einige Protokolle besser dafür geeignet als andere. Im Ende-zu-Ende-Modell setzt man vorwiegend IPSec im Tunnel-Modus und mittlerweile nur noch ganz selten das Point-to-Point Tunneling Protocol (PPTP) ein. Speziell in Microsoft-Umgebungen findet zunehmend das Layer 2 Tunneling Protocol (L2TP) in Verbindung mit IPSec-Transport Verwendung.

3.2 Standardisierte Tunneling-Protokolle

Man kann die Tunneling-Protokolle in drei verschiedene Klassen einteilen: die Layer-2-Tunneling-Protokolle, Layer-3-Tunneling-Protokolle und, etwas scherzhaft formuliert, Layer-2,5-Tunneling-Protokolle. Die Unterscheidung basiert auf der Schicht des OSI- oder IP-Schichten-Modells, deren Pakete eingekapselt werden. Layer-2-Protokolle kapseln Pakete der Sicherungsschicht (Layer 2) in andere Pakete, meist solche der Schicht 3, ein. Layer-3-Protokolle kapseln Pakete der Netzwerkschicht (Layer 3) in andere Pakete der Netzwerkschicht ein. Eine Sonderform ist MPLS, das eine spezielle Zwischenschicht (den Layer Zweieinhalb, im Fachjargon „Label“ genannt) zwischen Layer 2 und Layer 3 einfügt, um hiermit verschiedene virtuelle Pfade zu realisieren.

3.2.1 Layer-2-Protokolle

In Abbildung 3.3 sehen Sie die Funktionsweise des Layer-2-Tunnelings. In diesem Beispiel werden PPP-Rahmen (PPP, Point-to-Point Protocol, ein Protokoll, um verschiedene Netzwerkprotokolle über eine Punkt-zu-Punkt-Verbindung zu transportieren) in IP-Pakete eingekapselt. In den PPP-Rahmen können Pakete verschiedener anderer Netzwerkprotokolle wie IP, IPX, Vines-IP, NetBEUI usw. enthalten sein.

In unserem Beispiel tunnelt das Layer 2 Tunneling Protocol (L2TP) IP und IPX. Beim Tunneling entsteht ein gewisser Overhead, denn außer dem L2TP-Header werden zusätzliche UDP-, IP- und PPP-Header erzeugt. Bei großer Nutzdatenlänge fällt dieser Anteil allerdings nicht so sehr ins Gewicht. Der Vorteil von Layer-2-Tunneling-Protokollen ist der, dass eine Vielzahl von Netzwerkprotokollen getunnelt werden kann, ohne dass sich das Tunneling selbst darum kümmern muss, da dies bereits auf der PPP-Ebene erfolgt ist.

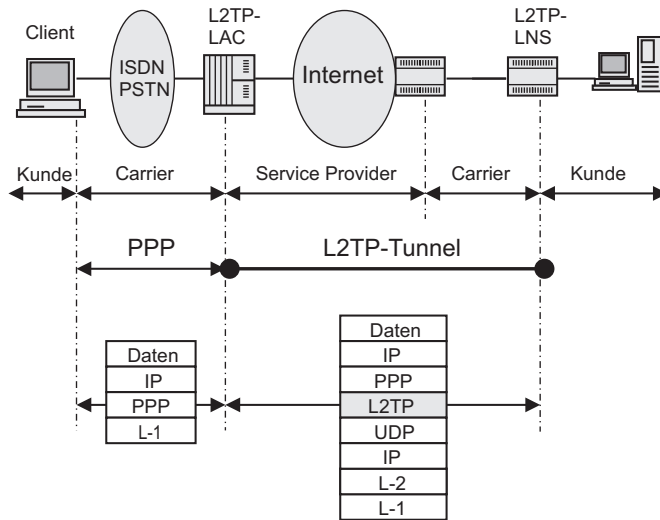


Abbildung 3.4: L2TP im Compulsary Mode (zwangsweises Tunneln), in dem vom Provider generell ein Tunnel zum Kundennetz aufgebaut wird

3.2.2 Layer 2 Tunneling Protocol (L2TP)

Das Layer 2 Tunneling Protocol, L2TP, wurde primär für den Einsatz im Provider-Enterprise-Modell entwickelt. Wie Sie in Abbildung 3.4 sehen, beginnt dabei der L2TP-Tunnel im Remote Access Concentrator (RAC) des Service Providers und endet in einem Gateway beim Kunden.

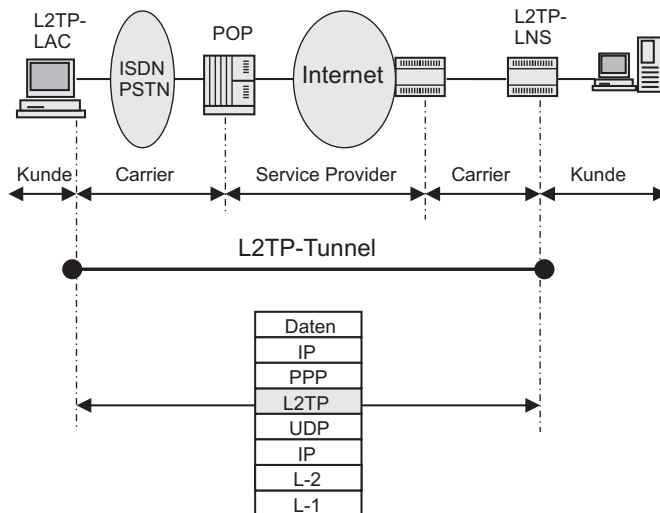


Abbildung 3.5: L2TP im Voluntary Mode (freiwilliges Tunneln), in dem der Client entscheidet, ob ein Tunnel aufgebaut werden muss

Die Funktionalität der Tunnelendpunkte ist rollenbasierend, der Tunnel wird von einem LAC (L2TP Access Concentrator) aufgebaut, der im RAC des Providers implementiert ist. Das entgegengesetzte Ende des Tunnels bildet der LNS, der L2TP Network Server, der entweder auf Routern oder auf speziellen VPN-Gateways implementiert ist.

Der RAC des Service Providers erkennt an bestimmten Kennzeichen eines eingehenden Rufs (angerufene Nummer, Benutzer-ID usw.), ob es sich um eine Verbindung in das Netzwerk des Providers handelt oder ob die Verbindung zu einem Endkunden getunnelt werden soll. Dieses Verhalten nennt man auch zwangsweises Tunneling (Compulsory Tunneling), da der Client nicht selbst entscheiden kann, ob ein Tunnel aufgebaut wird oder nicht. Im Provider-Enterprise-Modell muss das Equipment beim Kunden eine L2TP-LNS-Funktionalität bieten. Das betreffende System muss aber nicht notwendigerweise dem Kunden gehören und muss auch nicht von ihm verwaltet werden – oft bieten die Service Provider komplette Lösungen an, inklusive VPN-Systemen, Konfiguration, Wartung und Betrieb. Leider sind in L2TP keinerlei Sicherheitsfunktionen wie Datenverschlüsselung oder Paketauthentifizierung verfügbar.

L2TP kann auch im Ende-zu-Ende-Modell eingesetzt werden. Zu diesem Zweck wird die LAC-Funktionalität vom Remote-Access-Konzentrator des Service Providers auf den Client verlagert. Dieser so genannte „Virtual LAC“ auf dem Clientrechner baut den L2TP-Tunnel zum LNS auf, und die Carrier und Service Provider sind nicht mehr in das Tunneling involviert. Im Gegensatz zum zwangsweisen Tunneln nennt man dies freiwilliges Tunneln (Voluntary Tunneling). Ein Beispiel für eine solche Funktionalität finden Sie bei Windows 2000 Professional und Windows XP, das als VPN-Protokoll auch L2TP als virtuellen LAC implementiert hat.

Das Layer 2 Tunneling Protocol ist aufgrund seiner wachsenden Bedeutung in Kapitel 8 ausführlich beschrieben.

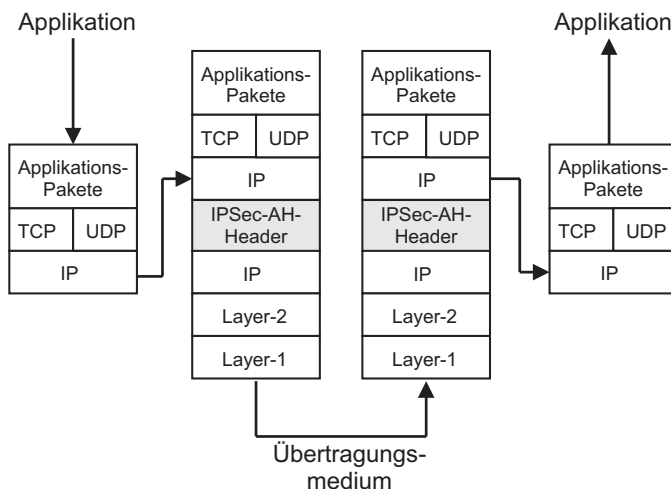


Abbildung 3.6: Layer-3-Tunneling zeichnet sich dadurch aus, dass Pakete auf der Schicht 3 eingekapselt werden.

3.2.3 Layer-3-Protokolle

Das Layer-3-Tunneling arbeitet eine Schicht höher als Layer-2-Protokolle. Hier werden Pakete der Netzwerkschicht in andere Pakete dieser Schicht eingekapselt, wie Sie in Abbildung 3.6 sehen. Der Paket-Overhead ist geringer als der von Layer-2-Protokollen. Allerdings muss der Tunneling-Prozess die von den höheren Schichten ankommenden Pakete analysieren und im Tunnel-Header vermerken, welche Art von Protokoll getunnelt wird. In dem Beispiel, das in Abbildung 3.6 zu sehen ist, wird IPSec im Tunnel-Modus eingesetzt. Das gezeigte IPSec-AH-Protokoll (vgl. Kapitel 7) fügt einen IPSec-AH-Header ein. Andere Protokolle, zum Beispiel IPSec-ESP, können neben dem Header auch einen Trailer in das zu erzeugende Paket einfügen.

3.2.4 IP Security (IPSec) im Tunnel-Modus

IPSec im Tunnel-Modus dient meist als Basis für das Ende-zu-Ende-Modell. Alle beteiligten Systeme müssen mit einer entsprechenden IPSec-Implementierung ausgerüstet sein. In Abbildung 3.7 sehen Sie am Beispiel eines Remote-Access-VPN die Funktionsweise. Die IPSec-Tunnel beginnen und enden auf dem Endsystem des Kunden, meist einem Rechner mit IPSec-Client-Software und einem IPSec-Gateway. Die Carrier und Service Provider sind nicht am Tunneling beteiligt und transportieren aus ihrer Sicht nur die IP-Pakete zwischen Client und Gateway. Das öffentliche Interface des IPSec-Gateways hat eine feste, offiziell registrierte IP-Adresse. Auch der Client bekommt bei der Einwahl in den POP eines Service Providers eine offizielle IP-Adresse zugewiesen. Das private Interface des Gateways kann in einem nicht registrierten IP-Netzwerk liegen.

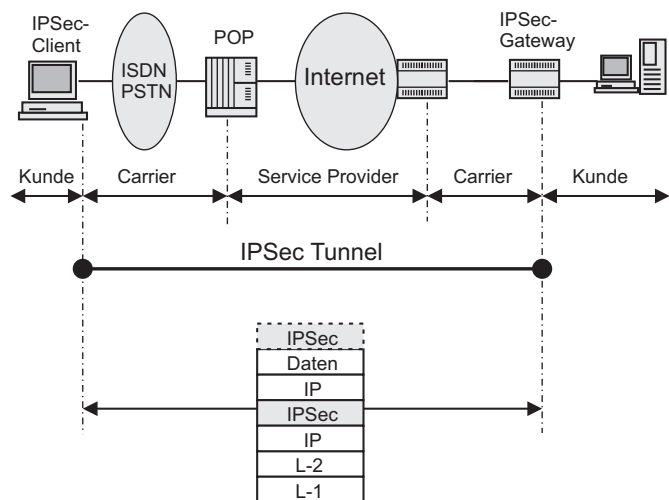


Abbildung 3.7: Obwohl Tunneling nur eine seiner Optionen ist, ist IPSec zum Inbegriff des Layer-3-Tunnelings geworden.

Beim Client ist die Sache ein klein wenig komplexer. Er hat auch zwei IP-Adressen: zum einen die ihm vom Provider für die DFÜ-Verbindung (DFÜ, Datenfernübertragung) zugewiesene, offizielle und zum anderen seine private IP-Adresse aus dem Adress-

bereich des Kundennetzes. Einige IPSec-Implementierungen erfordern leider, dass der Client immer die gleiche offizielle IP-Adresse zugewiesen bekommt, wodurch diese damit faktisch zu einer statischen Adresse wird. Das ist den Internet Service Providern (ISP) ein Gräuel, da deren verfügbare Adressen immer mehr zur Neige gehen. Üblicherweise werden den Einwähl-Clients die IP-Adressen von den ISP dynamisch aus einem so genannten IP-Adress-Pool zugewiesen. Diese Pools haben in der Regel so viele Adressen, wie der Provider an gleichzeitigen physikalischen Verbindungen zur Verfügung stellen kann. Da sich aber nie alle Kunden eines ISP gleichzeitig einwählen, haben die Provider ihre POPs in der Regel überbucht, sie haben also mehr Kunden als Einwählports – und damit auch weitaus weniger IP-Adressen als Kunden.

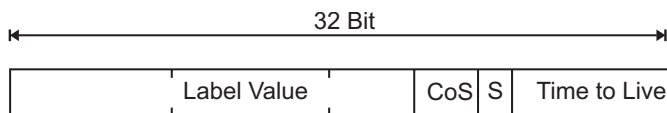
Falls nun pro Client eine feste IP-Adresse reserviert werden müsste, wären die Provider gezwungen, eine viel größere Anzahl von Adressen vorzuhalten, als wirklich zum Betrieb benötigt werden – und das wird langsam zu einem Problem, da die Adressen für IP Version 4 weltweit immer knapper werden. Manche Provider sträuben sich daher gegen statische IP-Adressen für Remote-Access-Clients, oder sie verlangen entsprechende Gebühren, was dem eigentlichen Zweck eines VPN, Kosten zu senken, widerspricht. Also sollte bei der Auswahl der IPSec-Client-Implementierung darauf geachtet werden, dass diese eine dynamische Zuweisung der offiziellen IP-Adressen unterstützt.

Die private IP-Adresse kann bei guten Client-Implementierungen auch dynamisch durch den Tunnel vom Gateway zugewiesen werden. Hier ist vom IPSec-Standard kein Verfahren festgeschrieben, es existieren lediglich standardisierte Funktionen, um herstellerspezifische Erweiterungen einzubinden. Manche Implementierungen unterstützen leider nur eine manuelle, statische Konfiguration der privaten Client-IP-Adresse.

IPSec ist primär ein Sicherheitsprotokoll auf IP-Ebene; IP-Tunneling ist eine mögliche Option – die allerdings in VPN fast immer eingesetzt wird. Die Sicherheitsfunktionen von IPSec sind in Kapitel 5 ausführlich beschrieben. Die Einkapselung der privaten IP-Pakete – IPSec kann nur IP-Pakete tunneln – erfolgt je nach Sicherheitsstufe durch einen IPSec-Header oder einen IPSec-Header und -Trailer. Ein Protokollfeld in den IPSec-Headern gibt an, welches Protokoll im IPSec-Paket eingekapselt ist. Dies sind im Tunnel-Modus ausschließlich IP-Pakete.

3.2.5 Multi Protocol Label Switching (MPLS)

Obwohl nicht primär als Tunneling-Protokoll entwickelt, eignet sich MPLS hervorragend zum Aufbau von VPNs, und zwar von Layer-2-VPNs, da das Forwarding in einem MPLS-Netzwerk ausschließlich auf Schicht 2 und den dieser Ebene zugehörigen Tabellen stattfindet.



CoS = Class of Service

S = Bottom of Stack Bit, 1 für den ersten (untersten) Label

Abbildung 3.8: Der Namensgeber für MPLS, das Label

Multiprotokoll-Label-Switching (MPLS) ist genau genommen keine Technologie, mit der Endanwender bzw. Endkunden in direkte Berührung kommen. Aber aufgrund der Bedeutung im Carrier- und Service-Provider-Bereich und den Schnittstellen zu anderen Technologien (VPN, Ethernet over WAN, QoS usw.) ist an dieser Stelle eine kurze Beschreibung dieser Technologie durchaus sinnvoll.

MPLS ist eine Technologie mit vielen Vätern, da sich in der Vergangenheit einige Unternehmen mit der exakt gleichen Zielsetzung, die auch MPLS zugrunde liegt, beschäftigt hatten und mit sehr ähnlichen Verfahren auf den Markt kamen – natürlich ohne untereinander interoperabel zu sein. Diese Zielsetzung sah vor, eine möglichst optimale Verbindung von Routing und Switching zu erreichen und gleichzeitig Investitionen in bestehende Netze auf Basis von Frame Relay und ATM zu schützen.

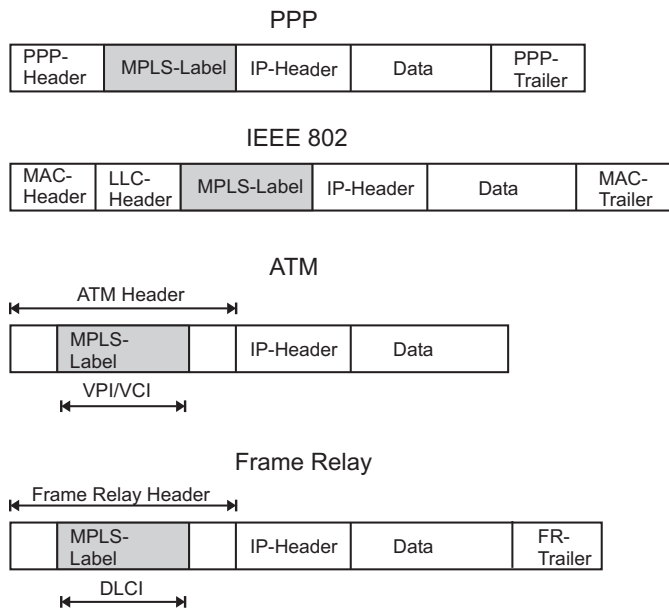


Abbildung 3.9: Die MPLS-Labels können entweder zwischen Layer 2 und Layer 3 eingefügt werden, oder es werden verwandte Felder von Layer-2-Protokollen wie ATM oder Frame Relay benutzt.

Die unterschiedlichen Ansätze von Routing und Switching werden in MPLS durch ein zweistufiges Konzept kombiniert, das folgende Eigenschaften aufweist:

In MPLS werden Steuer- und Signalisierungsdaten ganz strikt von den Nutzdaten getrennt, man bezeichnet die beiden Bereiche als Forwarding- und Control-Komponente. Sie werden (fast) völlig unabhängig voneinander implementiert. So kann ein ATM-Switch, der ja bereits schon eine eigene Forwarding-Komponente besitzt, durch Hinzufügen der MPLS-Control-Funktionalität zu einem MPLS-Router werden.

Der Weg durch ein Netzwerk wird nach wie vor durch klassisches Routing bestimmt. Zusätzlich sind weitreichende Möglichkeiten für Traffic Engineering gegeben.

Signalisierungsprotokolle, an denen alle beteiligten Systeme teilnehmen, legen vorab fest, welchen Weg Datenpakete nehmen, und konfigurieren diese Systeme hierfür.

Der Weg, den die Datenpakete nehmen sollen, wird aufgrund lokaler Tabellen (LIB, Label Information Base) innerhalb der Systeme bestimmt. Dies kann sehr schnell geschehen und stellt somit praktisch die Switching-Komponente von MPLS dar. So genannte Labels, die in die Pakete beim Eintritt in ein MPLS-Netzwerk eingefügt und beim Verlassen wieder entfernt werden, dienen zur Forwarding-Entscheidung.

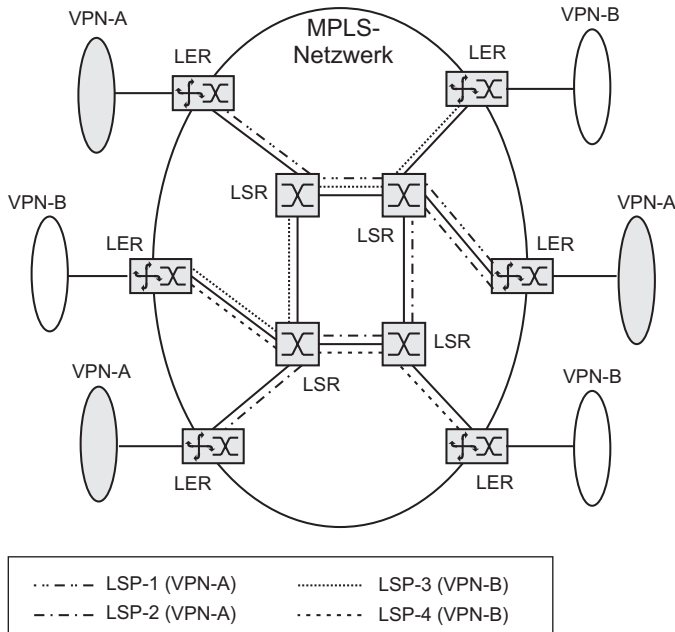


Abbildung 3.10: Ein typisches MPLS-VPN. Für den Endkunden ist MPLS meist nicht sichtbar, da es im Label Edge Router (LER) des Carriers beginnt.

MPLS kann seine Labels auf verschiedene Art erzeugen und benutzen. So sind drei Varianten möglich, von denen insbesondere die beiden ersten heutzutage stark genutzt werden:

Es wird ein eigener Protokoll-Header zwischen Schicht 2 und 3 erzeugt. Der Vorteil liegt hier in der Möglichkeit, mehr oder weniger unbegrenzt Labels ineinander verschachteln (Label Stacking) zu können.

MPLS benutzt Felder von Headern verschiedener Layer-2-Protokolle, insbesondere Frame Relay oder ATM, da diese bereits den größten Teil der Netze von Carriern und Service Providern ausmachen. So kann in Frame Relay der DLCI und in ATM die VPI/VCI-Kombination als MPLS-Label benutzt werden. Durch die limitierte Größe ist natürlich in diesem Fall das Label Stacking in seiner Verschachtelungstiefe eingeschränkt.

Auch im Layer-3-Header können je nach Protokoll Labelinformationen eingetragen werden, so könnte man sich das Flow-Label-Feld des IPv6-Protokolls durchaus für diesen Zweck vorstellen.

Die Control-Komponente von MPLS erzeugt die Bindung der Labels an bestimmte Routen, Flüsse oder Multicast-Bäume. Auch die Verteilung der notwendigen Informationen an die jeweils beteiligten LSR fällt in den Funktionsumfang dieser Komponente und wird

über ein Label-Distribution-Protokoll (LDP) realisiert. Falls Routen über Protokolle wie OSPF oder BGP bestimmt werden, muss der LSR diese Protokolle verstehen und seine Label Information Base (LIB) mit Informationen aus deren Forwarding-Tabellen füllen.

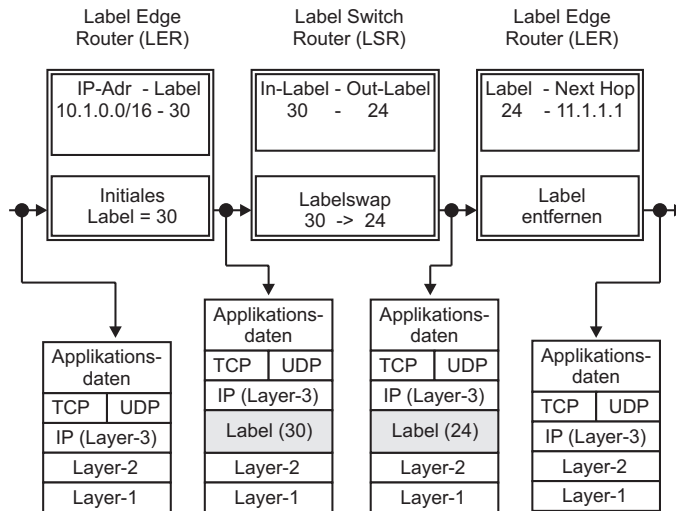


Abbildung 3.11: Das Austauschen der Labels

MPLS-Systeme werden in zwei Kategorien eingeteilt, abhängig davon, an welcher Stelle eines MPLS-Netzwerks sie eingesetzt werden:

Label Edge Router (LER) werden an den Grenzen von MPLS-Netzwerken eingesetzt. Sie versehen eingehende Pakete mit einem Label und entfernen Labels von ausgehenden Paketen. Sie können auch Prozesse zur Erzeugung eines Label Switched Path (LSP) einleiten, mit dem der Weg bestimmter Pakete durch ein MPLS-Netzwerk festgelegt wird.

Label Switch Router (LSR) sind Systeme, die ankommende Pakete aufgrund existenter Label weiterleiten (Forwarding) und die Label dabei durch andere ersetzen (Label Switching). Die Tabellen, aufgrund derer dieses Forwarding und Label Switching durchgeführt wird, reflektieren die verschiedenen LSPs und müssen vorher durch geeignete Signalisierungsprotokolle festgelegt worden sein.

Da in großen Netzen viele Datenpakete eine identische Behandlung hinsichtlich ihres Weges durch das Netz oder ihrer Priorität erfahren sollen, hat man in MPLS ein wichtiges Konzept zur Verkehrsaggregation eingeführt, die so genannte Forwarding Equivalence Class (FEC). Aufgrund verschiedener Verkehrs- und Qualitätskennzeichen eingehender Pakete kann ein LER diese einer bestimmten FEC zuordnen, um damit zusammengehörende Pakete über einen einzigen LSP zu transportieren. Das Forwarding in einem LSR geschieht in der Regel ausschließlich aufgrund des Labels, höhere Protokolle werden nicht ausgewertet, auch nicht bei Routern, die gleichzeitig sowohl auf MPLS- als auch auf IP-Ebene arbeiten.

Die Quality-of-Service-Thematik ist auch in MPLS ausreichend berücksichtigt worden. So erlaubt MPLS insbesondere auch die Konfiguration von Label-Bindungen, die nicht

mit dem zielbasierenden, z.B. durch Routing ermittelten, LSP übereinstimmen, beispielsweise um kritischen Verkehr durch breitbandige, verzögerungsfreie Verbindungen zu leiten. MPLS kann auch aufgrund von höheren Schichten Pakete klassifizieren, die drei Bits im Label mit dem Namen EXP (Experimentell, vgl. Abbildung 3.8) eignen sich zur Aufnahme von Class-of-Service-Informationen z.B. aus dem DSCP im IP-Header. Das oben erwähnte Konzept der FEC kann ebenfalls zur Aggregation von Paketen der gleichen QoS-Klasse eingesetzt werden.

Da in Carrier-Netzen teilweise andere Prämissen als in normalen Kundennetzen gelten, ist die Möglichkeit des Traffic Engineerings in MPLS sehr wichtig. Sie erlaubt dem Administrator, explizite Pfade festzulegen, um zum Beispiel QoS für bestimmte Klassen von Paketen zu garantieren oder Durchleitungsverträge mit anderen Providern umzusetzen, ohne mit dynamischem Routing in Konflikt zu kommen.

MPLS ist eine reine Carrier- und Service-Provider-Technologie und wird auch von diesen zunehmend eingesetzt. Einzige Ausnahme sind sehr große Unternehmen mit großen, selbst betriebenen Netzen, also Firmen mit, hinsichtlich ihres Netzwerks, Carrier-ähnlichen Eigenschaften. Diese Unternehmen sind natürlich auch potenzielle MPLS-Anwender.

An dieser Stelle ist es angebracht, ein paar Worte über die ziemlich sinnlose, weil falsche Diskussion zu verlieren, ob man besser IPsec oder MPLS zum Aufbau von VPNs einsetzt. Da zeugt schon die Frage selbst von Inkompetenz, denn IPsec ist ein Security-Protokoll, lediglich mit der Option versehen, über IP-in-IP Encapsulation (Layer-3-Tunnel) ein VPN aufzubauen. IPsec als Ende-zu-Ende-Protokoll findet hauptsächlich im Enterprise-Bereich Anwendung. MPLS hingegen ist ein infrastruktur-unabhängiges Switching-Protokoll, mit dem man aufgrund seiner Funktionalität sehr flexibel Layer-2-VPNs aufbauen kann. MPLS ist eine Technologie, die praktisch ausschließlich von Carriern und Providern eingesetzt wird.

Die Frage, wenn überhaupt, müsste richtig lauten: Soll man, wenn man MPLS einsetzt, zusätzlich noch IPsec als Sicherheitsprotokoll verwenden oder nicht?

Diese Frage ist aber nicht uninteressant, denn die meisten Unternehmen setzen ja selbst gar kein MPLS ein, sondern der Carrier, der seinen Kunden ein damit aufgebautes VPN zur Verfügung stellt. Das hat aber noch nichts mit Sicherheit im eigentlichen Sinn (Vertraulichkeit, Integrität und Authentizität von Daten) zu tun, denn MPLS bietet hierfür keine Mittel, die muss der Kunde selbst oder sein Provider mittels zusätzlicher Dienste bereitstellen – meist mit IPsec.

3.3 Nichtstandardisierte Tunneling-Protokolle

Neben den standardisierten Tunneling-Protokollen gibt es eine ganze Reihe anderer Protokolle, die jedoch niemals in Form eines verbindlichen Standards das Licht der Welt erblickt haben. Manche dieser Protokolle wurden zwar in Standardisierungsprozesse eingebracht, haben diese aber niemals vollständig durchlaufen oder existieren nur als informeller Standard. Das Point-to-Point Tunneling Protocol (PPTP) oder Layer 2 Forwarding (L2F) sind Beispiele hierfür. Andere Protokolle sind proprietär und werden nur innerhalb der Produktlinien bestimmter Hersteller verwendet, wie zum Beispiel der

Altavista Tunnel, der Bay Dial VPN Service (Bay-DVS) oder das Ascend Tunnel Management Protocol (ATMP). Manche dieser proprietären Protokolle schmücken sich zwar mit RFC-Nummern, allerdings sind diese nur informeller Natur und repräsentieren keinen verbindlichen Internetstandard.

3.3.1 Layer 2 Forwarding (L2F)

Das Layer 2 Forwarding ist ein Layer-2-Tunneling-Protokoll, das hauptsächlich von der Firma Cisco zum Einsatz im Provider-Enterprise-Modell entwickelt wurde. Es gibt praktisch keine Client-Implementierungen, sondern nur Software-Module für Remote-Access-Konzentratoren und Router. L2F ist sehr eng mit L2TP verwandt, viele L2TP-Network-Server können auch als Endpunkt für einen L2F-Tunnel dienen.

L2F bietet keine Sicherheitsdienste wie Datenverschlüsselung oder starke Authentifizierung, kann aber, wie auch L2TP und PPTP, verschiedene Netzwerkprotokolle tunneln.

3.3.2 Point-to-Point Tunneling Protocol

Das Point-to-Point Tunneling Protocol wurde von einer Reihe von Firmen entwickelt, die zu diesem Zweck das so genannte PPTP-Forum gründeten. Der Software-Riese Microsoft war, neben Unternehmen wie Ascend Communications, 3Com und anderen, maßgeblich an der Entwicklung beteiligt und hat seine PPTP-Client-Software in alle seine Betriebssysteme integriert bzw. für ältere Systeme wie Windows 95 Updates auf den Markt gebracht. PPTP kann sowohl als Basis für das Ende-zu-Ende-Modell dienen als auch durch Implementierungen in Remote-Access-Konzentratoren im Provider-Enterprise-Modell eingesetzt werden. PPTP hat einen dem L2TP ähnlichen Aufbau. Für den Steuerungskanal verwendet PPTP jedoch das TCP-Protokoll. Als Layer-2-Protokoll kapselt PPTP PPP-Rahmen mit einem modifizierten GRE-Header ein.

PPTP bietet auch einige Sicherheitsfunktionen wie Datenverschlüsselung (MPPE, Microsoft Point-to-Point Encryption) und Benutzerauthentifizierung, die allerdings bei weitem nicht die Stärke von IPSec aufweisen. Insbesondere die Ableitung des Schlüssels, mit dem die Daten verschlüsselt werden, aus dem Benutzerkennwort war in der Vergangenheit Zielscheibe einiger erfolgreicher Angriffe. Der Schlüssel des eingesetzten RC4-Protokolls ist zwar, je nach der Version von PPTP, entweder 40 oder 128 Bit lang – aber er wird aus einem Hashwert des Benutzerpassworts erzeugt. Also muss ein Brute-Force-Angriff (vgl. Kapitel 4) nicht alle 240 oder 2128 möglichen Schlüssel testen, sondern braucht praktisch nur einen Wörterbuchangriff auf das Benutzerpasswort durchzuführen. Microsoft hat mit einem Update auf die erfolgreichen Angriffe reagiert und eine neue, sicherere Authentifizierung namens MS-CHAPv2 implementiert. Aus Kompatibilitätsgründen kann das neue Verfahren beim Verbindungsaufbau jedoch durch die alte Version ersetzt werden, was eine nicht unbeträchtliche Angriffsfläche bildet. In jedem Fall basiert die Sicherheit der Verschlüsselung in PPTP auf der Sicherheit des Benutzerpassworts – und das ist vielen Anwendern ein zu hohes Sicherheitsrisiko. Viele gute Sicherheitsstrategien legen darüber hinaus auch gewisse Verfahren und Richtlinien zur Schlüsselerzeugung und -verwaltung fest, so dass der Einsatz von PPTP oft schon an dieser Hürde scheitert.

Auch Microsoft hat sich von seinem Problemprotokoll PPTP verabschiedet. Seit Windows 2000 werden verschiedene Tunneling-Protokolle – nach wie vor auch noch PPTP – angeboten, jedoch wird für neue Installationen die modernere und sicherere Variante L2TP/IPSec (siehe Kapitel 8) empfohlen. In Provider-Enterprise-Modellen setzen die Service Provider zunehmend auf L2TP, so dass auch hier PPTP immer weiter verdrängt wird.

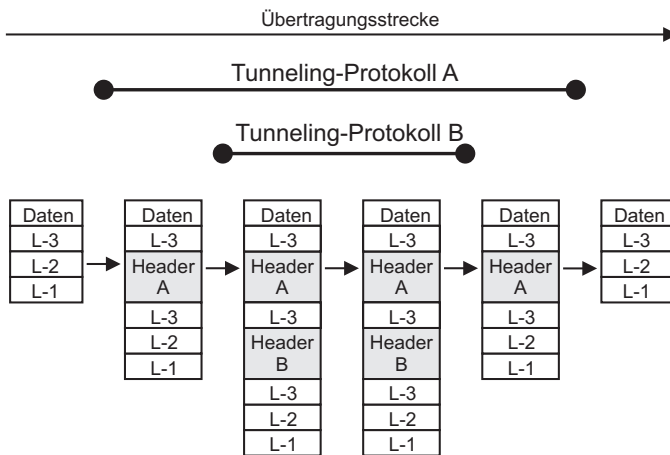


Abbildung 3.12: Man kann auch Tunneling-Protokolle ineinander verschachteln.

3.4 Verschachtelte Tunneling-Protokolle

Manchmal bietet ein Tunneling-Protokoll allein nicht alle Funktionen, auf die man großen Wert legt. Die in diesem Kapitel besprochenen Layer-2-Tunneling-Protokolle können verschiedene Netzwerkprotokolle wie IP, IPX usw. tunneln, zeichnen sich aber nicht gerade durch gute Sicherheitsfunktionen aus. IPSec auf der anderen Seite hat hervorragende Sicherheitsfunktionen, kann aber nur IP tunneln. Was liegt also näher, als zwei verschiedene Tunneling-Protokolle miteinander zu kombinieren. Das kann man auch tun, indem man Tunnel ineinander verschachtelt. Wie dies prinzipiell funktioniert, sehen Sie in Abbildung 3.12. Beim Verschachteln von Tunneln ist es wichtig, dass sich die Tunnel, wie Sie in der Abbildung sehen, nicht überlappen – die Tunnel müssen vollständig ineinander enthalten sein. Man kann grundsätzlich sowohl Protokolle gleichen als auch verschiedenen Typs ineinander verschachteln.

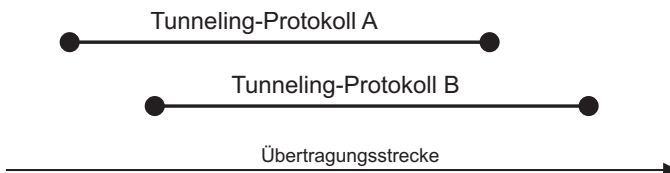


Abbildung 3.13: Das geht nicht, verschiedene Protokolle müssen sich immer vollständig umschließen.

Nehmen wir einmal als Beispiel aus der Praxis ein Remote-Access-VPN, um den Sinn und den Einsatz der Verschachtelung von Tunneling-Protokollen zu erläutern. IPSec ist ein Sicherheitsprotokoll, das man gern im Ende-zu-Ende-Modell einsetzt, um die gewünschte Sicherheitsfunktionalität möglichst auf der ganzen Übertragungsstrecke zwischen Client und VPN-Gateway zur Verfügung zu haben. Dabei gibt es aber möglicherweise ein Problem: Das Ende-zu-Ende-Modell bedeutet immer, dass der Client freiwillig tunnelt. Ein Benutzer könnte somit aber auf seinem Firmen-PC seinen Dialer manuell starten und sich zum ISP verbinden, ohne seinen IPSec-Client zu benutzen. Das kann unter Umständen von einer Sicherheitsstrategie verboten sein.

Mit dem Provider-Enterprise-Modell, zum Beispiel mit L2TP als Tunneling-Protokoll, wäre das nicht möglich, denn hier erfolgt ein zwangsweises Tunneling. Der Remote-Access-Konzentrator beim Service Provider kann z.B. aufgrund der angerufenen Nummer (also der Einwahlnummer des ISP, die der Dialer anrufen muss) entscheiden, ob und wohin ein Tunnel aufgebaut werden muss. Der Client kann dies nicht mehr beeinflussen. Allerdings ist L2TP kein Sicherheitsprotokoll.

Die Lösung ist eine Verschachtelung von IPSec und L2TP. In Abbildung 3.14 sehen Sie, wie dies realisiert wird: IPSec wird im Ende-zu-Ende-Modell betrieben und garantiert so die geforderte Sicherheit. Mit L2TP im Provider-Enterprise-Modell wird garantiert, dass, sobald sich der Client beim Provider einwählt, er zwangsweise mit L2TP bis zum LNS im Kundennetzwerk getunnelt wird. Diese Lösung hat allerdings auch ihren Preis in Form eines zusätzlichen Protokoll-Overheads in den Datenpaketen.

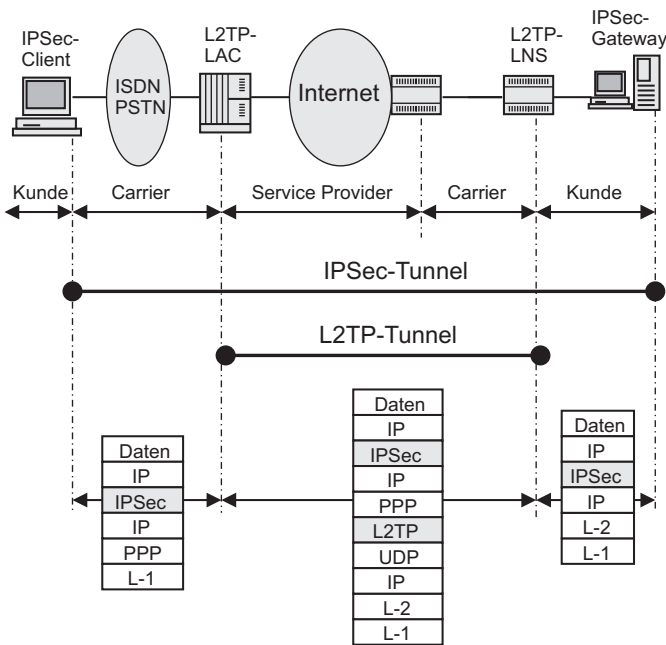


Abbildung 3.14: Eine Verschachtelung von L2TP und IPSec

Im Zusammenhang mit verschachtelten Tunneling-Protokollen fällt fälschlicherweise auch oft der Begriff „L2TP über IPSec“ oder „IPSec secured L2TP“ oder kurz „L2TP/IPSec“. Hierbei handelt es sich jedoch nicht um eine Verschachtelung von Tunneling-Protokollen, da IPSec hierbei nicht im Tunnel-, sondern im Transport-Modus arbeitet. Hier hat man die Vielfalt der Protokolle, die L2TP tunneln kann, mit der Sicherheit von IPSec kombiniert. L2TP wird dabei zum Tunneln benutzt, und die erzeugten UDP/IP-Pakete werden mit IPSec im Transport-Modus verschlüsselt und authentifiziert. Diese Kombination ist in Kapitel 8 ausführlich beschrieben. Sie wird auch seit Windows 2000 von Microsoft unterstützt.

3.5 Welches Protokoll für welchen Zweck?

Diese Frage ist eigentlich gar nicht so kompliziert, da verschiedene Protokolle für bestimmte Funktionen von vornherein ausscheiden. Wenn Sie zum Beispiel IPX über ein IP-Netzwerk transportieren müssen, scheidet IPSec an diesem Punkt bereits aus. Die Gegenüberstellung im folgenden Abschnitt soll die Protokolle nicht wertend miteinander vergleichen, denn sie wurden teilweise mit völlig unterschiedlichen Zielsetzungen und Einsatzszenarien entwickelt.

In Tabelle 3.1 sind die vier wichtigsten Tunneling-Protokolle – IPSec, L2TP, PPTP und MPLS – gegenübergestellt. Hier soll aber wie gesagt kein Vergleich im Sinne von gut oder schlecht stattfinden, sondern ein Vergleich der wichtigsten Funktionen der Protokolle. Hier können Sie anhand eines Pflichtenhefts oder Anforderungskatalogs schnell ein geeignetes Protokoll finden – oder eine Kombination von zwei Protokollen.

	IPSec	L2TP	PPTP	MPLS
Paketauthentifizierung	Ja	Ja	Nein	Ja
Benutzerauthentifizierung	Ja	Nein	Nein	Nein
Datenverschlüsselung	Ja	Ja	Ja	Nein
Schlüsselmanagement	Ja	Nein	Nein	Nein
QoS-Signalisierung	Ja	Nein	Nein	Ja
IP-Tunneling	Ja	Ja	Ja	Ja
IPX-Tunneling	Nein	Ja	Ja	Ja
Primäres Modell	Ende-zu-Ende	Provider-Enterprise	Ende-zu-Ende	Intra-Provider

Tabelle 3.1: Die wichtigsten Tunneling-Protokolle im Vergleich. Die Anzahl der „Ja“ oder „Nein“ ist keine Wertung, da die Protokolle sehr unterschiedliche Ausrichtungen haben.

Mit der Auswahl eines oder mehrerer Tunneling-Protokolle treffen Sie eine wichtige Entscheidung während der Planung des VPN. In heutigen, modernen Netzwerken ist man bestrebt, möglichst auf Standards oder wenigstens auf De-facto-Standards zu setzen. Aus diesem Grund reduziert sich die Menge der zur Auswahl stehenden Tunneling-Protokolle ganz erheblich: Es bleiben die in Tabelle 3.1 beschriebenen Verfahren, aus denen man ein geeignetes Protokoll auswählen sollte. Leider gibt es dafür keine allgemein gültige Checkliste oder eine Art VPN-Kochbuch, denn es spielen einfach zu viele unternehmensspezifische Planungs- und Entscheidungskriterien eine Rolle.

Es kann sich durchaus auch als sinnvoll zu erweisen, mehrere Protokolle miteinander zu kombinieren. So kann es sich einerseits aus Sicht der Infrastruktur durchaus als sinnvoll erweisen, von einem Provider oder Carrier seine Standorte per MPLS-VPN verbinden zu lassen, andererseits erfordern gesetzliche Vorgaben oder bestimmte Sicherheitsvorgaben die Verschlüsselung und Integritätssicherung der Daten durch IPSec.

Die wichtigsten Faktoren zur Auswahl sind:

- Die Netzwerkprotokolle, die übertragen werden müssen
- Die Sicherheitsstrategie des Unternehmens
- Die eingesetzten Applikationen
- Benötigte Client-Unterstützung (Betriebssysteme)
- Der VPN-Typ
- Kompatibilitätsanforderungen

Netzwerkprotokolle

Dieser Faktor ist neben der Sicherheitsstrategie der am meisten ausschlaggebende. Unter Umständen sollten Sie auch Ihre mittel- und langfristige Netzwerkstrategie zurate ziehen, ob nicht vielleicht eine Protokollkonsolidierung stattfinden soll – und nur noch IP übrig bleibt. Diese Tendenz ist im Augenblick bei fast allen Unternehmen zu beobachten; bei etlichen ist dieser Prozess auch schon abgeschlossen. Ein VPN ist in diesem Zusammenhang möglicherweise ein guter Punkt, um mit dieser Konsolidierung zu beginnen und nur noch IP zu tunneln. Jetzt werden Sie vielleicht einwenden, dass diese Überlegung nichts gebracht hat, denn alle vier Protokolle tunneln IP. Das ist richtig, aber es bleiben mehr Alternativen, um die nachfolgenden Kriterien erfüllen zu können.

Sicherheitsstrategie

Die Netzwerkabteilung eines Unternehmens ist in der Regel dafür verantwortlich, die Sicherheitsstrategie des Unternehmens im Netzwerkbereich zu implementieren. Im Fall von Internet-VPN entstehen dadurch oft ganz bestimmte Anforderungen an die Sicherheitsdienste, die ein geeignetes Tunneling-Protokoll zur Verfügung stellen muss. An diesem Punkt scheiden schon Protokolle wie MPLS und L2TP aus, es sei denn, sie werden mit entsprechenden Sicherheitsprotokollen kombiniert. Wenn die Anforderungen an die Sicherheit der Datenübertragung sehr hoch sind, bleibt praktisch keine Alternative zu IPSec. Müssen unbedingt auch andere Netzwerkprotokolle als IP getunnelt werden, kann man im Clientbereich L2TP/IPSec und im Bereich von Standortverbindungen MPLS mit IPSec auf IP-Ebene verwenden.

Applikationen

Hier ist nicht von Interesse, welche Applikationen benutzt werden, sondern welche Anforderungen an Verzögerungszeiten und Bandbreite sie stellen. Es ist auch von Interesse, ob im privaten Netzwerk bereits Quality-of-Service-Dienste auf IP-Ebene eingesetzt werden und ob diese Informationen an den Service Provider gesendet werden sollen. Hier spielt in gewissem Maße auch schon die Auswahl der VPN-Systeme mit hinein, auf die in Kapitel 11 näher eingegangen wird.

Client-Unterstützung

Wenn, zum Beispiel in einem Remote-Access-VPN, ein Ende-zu-Ende-Modell eingesetzt wird, müssen für das entsprechende Betriebssystem auch Client-Implementierungen des auszuwählenden Tunneling-Protokolls existieren. Dies dürfte für L2F schon das Aus sein; hier muss man sich zwischen den restlichen drei Protokollen entscheiden.

VPN-Typ

Natürlich ist es auch wichtig, welcher VPN-Typ zum Einsatz kommt, ob man also ein Remote-Access-, ein Branch-Office- oder ein Extranet-VPN aufbauen will – oder beliebige Kombinationen davon. Die Tunneling-Protokolle haben ihre spezifischen Schwerpunkte und sind für bestimmte Einsatzszenarien nicht geeignet. So bieten nur ganz wenige Service Provider im Augenblick ein Provider-Enterprise-Modell mit zwangsweisem Tunneling mit IPSec an. Und Branch-Office-VPNs mit L2F fallen auch unter die Rubrik exotisch. Und im Extranet-Bereich macht sich ohnehin ein Protokoll breit, das selbst gar kein Tunneling unterstützt: SSL.

Kompatibilität

Insbesondere wenn das VPN auch Extranet-Dienste zur Verfügung stellen muss, ist die Kompatibilität ein ganz wesentliches Entscheidungskriterium. Denn dann muss ein Tunneling-Protokoll eingesetzt werden, das die Organisationen, die angebunden werden sollen, auch benutzen. Natürlich muss nicht notwendigerweise nur ein einziges Tunneling-Protokoll im VPN eingesetzt werden. Falls man aus bestimmten Gründen zum Beispiel sein VPN mit PPTP aufbaut, die externen Extranet-Systeme aber IPSec einsetzen, kann man für diesen genau abzugrenzenden Bereich auch IPSec einsetzen. Falls die Forderung nach Kompatibilität jedoch einer grundsätzlichen Strategie des Unternehmens entspringt – mit dem Ziel, eine möglichst hohe Interoperabilität und Herstellerunabhängigkeit zu erzielen –, sollte man standardisierte Protokolle wie IPSec oder L2TP einsetzen.

4 Sicherheitstechnologie

In diesem Kapitel werden die Sicherheitsaspekte beim Einsatz von VPN betrachtet und die hierfür eingesetzten Technologien ausführlich behandelt. Zuerst erfolgt ein allgemeiner Überblick, mit welchen Techniken die Sicherheitsanforderungen erfüllt werden, die wir im vorangegangenen Kapitel an virtuelle private Netzwerke gestellt haben.

Anschließend werden die Grundlagen für diese Technologien ausführlich behandelt, wobei jedoch immer der aktuelle Bezug zum Thema VPN bestehen bleibt. Insbesondere werden nur solche Algorithmen und Verfahren besprochen, die in aktuellen Netzwerk-Sicherheitsprotokollen verwendet werden, so dass hier auch nur Teilaspekte der Kryptologie behandelt werden. Allgemeine und umfassende Grundlagen zum Thema Kryptologie und verwandte Themen finden Sie in der weiterführenden Literatur hierzu.

Weiterhin haben es aktuelle Entwicklungen in der Kryptoanalyse notwendig gemacht, den aktuellen Erkenntnisstand zur Sicherheit bestimmter Verfahren und Algorithmen zu erläutern und die Auswirkungen auf VPN zu diskutieren. Insbesondere neueste Erkenntnisse im Bereich der Hash-Funktionen und verschiedene Fortschritte in der Analyse bestimmter Verschlüsselungsalgorithmen haben es notwendig gemacht, bestimmte Dinge bei der Auswahl und dem Einsatz dieser Technologie neu zu überdenken.

4.1 Sicherheit in VPN

Dieses Thema kann und darf nicht isoliert betrachtet werden. Es ist ein integraler Bestandteil der Netzwerksicherheit, die wiederum nur ein kleiner Teil einer unternehmensweiten Sicherheitsstrategie ist. VPN-Systeme haben in heutigen Netzwerken zudem zahlreiche Schnittstellen zu anderen Systemen.

4.1.1 Sicherheit in Unternehmensdatennetzen

Der Aspekt Sicherheit spielte in der Vergangenheit aufgrund der Begrenzung der Datennetze auf das Unternehmensgelände und der noch übersichtlichen Anzahl von Weitverkehrsverbindungen eine eher untergeordnete Rolle. Ausgelöst durch veränderte Arbeits- und Kommunikationsweisen beginnt sich dies zu ändern:

Die Zahl der so genannten Remote-Access-Nutzer, also der Mitarbeiter, die von unterwegs, von einer Außenstelle oder von zu Hause aus Zugriff auf Daten des Firmennetzes benötigen, steigt kontinuierlich. Damit ist das Firmennetz nicht mehr länger auf den Campus begrenzt, sondern es muss eine Vielzahl von Remote-Access-Verbindungen eingerichtet, betrieben und überwacht werden.

Immer mehr Unternehmen öffnen ihre Firmennetze für Zulieferer, Partner und Kunden bzw. bieten ihre Produkte auf elektronischem Weg zum Verkauf an. Hierfür sind nationale und internationale Verbindungen notwendig. Da der Aufbau bzw. das Anmieten eines eigenen Leitungsnetzes aber für die wenigsten Firmen rentabel ist, werden für solche Anwendungen vermehrt öffentliche Netze wie das Internet oder Teilkapazitäten öffentlicher Netze als so genannte virtuelle private Netze genutzt.

Gleichzeitig hat das Netz für den wirtschaftlichen Erfolg des Unternehmens einen weit höheren Stellenwert als in der Vergangenheit. Denn je mehr geschäftskritische Anwendungen darüber miteinander kommunizieren und je mehr unternehmenskritische Ressourcen daran angeschlossen werden, desto mehr wird das Netz selbst zum geschäftsbestimmenden Faktor.

Damit stellt sich automatisch die Frage nach der Datensicherheit und anderen mit ihr zusammenhängenden Aspekten wie Sicherung des Netzzugangs, Systemsicherheit auf Rechner- und Programmebene oder Zugriffsschutz auf gespeicherte Informationen.

Datensicherheit ist ein komplexes Thema. Zum einen, weil die Sicherheitstechnologien selbst komplex sind und unterschiedliche Funktionen – und damit einen unterschiedlichen Grad an Sicherheit – bieten. Zum anderen, weil die Datensicherheit immer auch vom Gesamtsicherheitskonzept des Unternehmens betroffen ist. Insofern muss der Einführung von Sicherheitstechnologien eine Sicherheits- beziehungsweise Gefahrenanalyse vorausgehen, die wenigstens folgende Fragen beantworten sollte:

Was ist wem gegenüber wie lange vertraulich zu behandeln?

Welche Maßnahmen müssen von wem dementsprechend eingeleitet werden, und was dürfen sie kosten?

Welche Risiken bestehen für das Unternehmen, und wie sind sie zu bewerten?

Geschützt werden müssen im Wesentlichen die Systemhardware und -software sowie die Daten selbst. Dabei geht man meist davon aus, dass die Daten am meisten bedroht sind. Die Hardware kann in Sicherheitsbereichen oder räumlich getrennt im Standby-Mode betrieben werden, so dass Angriffe, sofern sie überhaupt möglich sind, schnell bemerkt und aufgefangen werden können. Das Gleiche gilt für die Software, die auf diesen Systemen läuft.

Viele Daten müssen vertraulich sein, also vor Ausspähung geschützt werden. Und sie müssen integer sein, also vor unerlaubten Veränderungen geschützt werden. Gleichzeitig müssen Daten aber auch verfügbar sein. Hier liegt das Problem, denn 100% sichere Daten sind nicht mehr verfügbar, und 100% verfügbare Daten sind nicht mehr sicher. Als praktikable Lösung bleibt demzufolge nur ein Kompromiss zwischen Verfügbarkeit und Sicherheit.

Aus diesem Grund muss der Einführung von Sicherheitstechnologien zunächst eine Datenanalyse vorausgehen, die Fragen wie diese beantwortet:

Welche Arten von Daten sind im Unternehmen vorhanden?

Wie sicher müssen diese Daten sein?

Vor wem müssen sie sicher sein?

Wie hoch sind die Verletzungen der Sicherheit zu bewerten?

Wie lange muss die Sicherheit, insbesondere die Vertraulichkeit, gewährleistet sein – Stunden, Tage, Wochen, Monate, Jahre oder Jahrzehnte?

Zu welchem Preis (Kosten, Verfügbarkeit, Benutzbarkeit) dürfen diese Ziele erreicht werden?

Die Konzernbilanz eines Unternehmens muss in der Regel nur kurze Zeit vertraulich behandelt werden, und selbst wenn während dieser Zeit Einzelheiten an die Öffentlichkeit dringen, wäre das Unternehmen dadurch nicht unbedingt existenziell gefährdet. Wenn andererseits aber Firmengeheimnisse wie Entwicklungsdaten, Patente oder vergleichbar sensible Informationen bekannt werden, kann das Unternehmen erheblichen Schaden davontragen, ja sogar in seiner Existenz gefährdet sein.

Ebenso hat die Analyse, von wem ein Unternehmen bedroht wird, starke Auswirkungen auf die einzuführende Sicherheitstechnologie. Eine Firma, die von einer Privatperson angegriffen wird, kann sich mit weitaus weniger Aufwand schützen als eine, die befürchten muss, von einem Konkurrenzunternehmen oder gar von einem Geheimdienst ausgespäht zu werden. Denn Letzteren stehen für einen Angriff ganz andere finanzielle Mittel und technische Ressourcen zur Verfügung als einer Privatperson. Entsprechend mehr Sicherheit zu höheren Kosten wird sich das betroffene Unternehmen leisten müssen.

Aspekte dieser Art wirken sich direkt auf bestimmte technische Details aus, zum Beispiel auf die Schlüssellänge bei der Datenverschlüsselung, auf das einzusetzende Verschlüsselungsverfahren oder auf die Kombination von mehreren Sicherheitstechnologien.

Nicht zu unterschätzen ist darüber hinaus der Einfluss, den die Anschaffung neuer sicherheitsrelevanter Technologien wie zum Beispiel virtueller privater Netzwerke auf bereits vorhandene Einrichtungen hat, vor allem wenn durchgängige Sicherheitskonzepte in einer verteilten Systemumgebung realisiert werden sollen. Beispielsweise können aus technischen Gründen bestimmte Authentifizierungsverfahren nicht zusammen mit bestimmten Benutzer- oder Passwort-Verwaltungssystemen eingesetzt werden. Um auszuschließen, dass sich Neuanschaffungen negativ auf vorhandene Technologien auswirken oder gar neue Bedrohungen nach sich ziehen, sollte vorab geklärt werden:

- wie die augenblickliche Systemsicherheit, insbesondere die Zugriffskontrolle auf Rechner- und Programmebene, realisiert ist,

- wie die Datensicherheit, also die Zugangskontrolle auf Dateiebene, organisiert ist,

- ob und welche Zugangskontrollen es zu Gebäuden, Räumen oder Zonen gibt und

- welche gesetzlichen Auflagen und betrieblichen Zwänge bei der Einführung berücksichtigt werden müssen.

Außerdem müssen verbindliche Richtlinien darüber existieren, wer für die Sicherheit verantwortlich ist. Hiervon sind alle Mitarbeiter einer Organisation betroffen, von den Systemadministratoren und Netzverantwortlichen bis hin zu jedem Endbenutzer. Denn was nützt das ausgefeilteste und teuerste Netz- und Systemzugangskonzept, wenn der Mitarbeiter sein Passwort an den PC-Bildschirm klebt?

4.1.2 Sicherheitsverfahren in VPN

Ein entscheidendes Kriterium beim Entwurf einer Kommunikationsinfrastruktur, die öffentliche Netze benutzt, sei es ISDN, UMTS, GSM, ein Provider-Netzwerk oder das Internet, ist die Sicherheit der zu übertragenden Daten. Diese müssen insbesondere gegen Verletzungen der Vertraulichkeit und der Integrität hinreichend geschützt werden, gefolgt von flankierenden Maßnahmen zum Schutz vor Denial-of-Service- und Replay-Angriffen und weiteren ähnlich gearteten Attacken.

Die Anforderungen an die Sicherheit eines VPN leiten sich aus der Sicherheitsstrategie (Security Policy) oder den allgemeinen Anforderungen an die Sicherheit des Unternehmensnetzwerks ab.

In der Vergangenheit wurde dies, meist weil die geeignete Technologie fehlte, durch die Verschlüsselung von einzelnen Teilstrecken, durch die Verbindungsverschlüsselung (Link-Encryption) oder die Verschlüsselung auf Applikationsebene gelöst. Mit der Verbindungsverschlüsselung werden die Verbindungen zwischen Vermittlungs- oder Übertragungssystemen oder die Pakete auf der Verbindungsschicht, der Schicht 2 des OSI-Schichtenmodells, verschlüsselt. Das immanente Sicherheitsproblem hierbei liegt in der Dynamik heutiger Netzwerke. Auf der Ebene der Netzwerk-Infrastruktur werden laufend neue Strecken hinzugefügt oder geändert, und die Vermittlungssysteme, die zwischen den verschlüsselten Links sitzen, befinden sich nicht im Einflussbereich des Endteilnehmers beziehungsweise des Endkunden. Auf diesen Vermittlungssystemen liegen die Daten aber unverschlüsselt vor, da ja nur auf den Verbindungswegen selbst verschlüsselt wird. Ein weiteres Problem der Verbindungsverschlüsselung ist die automatische Wegewahl (Routing) beim Transport von Paketen durch ein Netzwerk, die bei modernen gerouteten Netzwerken nicht mehr vorherbestimmbar ist. Somit besteht die Gefahr, dass Daten über nicht verschlüsselte Teilstrecken transportiert werden.

Vom Standpunkt der Datensicherheit wäre die beste Methode eine Ende-zu-Ende-Verschlüsselung auf Applikationsebene. Dies ist aber technisch und organisatorisch nicht mehr handhabbar, insbesondere wenn man viele verschiedene Applikationen einsetzt und diese einer gewissen Eigendynamik unterworfen sind.

Eine mittlerweile sehr verbreitete Lösung heißt hier Sicherheit auf Netzwerkebene, also auf der Schicht 3 des OSI-Schichtenmodells. Hier kann man, je nach Anforderung und Infrastruktur, auch gemischt Sicherheit von Rechner zu Rechner, von Rechner zu Netzwerk oder von Netzwerk zu Netzwerk realisieren. Die Sicherheit auf höheren Ebenen anzusiedeln, hat für ein virtuelles privates Netzwerk schon wieder mehr Nach- als Vorteile, da hierbei wieder niemals alle Daten verschlüsselt werden können. Ein VPN ist ein Netzwerk oder zumindest ein Teil eines Gesamtnetzwerks, also ist die notwendige Sicherheitstechnologie auch auf dieser Ebene, der Netzwerkebene, anzuwenden.

Da die Sicherheit des Datentransports in einem IP- oder Internet-VPN so wichtig ist, werden einige Aspekte und Lösungen im Folgenden detailliert erläutert.

Bevor man überlegt, auf welcher Schicht die für VPN notwendigen Sicherheitsfunktionen implementiert werden sollen, sollte festgelegt werden, was dies alles für Funktionen sind:

Die Vertraulichkeit der übertragenen Daten muss sichergestellt werden. Dritte dürfen nicht an die übertragenen Informationen gelangen.

Die Daten müssen authentisch sein, also nicht von gefälschten Absenderadressen kommen.

Die Daten müssen integer, also nicht von Dritten verändert sein.

Es muss ein Schutz vor wiederholtem Senden von Daten existieren.

Die Gegenstellen bzw. Benutzer müssen sich gegenseitig eindeutig identifizieren.

Die VPN-Systeme selbst müssen gegen Angriffe geschützt sein.

Es gibt mehrere Ansätze zur sicheren Übertragung von Daten in VPN, die sich im Wesentlichen darin unterscheiden, auf welcher Ebene des TCP/IP-Schichtenmodells sie eingesetzt werden. Die unterschiedlichen Möglichkeiten sind:

Verschlüsselung auf Ebene 1, der physikalischen Schicht

Verschlüsselung auf Ebene 2, der Sicherungsschicht

Verschlüsselung auf Ebene 3, der Vermittlungsschicht

Verschlüsselung auf Ebene 4, der Verbindungsschicht

Verschlüsselung auf der Applikationsebene

Für ein Internet-VPN, also ein VPN, das auf dem IP-Protokoll basiert, kommt man sehr schnell zu der Erkenntnis, dass eine Verschlüsselung auf der Ebene des IP-Protokolls das Sinnvollste und Sicherste ist. Warum ist dies so?

Sehen wir uns zuerst einmal die Möglichkeiten an, auf den unteren Ebenen 1 und 2 zu verschlüsseln. Dabei verschlüsseln wir Datenleitungen oder Datenrahmen auf der Sicherungsschicht. Wer eine solchermaßen verschlüsselte Leitung abhört, kann mit den abgefangenen Daten nichts anfangen. Aber in den Vermittlungssystemen liegen die Daten im Klartext vor, da sie in der Schicht 3 verarbeitet werden, an die sie daher unverschlüsselt von der Sicherungsschicht geliefert werden müssen. Jemand, der sich Zugriff auf ein solches System verschafft, kann auch auf die unverschlüsselten Informationen zugreifen. Ein anderes Problem besteht darin, dass in modernen Netzen eine dynamische Wegwahl für die Datenpakete erfolgt, um sie so schnell und so kostengünstig wie möglich zu transportieren. Daraus folgt aber auch, dass der Weg, den die Datenpakete nehmen, nicht vorher bestimmbar ist und deshalb alle theoretisch möglichen Teilstrecken verschlüsselt werden müssen. Das ist aufgrund der Komplexität weder im Internet noch in Frame-Relay- oder ATM-Netzen realisiert und wird dort auch niemals realisiert werden.

Setzen wir nun an anderer Stelle an, und schauen wir uns einmal die Verschlüsselung auf der Applikationsschicht an. Im Extremfall erfolgt die Ver- und Entschlüsselung durch die Anwendungen. Es müssen dabei natürlich alle möglichen Anwendungen mit den entsprechenden Funktionen ausgerüstet werden, um den gesamten Datenverkehr sicher zu machen, was sowohl technisch als auch organisatorisch praktisch nicht realisierbar ist. Auch auf tieferen Sublayern in der Anwendungsschicht bleibt immer das Problem bestehen, dass alle gleichzeitig betriebenen Anwendungen, Protokolle und Verbindungen mit entsprechenden Verschlüsselungsfunktionen ausgerüstet sein müssen, sollen nicht einige Informationen doch noch im Klartext übertragen werden.

Die Ebene 3 weist in einem IP- oder Internet-VPN diese ganzen Nachteile nicht auf, da dort nur ein einziges Protokoll, nämlich das Internet-Protokoll (IP), betrieben wird. Integriert man seine Sicherheitstechnologie in das Internet-Protokoll, dann werden alle

Datenpakete dieses Protokolls von den eingesetzten Sicherheitsverfahren bearbeitet, unabhängig von der darunter liegenden Infrastruktur und unabhängig von den Anwendungen darüber.

Wie eingangs erwähnt, kann man in einigen wenigen, speziellen und klar abgegrenzten Fällen auch ohne IP-Sicherheitstechnologie in Internet-VPN auskommen – und zwar dann, wenn alle Applikationen oder Protokolle, die das VPN benutzen, bereits entsprechende Sicherheitsfunktionen unterstützen. Falls zum Beispiel in einem Remote-Access-VPN die Benutzer ausschließlich webbasierende Anwendungen und E-Mail benutzen, dann kann man bei der Verwendung des SSL-Protokolls und eines sicheren Mailsystems wie zum Beispiel PGP (Pretty Good Privacy, ein kryptografisch geschütztes Mailsystem) die notwendige Sicherheit auch ausschließlich auf höheren Ebenen des OSI-Schichtenmodells ansiedeln. Aber wehe, wenn auch nur einer der Benutzer in einem derart aufgebauten VPN ein anderes Kommunikationsprogramm benutzt. Dann ist alles, was damit übertragen wird, ungesichert!

4.1.3 Sicherheit in der Netzwerkschicht mit IP-Security

Da man beim Design eines VPN auf Skalierbarkeit und Migrationsfähigkeit Wert legen sollte, ist es ein Muss, so weit es irgend möglich ist, mit Standards zu arbeiten. Der Standard für Sicherheit auf der IP-Ebene heißt IP-Security oder kurz IPSec. In ihm und dem damit eng verbundenen IKE-Protokoll (IKE, Internet Key Exchange) sind verschiedene Verschlüsselungs- und Authentifizierungsverfahren sowie Schlüsseltausch- und Schlüsselverwaltungsprotokolle festgelegt, die bei korrekter Implementierung eine hohe Interoperabilität gewährleisten. Im Folgenden werden die für ein IP-VPN essenziellen Sicherheitsmerkmale dieses Protokolls behandelt, in Kapitel 5 werden die technischen Aspekte und die genaue Funktion des IPSec-Protokolls ausführlich beschrieben. Die Details zu den hier angesprochenen Verfahren und Algorithmen folgen in diesem Kapitel.

Datenverschlüsselung

In IPSec können die Daten auf verschiedene Arten verschlüsselt werden, also mit den unterschiedlichsten Algorithmen und unterschiedlichen Schlüssellängen. Wichtig dabei ist, dass sich die beteiligten Gegenstellen immer auf ein Verfahren einigen können. Aus diesem Grund gibt es einen Verschlüsselungsalgorithmus, der von jeder IPSec-Implementierung unterstützt werden muss, den Data Encryption Standard (DES) mit seiner festen Schlüssellänge von 56 Bit. Viele Implementierungen bieten auch optional das stärkere Triple-DES-Verfahren mit einer Schlüssellänge von insgesamt 168 Bit und den neuen AES (Advanced Encryption Standard) mit Schlüssellängen von 128, 192 oder 256 Bit. Um zu vermeiden, dass identische Klartextblöcke nach der DES-Verschlüsselung identische Chiffretextblöcke erzeugen, ist in IPSec das so genannte Cipher-Block-Chaining-Verfahren (CBC) vorgeschrieben, das diesen Mangel von DES behebt.

Datenintegrität und Datenauthentizität

Die Datenintegrität, also der Schutz vor unerlaubter und unbemerkter Veränderung der Datenpakete, wird in IPSec durch spezielle, kryptografisch geschützte Prüfsummen, genannt HMAC (Hash-based Message Authentication Code, Prüfsummen, die durch Ver-

wendung Hash-Funktionen berechnet wurden), erreicht. Das gleiche Verfahren garantiert in Verbindung mit so genannten Pre-Shared Secrets auch die Feststellung der Authentizität eines IP-Pakets. HMACs basieren auf Einwegfunktionen, die aus den zu schützenden Daten und einem Schlüssel einen Wert erzeugen, der nicht mehr zurückzuberechnen ist, auch nicht bei Kenntnis eines der Eingangsparameter. In IPSec werden zwei Funktionen vorgeschrieben, HMAC-MD5 und HMAC-SHA-1, die auf den Hash-Funktionen MD5 und SHA-1 aufgebaut sind und mit symmetrischen Schlüsseln von 128 Bit Länge arbeiten. Auch in diesem Bereich garantiert der Standard eine hohe Interoperabilität.

Schlüsselmanagement

Die Verfahren zur Datenverschlüsselung, Authentifizierung und Integritätsprüfung benötigen auf beiden Seiten einer so genannten IPSec-Sicherheitsassoziation jeweils den gleichen Schlüssel. Der DES-Algorithmus benötigt beispielsweise auf der Empfängerseite den gleichen 56 Bit großen Schlüssel zum Entschlüsseln, mit dem der Sender die Daten verschlüsselt hat. Ebenso benötigen die Hash-Verfahren auf beiden Seiten einen identischen 128 Bit langen Schlüssel zur Integritätsprüfung und Authentifizierung. Wie kommen die Schlüssel dorthin, und wie werden sie geändert? Der IPSec-Standard erlaubt unter anderem die Möglichkeit, dies manuell zu tun.

Manuelle Verfahren oder die Übertragung über ein anderes Medium sind aber, vor allem in großen Netzen, nicht praktikabel und oft unsicher. Die Lösung in IPSec ist ein spezielles Protokoll, das über ein unsicheres Übertragungsmedium, in der Regel die zu verwendende Übertragungsstrecke selbst (In-Band), mit Hilfe eines so genannten asymmetrischen Verschlüsselungsverfahrens die benötigten Schlüssel erzeugt. Dieses Protokoll, IKE (Internet Key Exchange), erledigt nicht nur das Schlüsselmanagement, sondern auch die Aushandlung und Konfiguration der IPSec-Sicherheitsassoziation und deren Authentifizierung.

Schutz vor Denial-of-Service-Angriffen

In IPSec und IKE wurden neben all diesen Diensten auch Verfahren und Protokolle zum Schutz vor einer ganzen Reihe von Angriffen implementiert, die sich gegen das IPSec-System selbst richten. Diese Maßnahmen sind sowohl in den IPSec-Sicherheitsprotokollen als auch im IKE-Protokoll zu finden und schützen vor Denial-of-Service-Angriffen, vor Angriffen durch wiederholtes Senden (Replay-Angriffen) und vor so genannten Man-in-the-Middle-Angriffen, in denen sich ein Angreifer unbemerkt in eine Verbindung einschaltet und diese abhört oder modifiziert. Auch hier wurde durch eine konsequente Standardisierung erreicht, dass diese Funktionen in einem Mindestumfang zwischen jedem IPSec-System vereinbar sind.

Benutzerauthentifizierung

Beim Einsatz von Remote-Access-VPN oder generell von VPNs, an denen Endgeräte wie PCs oder Workstations als Einzelbenutzersystem teilnehmen, ist eine Authentifizierung der Benutzer notwendig. Auf die heute eingesetzten Technologien wird in Kapitel 5 detailliert eingegangen.

An dieser Stelle sei darauf hingewiesen, dass in IPSec nur eine Authentifizierung auf Netzwerkebene stattfindet. Es identifizieren sich also nur die IP-Systeme, nicht die Benutzer oder Applikationen, die diese Systeme zur Kommunikation benutzen. Der Benutzer muss sich also getrennt davon authentifizieren, oder die beiden Authentifizierungen müssen auf eine Art und Weise, die außerhalb des Definitionsbereichs von IPSec liegt, miteinander kombiniert werden.

4.1.4 Sicherheit auf der Transportschicht mit Transport Layer Security (TLS) und Secure Socket Layer (SSL)

In letzter Zeit wird das Thema SSL-VPN sehr hoch gehandelt. Wenn man davon absieht, dass es ein SSL-VPN eigentlich gar nicht gibt, da darin keine Netzwerkpakete (IP-Pakete) getunnelt werden, finden sich für die SSL-Technologie einige durchaus sinnvolle Einsatzgebiete im Remote-Access- und Extranet-Bereich. Basis für diese neue Technologie ist das SSL-Protokoll, das ursprünglich von der Firma Netscape als Sicherheitserweiterung für das allgegenwärtige HTTP entwickelt wurde. Da es außer HTTP allerdings auch noch andere zu schützende Protokolle und Applikationen gibt, muss man einige Maßnahmen treffen, um mit SSL eine ausreichende Gesamtsicherheit zu gewährleisten. Transparente Netzwerksicherheit ist mit einem Protokoll, das oberhalb von TCP/IP arbeitet, natürlich nicht möglich. Aber wer diese überhaupt nicht oder unter ganz bestimmten Bedingungen nicht braucht, sollte sich mit der Alternative SSL-VPN (wir werden es im Weiteren so nennen) ernsthaft befassen.

Obwohl man in Kapitel 7 sieht, dass in SSL im Prinzip die gleichen Algorithmen wie in IPSec benutzt werden, sollte man sich davor hüten, beide Protokolle in einen Topf zu werfen oder gar ohne weitere, flankierende Maßnahmen IPSec durch SSL zu ersetzen. Denn beide Ansätze sind grundverschieden, und in diesem Fall drohen fatale Sicherheitslücken.

TLS ist ein Standard der IETF, der auf der letzten SSL-Version basiert und nur minimal von SSL abweicht. Im Weiteren wird, wie allgemein üblich, nur von SSL gesprochen und lediglich in speziellen Fällen auf TLS Bezug genommen.

Datenverschlüsselung

Im Gegensatz zu IPSec, das die verschiedenen Verfahren für Verschlüsselung, Datenauthentifizierung und Datenintegritätsprüfung frei kombinieren kann, sind unter SSL so genannte Cipher-Suites definiert, die diese Verfahren in bestimmten Kombinationen festgelegt haben. Allerdings gibt es sehr viele Kombinationen, und es finden sich dabei auch sehr sichere. Als Verschlüsselungsalgorithmus finden in der Regel DES mit 40 oder 56 Bit Schlüssellänge, Triple-DES, RC4 mit 40 oder 128 Bit Schlüssellänge und zunehmend auch der neue AES Verwendung.

Datenintegrität und Datenauthentizität

Auch hier finden sich wieder die üblichen Verdächtigen, nämlich die gleichen Hash-Funktionen, die auch in IPSec Verwendung finden.

Schlüsselmanagement

Hier trifft man im Gegensatz zu IPSec, das nur einziges Verfahren zulässt, auf eine Vielfalt verschiedener Algorithmen in den verschiedensten Betriebsarten. Es werden sowohl Schlüsselerzeugungsverfahren als auch Schlüsseltauschverfahren, bei denen eine Seite einen Schlüssel erzeugt und diesen speziell verschlüsselt zur Gegenstelle sendet, angeboten. Sowohl RSA als auch Diffie-Hellman und der DSA (Digital Signature Algorithm) werden unterstützt.

Schutz vor Denial-of-Service-Angriffen

Hier ist bei SSL eine völlige Fehlanzeige. Aber nicht etwa weil SSL in diesem Bereich unsicher ist, sondern weil solche Angriffe auf Netzwerksysteme in der Regel auf IP, ICMP, TCP und seltener UDP zielen. Und SSL tritt erst dann in Aktion, wenn all diese Protokollschichten bereits durchlaufen sind. Die Mechanismen, die SSL zum Selbstschutz aufweist, arbeiten natürlich nur auf der SSL-Ebene und können auf Layer 3 oder 4 keinerlei Schutz bieten.

Den Schutz vor DoS-Angriffen muss man im Fall eines SSL-VPN also zusätzlich auf den entsprechenden Schichten implementieren.

Benutzerauthentifizierung

Auch in SSL findet nur eine Authentifizierung des SSL-Servers gegenüber dem Browser oder optional zusätzlich eine Authentifizierung des Browsers gegenüber dem Server statt. Benutzer gibt es in SSL genauso wenig wie in IPSec. Diese müssen sich also getrennt von SSL (aber unter dem Schutz von SSL) authentifizieren, oder die beiden Authentifizierungen müssen auf eine Art und Weise, die außerhalb des Definitionsbereichs von SSL liegt, miteinander kombiniert werden.

4.2 Die Grundlagen der Kryptografie

Die Kryptologie ist heutzutage ein Teilbereich der theoretischen Mathematik. Sie besteht aus der Kryptographie, welche die Vertraulichkeit von Nachrichten gewährleisten soll, und der Kryptoanalyse, der Kunst, eine verschlüsselte Nachricht wieder lesbar zu machen. Obwohl man oft den Begriff der Verschlüsselung mit Kryptografie gleichsetzt, kann man allerdings auch mit anderen Verfahren, z.B. der Verschleierung, den Inhalt von Nachrichten vor Unbefugten schützen.

4.2.1 Geschichtliches

Die Kryptografie war schon immer eng mit dem militärischen Bereich verknüpft. Hier entstand erstmals der Bedarf an öffentlicher Übertragung geheimer Daten. Bereits aus dem Altertum gibt es hierzu die ersten Überlieferungen, vorwiegend über Methoden zur Verschleierung sowie über einfache Substitutions- oder Transpositionsverfahren.

Im letzten Jahrhundert, dem Jahrhundert der Weltkriege, wurde die Kryptografie fast ganz vom Militär und von Regierungsbehörden vereinnahmt. Erst in den letzten 30 Jahren wurde sie wieder ein mehr und mehr öffentliches Thema, vor allem bedingt durch die Zunahme der Datenkommunikation über öffentliche Netzwerke (Internet, E-Commerce, Home-Banking usw.) und dem daraus resultierenden Wunsch nach Datenschutz und Datensicherheit.

Allerdings üben die Nachrichtendienste dieser Welt immer noch einen starken Einfluss aus, der sich auch vielfach in staatlichen Regulierungen niederschlägt. In einigen Ländern ist sogar die private Datenverschlüsselung bei Todesstrafe untersagt, zivilisiertere Länder begnügen sich mit Einfuhrverboten (Frankreich) von starker Verschlüsselung durch Privatpersonen oder Firmen oder beschränkten lange Zeit den freien Export (USA) von Kryptosystemen. Aber auch in Europa oder namentlich in Deutschland fallen Produkte mit starker Verschlüsselung unter das Kriegswaffenkontrollgesetz und sind einer gewissen Kontrolle und Regulierung unterworfen.

Insbesondere die US-amerikanische National Security Agency (NSA), die bis Mitte der 70er-Jahre ihre Existenz rundweg ableugnete, obwohl sie jährlich Milliardenbudgets verschlang und einige tausend der besten Mathematiker beschäftigte, wird gern als der böse Bube vom Dienst hingestellt. Tatsache ist, dass ihr Auftrag im weltweiten Abhören und Entschlüsseln von Informationsübermittlungen besteht, um amerikanische Interessen zu schützen. Die Dehnbarkeit dieses Begriffs demonstrierte der ehemalige CIA-Direktor James Woolsey im März 2000, als er in einem Interview in der Washington Post ganz offen zugab, dass bestimmte Informationen aus Abhörmaßnahmen der NSA an US-Firmen weitergegeben wurden – um amerikanische Firmen vor der National Culture of Bribery (nationale Bestechungskultur) der europäischen Konkurrenz zu schützen, wie er sich ausdrückte. Manch andere, vor allem die Opfer in Europa, werden so etwas allerdings richtig als Industriespionage bezeichnen. In der Tat hat das Ende des Kalten Krieges die Geheimdienstler dieser Welt nicht etwa arbeitslos gemacht. Deren neue Hauptaufgabe, zumindest in den Industrienationen, besteht mittlerweile in der Industriespionage.

Die Basis dieser Aktivitäten bildet das US-amerikanische ECHELON-System, das unter aktiver Mitwirkung von Großbritannien und Australien aktiv alles an Kommunikationsmedien abhört, was technisch möglich ist. Und das ist leider nicht wenig. Der Kern des Systems sind Rechnersysteme, welche die ausspionierten Informationen auf bestimmte, für die NSA interessant erscheinende Schlüsselbegriffe absuchen. Wenn jemand die Rechner der NSA zu umfangreichen Berechnungen veranlassen möchte, dann braucht er einfach nur eine mit dem E-Mail-Verschlüsselungsprogramm PGP verschlüsselte sinnlose E-Mail in die USA zu schicken und im Betreff-Feld die Begriffe „Airbus“ und „Boeing“ einzutragen. Es reicht aber auch vermutlich schon aus, diese beiden Wörter in einem Inlands-Telefongespräch fallen zu lassen – die NSA betreibt im bayrischen Bad Aibling bisher ungehindert und unkontrolliert eine große, offenbar gut ausgestattete Abhörstation!

Bei rechtem Licht besehen, ist die NSA aber nicht mehr oder weniger böse als andere Geheimdienste auch, sie ist aufgrund ihrer Ressourcen nur leistungsfähiger und damit als gefährlicher einzustufen. Das wirkliche Problem für jemanden, der sich vor Ausspähung durch die NSA schützen muss, ist die Frage, was die NSA wirklich zu leisten vermag, welche Ressourcen und welchen Wissens- und Technologievorsprung sie tatsächlich hat. Diese Frage ist bislang ungeklärt und gibt dadurch Anlass für wüste Spe-

kulationen und Verschwörungstheorien. Andererseits sollte man sich im Bereich der Daten- und Netzwerksicherheit ruhig eine Portion kontrollierten Verfolgungswahns gönnen, denn wer sagt „Mein Netzwerk ist sicher“, der hat schon verloren.

Im letzten Jahrhundert gab es in der Kryptografie dramatische Entwicklungen. Es wurden zum ersten Mal in großem Stil Maschinen zum Verschlüsseln und auf der Gegenseite zum Brechen der Codes eingesetzt. Dies war weniger eine Neuerung im Grundlagenbereich, sondern es wurden zum ersten Mal in der Geschichte durch Automaten Geheimtexte erzeugt, die von Menschen nicht mehr zu brechen waren. Dies gelang nur noch mit Automaten mit wesentlich höherer Leistungsfähigkeit als mit der der Chiffriermaschinen. Der Auslöser für diese Entwicklungen waren die neuen Kommunikationstechnologien, die Nachrichten per Kabel oder Funk übertrugen und dadurch nicht mehr abhörsicher waren.

Anfangs verbreiteten sich mechanische Systeme, gefolgt von elektromechanischen Rotormaschinen in den beiden Weltkriegen. Später, ab Ende der 70er-Jahre, waren es rein elektronische Systeme, und heute befassen sich ausschließlich Computer mit der Arbeit der Kryptografie und Kryptoanalyse.

Die elektromechanischen Rotormaschinen (z.B. die deutsche Enigma) produzierten bereits Chiffretexte, die von Menschen, selbst mit höchstem Personen- und Zeitaufwand, nicht mehr zu entschlüsseln waren. Zur Entschlüsselung wurden erstmalig in der Geschichte in England elektronische Rechenmaschinen, die so genannten „Bombs“, eingesetzt, die aus einer Vielzahl von Tyatronen, speziellen Schalterröhren, aufgebaut waren. Diese Vorläufer heutiger Computer waren schließlich in der Lage, die Chiffretexte der relativ einfach aufgebauten – und vielleicht gerade dadurch sehr sicheren – deutschen Enigma mit Known-Plaintext- und Chosen-Plaintext-Angriffen zu entschlüsseln.

Interessanterweise werden die Grundfunktionen der Enigma, die polyalphabetische Substitution und Permutation, auch heute noch in sehr vielen Neuentwicklungen von Verschlüsselungsverfahren eingesetzt.

4.2.2 Datenvertraulichkeit

Die Kryptografie wird benutzt, um einen bestimmten Grad an Vertraulichkeit von öffentlich zugänglichen Informationen zu erreichen. Die Information darf nur den dazu Berechtigten, im Falle der Datenkommunikation dem Sender und Empfänger, im Klartext zur Verfügung stehen. Während sie zwischen Sender und Empfänger transportiert wird, muss die Information in einer Weise verändert sein, die es Dritten unmöglich macht, Art und Inhalt der Information zurückzugewinnen.

4.2.3 Verschleierung und Verschlüsselung

Als Methoden bieten sich grundsätzlich zwei Verfahren an, die sich aufgrund ihres möglichen Einsatzbereiches unterscheiden: die Verschleierung und die Verschlüsselung von Informationen. Die Verschleierung wird nur noch selten eingesetzt, da ihre kryptografische Stärke nicht sehr hoch und sie in größeren Umgebungen mit sehr vielen Beteiligten nicht zu handhaben oder sehr unsicher ist. Im militärischen Bereich wird sie gelegentlich noch benutzt, um taktische Informationen begrenzter Lebensdauer und mit

eingeschränktem Gültigkeitsbereich zu übertragen. Typischerweise werden hier Begriffe durch andere Begriffe ersetzt, um die Bedeutung von Informationen, die zum Beispiel durch Sprechfunkverkehr übertragen werden, zu verschleiern.

Ein Beispiel aus der Datenkommunikation ist das Übertragen von vertraulichen Informationen innerhalb anderer, auf den ersten Blick unverdächtiger Informationen. Zum Beispiel kann man die niederwertigen Bits einer Bilddatei benutzen, um darin Informationen verschleiert zu übertragen. Jemand, der die Übertragung mitschneidet oder sich die Datei verschafft, bekommt nur ein Bild angezeigt, und selbst der augenscheinliche Vergleich mit dem Originalbild zeigt im Allgemeinen keinen Unterschied. Jedoch würde ein strenger Vergleich der Bilddateien die Unterschiede an den Tag bringen.

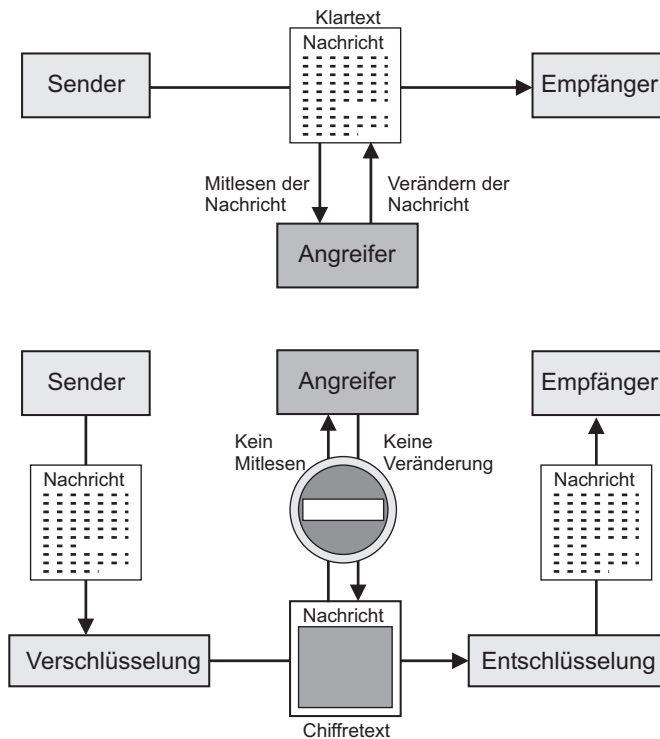


Abbildung 4.1: Durch die Verschlüsselung wird die Nachricht für Dritte unlesbar gemacht.

Die Verschlüsselung ist meist universeller angelegt, in der Regel sind die Verschlüsselungsverfahren selbst sogar offen gelegt. Die Sicherheit wird durch eine Zusatzinformation, den so genannten Schlüssel, erreicht, der neben dem Klartext als zweite Eingangsgröße den Chiffretext erzeugt. Mit Hilfe dieses Schlüssels können Sender und Empfänger die zwischen ihnen ausgetauschten Informationen ver- und entschlüsseln. Die Verschlüsselungsverfahren, die heute in der Datenkommunikation Anwendung finden, sind sehr universell angelegt und arbeiten nicht mehr wie die Verschleierung auf der Ebene von Sprachbegriffen, sondern auf der Ebene von abstrakten Informationseinheiten wie elektrischen Spannungspegeln, Bits, Bytes usw.

Die Kryptoanalyse ist die Kunst, eine verschlüsselte Nachricht wieder lesbar zu machen. Dies ist leichter gesagt als getan. Denn sie ist in der Tat eine Kunst oder genau genommen ein Beweis menschlicher Erfindungsgabe und Fantasie. Es gibt eine ganze Reihe verschiedener Angriffsarten, die sich darin unterscheiden, welche Informationen und Möglichkeiten dem Kryptoanalytiker zur Verfügung stehen.

Unter der Kryptoanalyse versteht man im Allgemeinen nicht das Erzeugen des Klartextes aus dem Chiffretext, sondern das Auffinden des zur Verschlüsselung verwendeten Schlüssels. Alle heutigen Chiffrierverfahren benutzen praktisch allgemein bekannte, offen gelegte Algorithmen, deren Sicherheit darauf basiert, dass der Schlüssel zum Ver- und Entschlüsseln geheim gehalten wird. Die Sicherheit dieser Verfahren basiert also ausschließlich auf dem Schlüssel, der damit auch zum Hauptangriffsziel der Kryptoanalytiker avanciert.

Ciphertext-only-Angriff

Dies ist die größte Herausforderung an einen Kryptoanalytiker. Es steht ihm nur der Chiffretext zur Verfügung, und er kennt weder die Art noch den Inhalt auch nur von Teilen des Klartexts. Das Hauptproblem dieses Angriffs ist die Tatsache, dass der Angreifer gar nicht weiß, wann er einen Teil des Chiffretexts entschlüsselt hat, da er ja keinerlei Kenntnis vom Klartext hat.

Known-Plaintext-Angriff (Brute-Force-Attack)

Besser hat es ein Kryptoanalytiker schon, wenn er zum Chiffretext wenigstens einen Teil des zugehörigen Klartextes kennt, also auch weiß, wann er den richtigen Schlüssel gefunden hat. Der Brute-Force-Angriff, also ein Angriff mit Brachialgewalt, ist der typische Vertreter dieser Angriffsart. Man probiert am Chiffretext mit dem bekannten Dechiffrieralgorithmus so lange alle in Frage kommenden Schlüsselvarianten aus, bis man den richtigen gefunden hat. Dies ist der Fall, wenn das Ergebnis der Entschlüsselung gleich dem bekannten Klartext ist.

Der folgende Pseudocode demonstriert einen Brute-Force-Angriff auf eine DES-Verschlüsselung. Die Funktion bekommt als Eingangsparameter den Chiffretext und den Klartext und liefert als Ergebnis den Schlüssel zurück. Qword sind 64 Bit große Variablen. Die externe Funktion DES entschlüsselt den Ciphertext mit dem Key und liefert als Ergebnis einen 64-Bit-String zurück:

```
Function Qword Brute_Force_Attack(Qword Cleartext, Qword Ciphertext)
Extern Qword DES_Decrypt(Qword Key, Qword Ciphertext);
Qword key;
For key = 0 to 72057594037927936 do
    IF Cleartext == DES_Decrypt(key, Ciphertext) then Return(key)
done
Return(0)
```

Wie man leicht erkennen kann, ist das Hauptproblem dieses Verfahrens die Zeit, die für die Suche aufgewendet werden muss. Nehmen wir einmal ein Beispiel aus der Praxis. Der Data Encryption Standard benutzt einen Schlüssel, der 56 Bit lang ist, also insgesamt 72.057.594.037.927.936 verschiedene Werte annehmen kann. Angenommen, ein sehr

schneller Rechner bräuchte für jede Dechiffrierung eine Mikrosekunde und der Schlüssel würde statistisch gesehen nach 50% aller Versuche gefunden, dann würde solch ein Angriff immer noch über 1.142 Jahre dauern und wäre praktisch sinnlos.

Der Klartext zu einem gegebenen Chiffretext ist sehr oft bekannt, da darf man sich keine Illusionen machen. Bestimmte Datei-Header, Floskeln im Text, Mail-Signaturen usw. sind immer gleich und dadurch auch leicht zu erraten. Es werden auch sehr oft Blockverschlüsselungsverfahren eingesetzt, die den letzten Block nach einem allgemein bekannten Verfahren mit Füllzeichen (Padding) auffüllen.

Das Verhältnis der Schnelligkeit eines Brute-Force-Angriffs zum vorhandenen Budget hierfür ist innerhalb bestimmter technischer Grenzen annähernd linear. In realen Systemen zur Kryptoanalyse von DES arbeiten viele DES-Prozessoren parallel mit eigenen Teilbereichen des zu durchsuchenden Schlüsselraumes. Entweder verwendet man Parallelrechner oder spezielle Systeme mit insgesamt Hunderten oder Tausenden von parallel arbeitenden DES-Chips, die einen 56 Bit langen Schlüssel innerhalb von Stunden oder gar Minuten knacken können. Der NSA unterstellt man, mittlerweile weniger als eine Sekunde zu benötigen, um eine 56-Bit-DES-Verschlüsselung zu brechen.

Schutz gegen einen Brute-Force-Angriff erzielt man ganz einfach durch Verlängern des Schlüssels, denn jedes Bit mehr Schlüssellänge verdoppelt die Suchzeit.

Der Brute-Force-Angriff ist aber letztendlich auch die Messlatte für die Sicherheit eines Verschlüsselungsalgorithmus. Ein Verschlüsselungsverfahren gilt definitionsgemäß dann als sicher, wenn der Brute-Force-Angriff die schnellste bekannte Methode ist, den Schlüssel zu ermitteln. Damit kann man auch seine benötigte Schlüssellänge kalkulieren, indem man den Aufwand eines möglichen Angriffs abschätzt und als Resultat seine Schlüssel, natürlich mit einer kräftigen Sicherheitsreserve versehen, hinreichend lang macht.

So ist denn auch die Hauptentwicklungsrichtung der meisten Kryptoanalytiker nicht das Optimieren von Brute-Force-Angriffen, sondern die Suche nach Verfahren, die schneller als diese sind.

Beim Chosen-Plaintext-Angriff kann der Kryptoanalytiker selbst Klartext in die Verschlüsselung einschleusen, und beim Adaptive-Chosen-Plaintext-Angriff kann er dies sogar in einer Schleife mit immer neuem Klartext tun, abhängig vom Ergebnis der Chiffrierung. Diese beiden Angriffe sind, vor allem im letzteren Fall, eher eine Angelegenheit fürs Labor. Genau dort werden diese Verfahren auch eingesetzt, und zwar um Verschlüsselungsverfahren auf ihre Stärke zu testen.

Ein in letzter Zeit populär gewordener Vertreter dieser Angriffsart ist die differenzielle Kryptoanalyse, mit der man zum Beispiel DES, zumindest theoretisch, analysieren konnte. Das Ziel dieses Verfahrens ist festzustellen, wie sich der Chiffretext bei gezielten Änderungen des Klartextes ändert, um so auf den Schlüssel zu schließen.

Dies sind bisher jedoch alles theoretische Angriffe, die obendrein recht futuristisch anmutende Computer mit 1000 Terabyte RAM und noch viel größere Massenspeicher voraussetzen. Außerdem ist die Komplexität einer differenziellen Kryptoanalyse immer noch weitaus größer als die eines Brute-Force-Angriffs. Die weiteren Rahmenbedingun-

gen zu diesem Verfahren sind darüber hinaus völlig unrealistisch, es müssen zum Beispiel zu einem bestimmten Schlüssel 1012 GByte Chiffretext und der dazugehörige Klartext bekannt sein, um ihn zu berechnen. Dies kommt in der Praxis äußerst selten vor, und es lässt sich mit modernen Sicherheitsprotokollen, wie zum Beispiel IPSec und IKE, auch wirksam verhindern, dass mit einem einzigen Schlüssel überhaupt so viele Daten verschlüsselt werden.

Grundbausteine der Kryptografie

Das Grundgerüst der Kryptografie besteht aus einer möglichst starken Kombination von Diffusion und Konfusion. Die Diffusion verschleiert gewissermaßen den Zusammenhang zwischen den Informationselementen im Klartext und Chiffretext. Die Konfusion verteilt die Informationselemente des Klartextes im Chiffretext. Oder sie verstärkt – hier haben wir dann unsere Kombination beider Prinzipien – die Diffusion dadurch, dass die Informationselemente im Chiffretext umverteilt werden. Das klingt noch etwas abstrakt, ist aber, wie Sie anschließend sehen werden, mit verblüffend einfachen Verfahren zu realisieren. Die Diffusion wird in der Praxis durch Substitution und die Konfusion durch Permutation oder Transposition realisiert.

Substitution

Bei der Substitutionschiffrierung wird jedes Zeichen im Klartext durch ein anderes Zeichen im Chiffretext ersetzt. Die Zeichen des Klartexts tauchen nicht mehr im Chiffretext auf. Die Zuordnungsvorschrift der Zeichen, also der Schlüssel, ist nur den Berechtigten, dem Sender und dem Empfänger, bekannt. Man unterscheidet unterschiedliche Substitutionsmethoden.

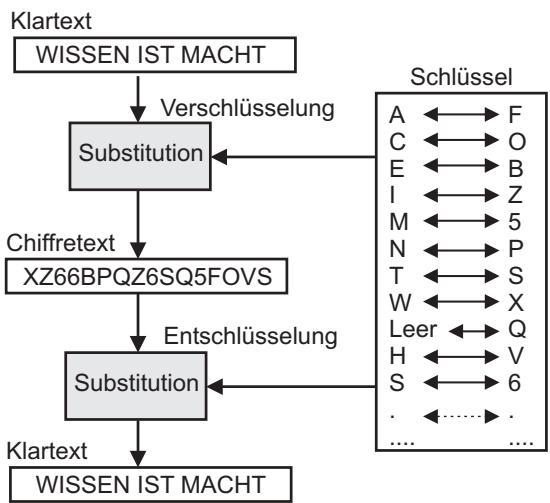


Abbildung 4.2: Eine monoalphabetische Substitutionschiffrierung

Bei der monoalphabetischen Substitution wird jedes mögliche Zeichen des Klartexts durch ein einziges, fest zugeordnetes Zeichen ersetzt. Abbildung 4.2 zeigt ein Beispiel für eine monoalphabetische Substitution. Der ASCII-Text „WISSEN IST MACHT“ wird entsprechend der angegebenen Substitutionstabelle – sie stellt den Schlüssel dieses Verfahrens dar – in den Chiffretext „XZ66BPQZ6SQ5FOVS“ transformiert.

Bei der polyalphabetischen Substitution kann jedes mögliche Zeichen des Klartexts durch verschiedene Zeichen im Chiffretext ersetzt werden, zum Beispiel kann A zu X, R oder Z werden, je nachdem, an welcher Stelle es im Klartext vorkommt. Dieses Verfahren fand auch in der bekannten deutschen Schlüsselmaschine Enigma erfolgreich Anwendung.

Die Nachteile der monoalphabetischen Substitution lassen sich aus dem Chiffretext in Abbildung 4.2 ablesen. Aufgrund der festen Zuordnung zueinander tauchen sowohl im Klar- als auch im Chiffretext bestimmte Zeichen mehrfach auf. Beispielsweise erscheint die Ziffer 6 dreimal in der kurzen Zeile, dabei zweimal hintereinander; der Buchstabe Q kommt zweimal vor. Ein geübter Angreifer erkennt daraus sofort Folgendes: Es handelt sich hier um die Verschlüsselung eines ASCII-Textes, die Doppelzeichen müssen auch Doppelbuchstaben im Klartext sein, und die häufig auftauchenden Zeichen sind entweder Leerzeichen im Klartext, oder es sind Buchstaben, die in einer bestimmten Sprache oft vorkommen.

Dadurch, dass sich durch die monoalphabetische Substitution die Häufigkeit und die Position der im Klartext vorkommenden Zeichen im Chiffretext nicht verändert, lässt sich eine solche Chiffrierung leicht knacken, sogar durch einen Ciphertext-only-Angriff. Diesen Nachteil umgeht die polyalphabetische Substitution, indem dort ein Eingangswert im Klartext mehrere verschiedene Ausgangswerte im Chiffretext zur Folge haben kann.

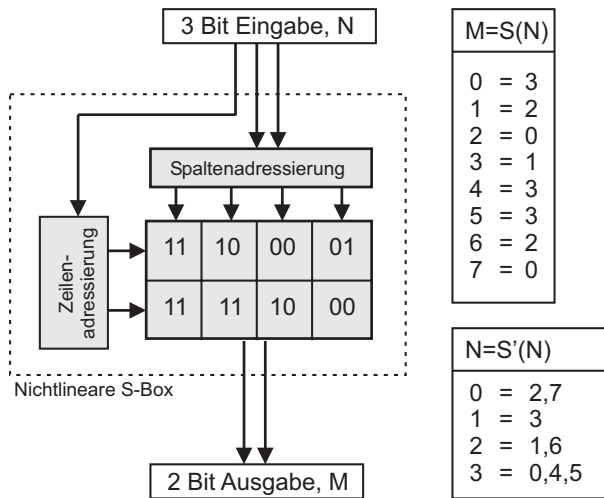


Abbildung 4.3: Eine nichtlineare S-Box substituiert unumkehrbar binäre Informationsgrößen.

In modernen Hard- und Software-Implementierungen erfolgt die Substitution mit Hilfe so genannter S-Boxen (Substitution Box). Eine S-Box liefert zu einem Eingangswert N einen Ausgangswert M. Der Wertebereich von N kann kleiner oder gleich dem von M sein, im ersten Fall handelt es sich um eine lineare, im zweiten um eine nichtlineare S-Box.

Kryptografisch interessant ist die nichtlineare Substitution, weil hier verschiedene Zeichen des Klartexts auf ein Zeichen Chiffretext abgebildet werden, so dass sich nicht mehr zurückverfolgen lässt, welcher Eingabewert für einen bestimmten Ausgabewert verantwortlich war. Damit hat man bereits eine, wenngleich noch schwache, Einwegfunktionalität realisiert.

Die nichtlineare S-Box in Abbildung 4.3 transformiert den 3-Bit-Eingabewert N in den 2-Bit-Ausgabewert M. Der Ausgabewert 0 beispielsweise kann zwei mögliche, der Ausgabewert 3 bereits drei mögliche Eingangswerte gehabt haben. In der Praxis werden mehrere verschiedene S-Boxen so lange kaskadiert, bis eine Rückverfolgung des Eingangswertes aus dem Ausgangswert nicht mehr möglich ist. Damit erzielt man eine Einwegfunktionalität und kann den Ausgangswert nicht mehr zurückberechnen, auch nicht wenn man die internen Tabellen der S-Box(en) kennt. Es ist offensichtlich, dass man solch ein Konstrukt zwar zum Verschlüsseln, aber nicht mehr zum Entschlüsseln benutzen kann. In der Praxis dienen solche Funktionen, also nichtlineare S-Boxen in Verbindung mit Permutationen, auch nicht zum eigentlichen Verschlüsseln, sondern zur Transformation von Schlüsseln.

Permutation (Transposition)

Bei der Transpositionschiffrierung werden alle Zeichen des Klartextes auf andere Positionen im Chiffretext umgesetzt. Die Zeichen des Originaltextes bleiben im Chiffretext erhalten. Die Art der Transposition, d. h. der Schlüssel, ist nur den Berechtigten, also dem Sender und dem Empfänger, bekannt. Je geringer der Informationsgehalt der Zeichen ist, die permutiert werden sollen, desto stärker ist dieses Verfahren. Eine Permutation von Wörtern einer Sprache ist leicht zu analysieren, die von reinen Buchstaben schon weniger leicht, aber Permutationen auf Bit-Ebene sind nur sehr schwer zurückzuverfolgen.

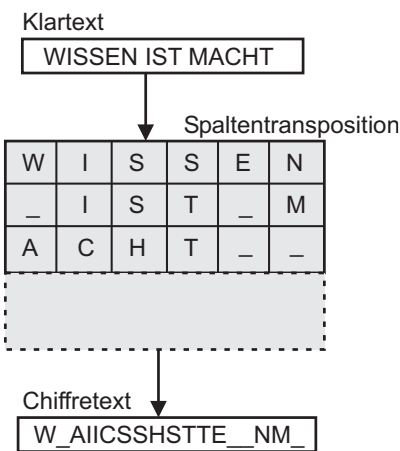


Abbildung 4.4: Die Spaltentransposition vertauscht die ursprünglichen Positionen der Zeichen.

Abbildung 4.4 zeigt ein Beispiel für eine Transpositionschiffrierung, die so genannte Spaltentransposition. Hierbei wird der Klartext „WISSEN IST MACHT“ zeilenweise so lange in eine Matrix eingetragen, bis der Text zu Ende ist. Der Chiffretext wird durch ein anschließendes spaltenweises Auslesen der Tabelle erzeugt.

Zur Transposition werden in modernen, praktischen Implementierungen so genannte P-Boxen (Permutation Box) eingesetzt. Der Ausgangswert einer P-Box entsteht aus einer bitweisen Vertauschung der Bits des Eingangswerts. P-Boxen können wie S-Boxen in Hard- und Software realisiert werden. Bei Hardware-Implementierungen kann die Permutation durch eine Festverdrahtung mit extrem hohen Geschwindigkeiten realisiert werden.

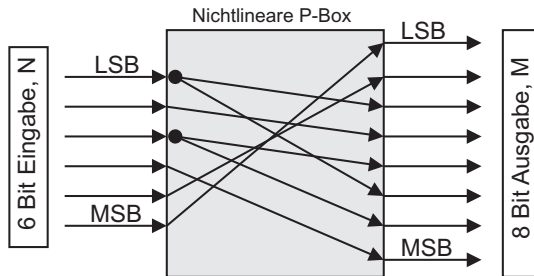


Abbildung 4.5: Die Expansionspermutation erzeugt einen Lawineneffekt.

Wie bei der Substitution gibt es hier ebenfalls lineare und nichtlineare Varianten der P-Boxen. Lineare P-Boxen mit einer festen Zuordnung sind kryptografisch wenig interessant. In der Praxis kommen in der Regel nichtlineare P-Boxen vor, die eine so genannte Kompressions- oder Expansionspermutation durchführen. Bei der Expansionspermutation ist die Anzahl der Bits des Ausgangswerts größer als die des Eingangswerts. Bei der Kompressionspermutation ist die Anzahl der Bits des Ausgangswerts kleiner als die des Eingangswerts, hierbei gehen Informationen verloren.

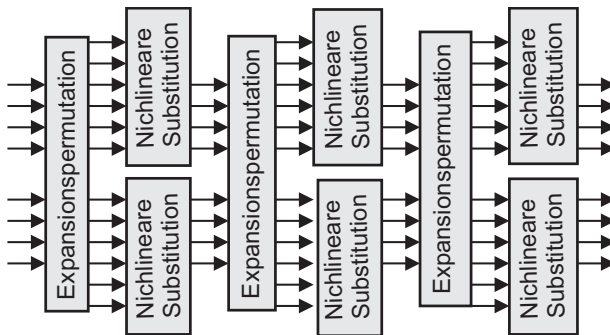


Abbildung 4.6: Die Produktchiffre verkettet mehrfach Substitution und Permutation.

Abbildung 4.5 zeigt ein Beispiel einer nichtlinearen P-Box. Es handelt sich um eine Expansionspermutation mit einem 6-Bit-Eingangswert N und einem 8-Bit-Ausgangswert M . Um den 8-Bit-Ausgangswert zu erhalten, werden jeweils zwei Bit des Eingangswerts verdoppelt. Die Änderung eines einzigen Bits dieser beiden hat eine Änderung von gleich zwei höherwertigen Bits des Ausgangswerts zur Folge. Das heißt, durch minimale Änderungen des Eingangswerts lassen sich starke Änderungen des Ausgangswerts erzielen. Man spricht hierbei von einem so genannten Lawineneffekt.

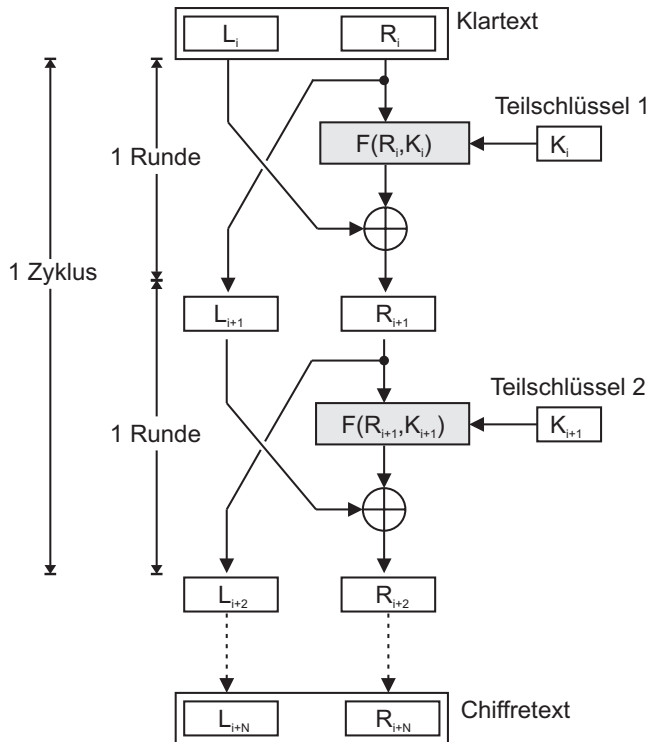


Abbildung 4.7: Ein aus zwei Runden eines Feistel-Netzwerks bestehender Feistel-Zyklus

Um eine Dechiffrierung zu erschweren, werden in der Praxis Substitutions- und Permutationsverfahren miteinander verknüpft und hinreichend oft kaskadiert, um an einem bestimmten Punkt einen Zustand zu haben, an dem von dem Ausgangswert nicht mehr auf den Eingangswert geschlossen werden kann. Eine solche Produktchiffre sehen Sie in Hier werden nichtlineare Substitutionen und Expansionspermutationen so lange verketet, bis ein Ausgangswert entstanden ist, aus dem der Ursprungswert nicht mehr zurückberechnet werden kann.

Feistel-Netzwerk

Feistel-Netzwerke sind der Grundbaustein fast aller heute gebräuchlichen symmetrischen Blockverschlüsselungsverfahren. Der Name stammt von dem damaligen IBM-Mitarbeiter Horst Feistel, der sich Anfang der 70er-Jahre im Auftrag seines Arbeitgebers mit der Entwicklung von Chiffrierverfahren beschäftigte. Aus seinem Team stammt auch der Lucifer-Chiffre, ein Algorithmus, der später nach einigen Modifikationen zum Data Encryption Standard (DES) wurde und auch heute noch weltweit in Gebrauch ist. Auf den ersten Blick sieht ein Feistel-Netzwerk oder ein Feistel-Zyklus, der aus zwei so genannten Runden eines Feistel-Netzwerks besteht, gar nicht so aufregend aus. Aber es besticht durch eine Eigenschaft, die es zur Basisstruktur moderner Block-Chiffrierverfahren gemacht hat. Es erlaubt nämlich, die unumkehrbaren Produktchiffren, die aus nichtlinearen Substitutionen und Permutationen bestehen, sowohl zur Ver- als auch zur Entschlüsselung zu benutzen!

In Abbildung 4.7 ist ein Feistel-Zyklus dargestellt, der aus zwei Runden eines Feistel-Netzwerks besteht. Die Funktion F ist eine Produktchiffre, die auf unumkehrbare Weise aus ihren Eingangswerten – dem Teilschlüssel und einer Hälfte des Klartextes – ihren Ausgangswert erzeugt. Die eigentliche Transformation der anderen Hälfte des Klartextes in den Chiffretext erfolgt jedoch – und das ist der springende Punkt in diesem Verfahren – durch eine einfache Exklusiv-oder-Verknüpfung. Und diese Funktion, Exklusiv-oder, ist umkehrbar, denn man kann den Chiffretext mit dem gleichen Schlüssel, mit dem er erzeugt wurde, wieder in den ursprünglichen Klartext umwandeln. Auf unser Beispiel bezogen folgt somit, dass bei Kenntnis der Funktion F und des Schlüssels, aus dem die Teilschlüssel abgeleitet werden, eine Dechiffrierung dadurch möglich ist, dass man den Chiffretext in den Feistel-Zyklus einspeist und lediglich die Reihenfolge der zu benutzenden Teilschlüssel umkehrt. Das Ergebnis ist wieder der Klartext. In unserem Beispiel würde nach einem Zyklus aus dem Klartext (L_i, R_i) der Chiffretext (L_{i+2}, R_{i+2}) durch folgende Transformationen:

Erste Runde (die Exklusiv-oder-Verknüpfung ist im Folgenden mit $\oplus$ notiert):

$$L_{i+1} = R_i$$

$$R_{i+1} = L_i \oplus F(R_i, K_i)$$

Zweite Runde:

$$L_{i+2} = R_{i+1}$$

$$R_{i+2} = L_{i+1} \oplus F(R_{i+1}, K_{i+1})$$

Durch die Umkehrung, also wenn dieser Zyklus mit dem Chiffretext (L_{i+2}, R_{i+2}) und dem Teilschlüssel K_{i+1} beginnt, erhalten wir wieder den Klartext (L_i, R_i) :

Erste Runde:

$$L_{i+1} = R_{i+2}$$

$$R_{i+1} = L_{i+2} \oplus F(R_{i+2}, K_{i+1})$$

Zweite Runde:

$$L_i = R_{i+1}$$

$$R_i = L_{i+1} \oplus F(R_{i+1}, K_i)$$

Des Weiteren folgt daraus auch, dass es – eine entsprechend unumkehrbare Funktion F vorausgesetzt – nicht mehr möglich ist, auch bei Kenntnis von zusammengehörendem Klar- und Chiffretext, einen oder mehrere Teilschlüssel zu rekonstruieren. Die einzig verbliebene Angriffsart ist somit ein Brute-Force-Angriff.

Praktische Implementierungen begnügen sich nicht mit nur einem Feistel-Zyklus, sondern verketteten ihn zu mehreren Runden. DES benutzt zum Beispiel acht Zyklen, also insgesamt 16 Runden. Die zu verwendende Anzahl der Runden hängt maßgeblich von der Qualität der Funktion F ab, die als einzige über die Sicherheit des Chiffrierverfahrens entscheidet.

4.2.6 Verschlüsselungsverfahren

Alle in der Praxis eingesetzten Verschlüsselungsverfahren benutzen einen allgemein bekannten Algorithmus, der den Chiffretext aus dem Klartext und aus einem Schlüssel erzeugt. Die Sicherheit dieser Verfahren basiert damit ausschließlich auf der Geheimhaltung und der Qualität des Schlüssels, nicht auf der Vertraulichkeit des Algorithmus selbst. Aus diesem Grunde sind auch alle ernst zu nehmenden Algorithmen offen gelegt. Dies bedeutet, dass jeder diese Algorithmen analysieren und auf Schwachstellen testen kann und somit unsichere Kandidaten sehr schnell entdeckt werden. So gibt es vermutlich kein Verfahren, an dessen Kryptoanalyse sich mehr Personen und Organisationen versucht haben als an DES und Triple-DES.

Die Tatsache, dass es noch keiner „geknackt“ hat, also noch niemand ein Verfahren angewendet hat, das schneller als ein Brute-Force-Angriff ist, spricht letztendlich für dessen Sicherheit. Die Tatsache, dass der normale DES mit seinem 56 Bit langen Schlüssel nicht mehr als sicher gilt, liegt denn auch nicht daran, dass jemand ein Verfahren gefunden hat, das schneller als ein Brute-Force-Angriff auf DES ist, sondern dass dieser aufgrund der Leistungsfähigkeit heutiger Rechner tatsächlich durchführbar geworden ist.

Es gibt zwei grundsätzlich verschiedene Verfahren, die auf Schlüsseln basieren: die symmetrische oder Secret-Key-Verschlüsselung und die asymmetrische oder Public-Key-Verschlüsselung.

Symmetrische Verschlüsselung

Bei der symmetrischen Verschlüsselung kennen alle beteiligten Gegenstellen einen geheimen Schlüssel, den so genannten Secret Key, der sowohl zum Ver- als auch zum Entschlüsseln benutzt wird. Symmetrische Verfahren werden in Form von Datenstrom- oder Datenblockverschlüsselungen verwendet.

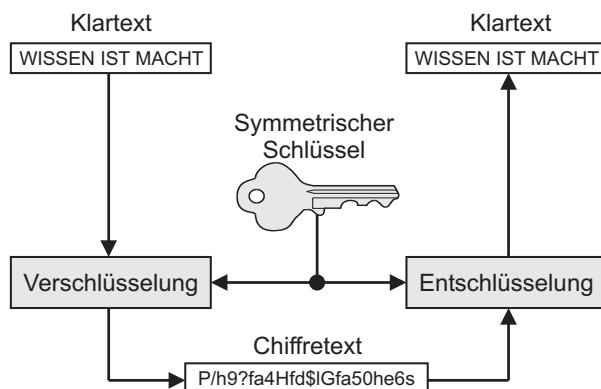


Abbildung 4.8: Die symmetrische Verschlüsselung benutzt gleiche Schlüssel zur Ver- und Entschlüsselung.

Das Grundprinzip funktioniert wie in Abbildung 4.8 dargestellt. Zunächst vereinbaren Sender und Empfänger einen gemeinsamen Schlüssel. Dieser dient dem Sender zur Verschlüsselung des Klartexts und dem Empfänger zur Entschlüsselung des Chiffretexts.

Ältere Algorithmen wurden in der Regel zur Implementierung in Hardware entwickelt, neuere Verfahren wurden universeller ausgelegt und erreichen auch als Software-Implementierung eine hohe Performance. So gibt es heute bereits DES-Chips mit mehreren GBit/s Durchsatz. Der RC4-Algorithmus hingegen ist ein typischer Vertreter von Verfahren, die für Software-Verschlüsselung, im Fall von RC4 speziell für die Intel-Prozessor-Architektur, entwickelt wurden.

Da die Sicherheit dieser Verfahren ausschließlich von der Länge und der Vertraulichkeit des Schlüssels abhängt, sind sie leicht an verschiedene Sicherheitsanforderungen anzupassen. Um sich vor einem Brute-Force-Angriff zu schützen, gibt es verschiedene Empfehlungen, wie lang der Schlüssel sein sollte, und zwar abhängig davon, von wem ein Angriff erwartet wird und wie lange Informationen geheim gehalten werden müssen. Allerdings sind solche Empfehlungen mit einer gewissen Vorsicht zu genießen, denn wer kann heute schon sagen, welche Technologie in 20 Jahren zur Verfügung stehen wird. Zu diesem Zeitpunkt sind Daten, die heute mit einem jetzt noch sicheren Verfahren verschlüsselt werden, vielleicht innerhalb von Stunden oder Minuten zu dechiffrieren, wie das Beispiel DES nur allzu deutlich vor Augen geführt hat.

Schlüsselart	Länge	Mögliche Kombinationen	Testzeit	Parallele Tests	Mittlere Suchzeit
Kurzes Passwort	28 Bit	8.1450.625	1 s	1	40 Sekunden
RC4-Schlüssel	40 Bit	1.099.511.627.776	1 s	1	6 Tage
RC4-Schlüssel	40 Bit	1.099.511.627.776	1 s	50	2,8 Stunden
RC4-Schlüssel	40 Bit	1.099.511.627.776	1 s	1.000.000	0,05 Sekunden
DES-Schlüssel	56 Bit	72.057.594.037.927.900	1 s	1	1.142 Jahre
DES-Schlüssel	56 Bit	72.057.594.037.927.900	1 s	1.000.000	10 Stunden
IDEA-Schlüssel	128 Bit	$3,4 \cdot 10^{38}$	1 s	1	$5,4 \cdot 10^{24}$ Jahre
IDEA-Schlüssel	128 Bit	$3,4 \cdot 10^{38}$	1 s	1.000.000	$5,4 \cdot 10^{18}$ Jahre

Tabelle 4.1: Brute-Force-Angriffe auf verschieden lange Schlüssel

In Tabelle 4.1 sehen Sie einen Vergleich zwischen Brute-Force-Angriffen auf Schlüssel verschiedener Länge. Als Zeitbedarf wurde eine Mikrosekunde pro Test angesetzt. Für das RC4-Verfahren mit 40 Bit wirksamer Schlüssellänge, wie es in praktisch allen Webbrowsern unter der (alten) US-Exportlizenz außerhalb der USA eingesetzt wurde und mit dem somit unter Umständen auch Ihre Kreditkarteninformationen verschlüsselt wurden, sehen Sie in der Zeile 4, dass der Zeitaufwand für einen Angriff mit 50 parallelen Systemen nur 2,8 Stunden beträgt. Das DES-Verfahren mit seiner Schlüssellänge von 56 Bit beschäftigt eine Workstation schon über 1.000 Jahre – falls der Rechner überhaupt so lange funktioniert und das Resultat dann noch jemanden interessiert. Ein Brute-Force-Angriff auf ein starkes Verschlüsselungsverfahren mit 128-Bit-Schlüssel würde theoretisch $5,4 \times 10^{24}$ Jahre dauern.

Man kann den Aufwand zum Suchen eines Schlüssels vergrößern, indem man sehr viele, schnelle Hardware-Systeme parallel einsetzt und jedes nur einen Teilbereich des möglichen Schlüsselraums durchsuchen lässt. Mit Investitionen im siebenstelligen Dollarbereich sind Systeme denkbar – wer weiß, vielleicht gibt es sie auch schon –, die 1.000.000 Suchvorgänge pro Minute schaffen können. Der 56-Bit-DES-Schlüssel wäre mit einem solchen System in zehn Stunden gefunden; der IDEA-Schlüssel wäre mit einer Suchzeit von $5,4 \times 10^{18}$ Jahren immer noch sicher. Ein 40-Bit-Schlüssel wäre damit jedoch bereits im Schnitt nach 50, garantiert nach 100 Millisekunden gefunden!

Anhand dieser Tabelle können Sie erkennen, dass Verschlüsselungsverfahren mit einer Schlüssellänge von 128 Bit oder größer vor einem Brute-Force-Angriff sicher sind. Selbst wenn sich in den nächsten zehn Jahren die Geschwindigkeit der Prozessoren um den Faktor 1.000.000.000 steigern sollte, betrüge die Suchzeit immer noch $5,4 \times 10^9$ Jahre mit einem solchen Supercomputer!

Im Gegensatz zur symmetrischen Verschlüsselung verwenden die asymmetrischen Verfahren zwei verschiedene Schlüssel. Auf der einen Seite ist dies der öffentliche Schlüssel oder Public Key, den jeder kennt und der zum Verschlüsseln (Chiffrierschlüssel) einer Nachricht verwendet wird. Auf der anderen Seite gibt es einen geheimen Schlüssel oder Private Key, den nur der Empfänger der Nachricht kennt und der zum Entschlüsseln (Dechiffrierschlüssel) dient. Eine mit einem öffentlichen Schlüssel verschlüsselte Nachricht kann mit demselben oder einem anderen öffentlichen Schlüssel nicht mehr entschlüsselt werden.

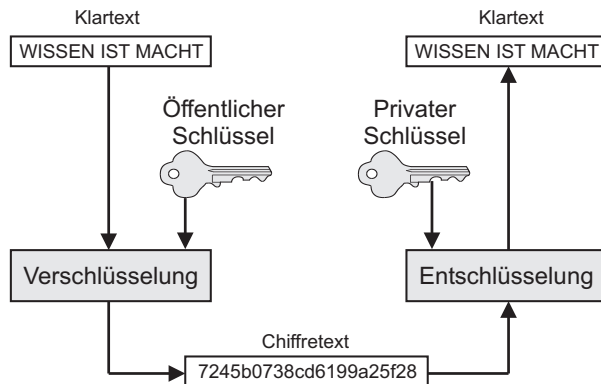


Abbildung 4.9: Die asymmetrische Verschlüsselung benutzt unterschiedliche Schlüssel zur Ver- und Entschlüsselung.

Private Key und Public Key sind einander fest zugeordnete „Schlüsselpaare“, die aus einem mathematischen Verfahren, das nicht zurückzuverfolgen ist, voneinander abgeleitet werden. Die bekanntesten Public-Key-Verfahren sind RSA und Diffie-Hellman, die später in diesem Kapitel noch ausführlich besprochen werden.

Abbildung 4.9 zeigt den Ablauf einer asymmetrischen Verschlüsselung. Der Klartext wird mit dem öffentlichen Schlüssel verschlüsselt. Auf der Gegenseite wird der Chiffre-

text mit dem privaten Schlüssel entschlüsselt. Der öffentliche Schlüssel ist jedermann zugänglich. Er kann zum Beispiel auf einem öffentlichen Server abgelegt sein, da jede Person, die einer anderen Person nach diesem Prinzip eine Nachricht zusenden will, deren öffentlichen Schlüssel zum Verschlüsseln benötigt. Der Empfänger benutzt seinen privaten Schlüssel, um die Nachricht wieder zu entschlüsseln. Nur er allein kann die Nachricht entschlüsseln, weil nur er seinen privaten Schlüssel besitzt.

Es gibt asymmetrische Verfahren, zum Beispiel RSA, die umgekehrt auch mit dem privaten Schlüssel verschlüsseln und mit dem öffentlichen Schlüssel wieder entschlüsseln können. Diese Betriebsart spielt zur Verschlüsselung von Nachrichten natürlich keine Rolle, sondern wird für Anwendungen zur Authentifizierung durch digitale Signaturen eingesetzt.

Auch bei asymmetrischen Algorithmen muss man sich vor Angriffen schützen. Asymmetrische Verfahren arbeiten aus technischen Gründen allerdings mit weit größeren Schlüssellängen als symmetrische. Ein heute sicherer asymmetrischer RSA-Schlüssel beginnt nach den allerneuesten Erkenntnissen der Kryptologie (Stand Juli 2005) bei einer Länge von 2048 Bit. Deshalb gelten hier andere Werte zum Schutz vor Angriffen. Nur um solche Angaben in die richtige Perspektive zu rücken: Zum Zeitpunkt der Entstehung der Erstauflage dieses Buches (1999) waren sich die meisten Wissenschaftler einig, dass ein 1024 Bit Schlüssel eine ausreichende Sicherheit böte.

Asymmetrische Verfahren sind wesentlich langsamer als symmetrische. Deshalb werden sie selten zur Verschlüsselung von Nutzdaten, wie Datenpaketen und Dateien, eingesetzt. In der Praxis dienen sie dazu, ein symmetrisches Schlüsselpaar für den Sender und Empfänger zu erzeugen oder zu übertragen. Dieser Vorgang dauert mit einem 133-MHz-Pentium-PC kaum länger als zwei bis drei Sekunden bei einem 2048 Bit großen Schlüssel. Da sich keine nennenswerte Verzögerung ergibt, kann der Schlüssel so groß wie möglich gewählt werden. Dies ist auch aus einem anderen Grund empfehlenswert.

Alle asymmetrischen Verfahren benutzen Algorithmen, die auf bestimmten, bisher als unlösbar geltenden mathematischen Problemen basieren. Es ist zwar sehr unwahrscheinlich, dass sie jemals vollständig gelöst werden, aber es können durchaus drastische Optimierungen eingeführt werden. Das heißt, dass zumindest theoretisch in Zukunft die Gefahr der Rückberechnung besteht, auch wenn der Aufwand heute noch im Bereich von mehreren Milliarden Jahren liegt. Auch hier gilt deshalb die gleiche Empfehlung wie bei symmetrischen Verfahren: Einen Schutz vor der Rückberechnung des Schlüssels und vor Brute-Force-Angriffen bieten nur entsprechend große Schlüssellängen, die in Abhängigkeit davon zu wählen sind, welche Informationen wie lange geschützt werden müssen.

Generell sollte man seine Schlüssellängen, sofern es das Verarbeitungsvermögen der eingesetzten Systeme und die verwendeten Algorithmen zulassen, mit einer recht kräftigen Sicherheitsreserve versehen.

Die symmetrischen Verfahren verwenden in der Regel entweder die Datenblock- oder die Datenstromverschlüsselung. Bei einer Datenblockverschlüsselung werden jeweils komplette Blöcke einer bestimmten Größe mit einem Schlüssel verschlüsselt (siehe Abbildung 4.10). Beim bekanntesten Vertreter dieser Verfahren, dem Data Encryption

Standard (DES), ist der Datenblock beispielsweise 64 Bit groß. Kleinere Blöcke werden auf die benötigte Größe aufgefüllt (Padding), um korrekt verarbeitet werden zu können.

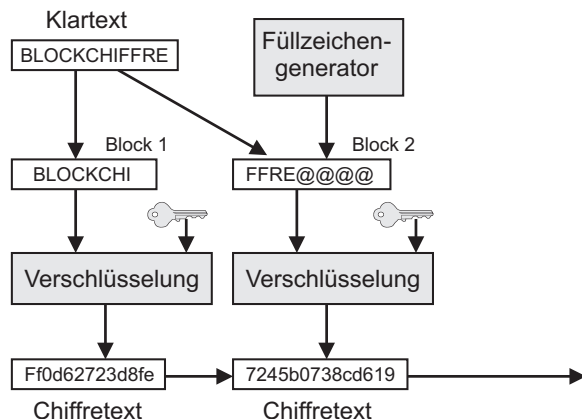


Abbildung 4.10: Die Datenblockverschlüsselung

Bei der Datenstromverschlüsselung (siehe Abbildung 4.11) wird ein Datenstrom, der aus Bits, Bytes oder anderen systemabhängigen Informationsgrößen besteht, so lange fortlaufend verschlüsselt, bis der Eingangsdatenstrom vollständig verarbeitet wurde. Die Datenstromverschlüsselung ist oft schneller als die Datenblockverschlüsselung und hat eine geringere Latenz.

Eine Schwachstelle der Datenblockverschlüsselung ist das Padding, denn die Art und Weise, wie aufgefüllt werden muss, muss sowohl dem Sender als auch dem Empfänger bekannt sein und ist sogar oft in Standards veröffentlicht. Da man statistisch gesehen davon ausgehen kann, dass im Schnitt der letzte Block zu 50% aufgefüllt wurde, bietet sich bei der Datenblockverschlüsselung immer die Möglichkeit eines Known-Plain-Text-Angriffs. Deshalb sollte die reine Datenblockverschlüsselung möglichst nie zur Übertragung von sehr kurzen Nachrichten, zum Beispiel von Telnet- oder Terminal-Sessions, angewandt werden.

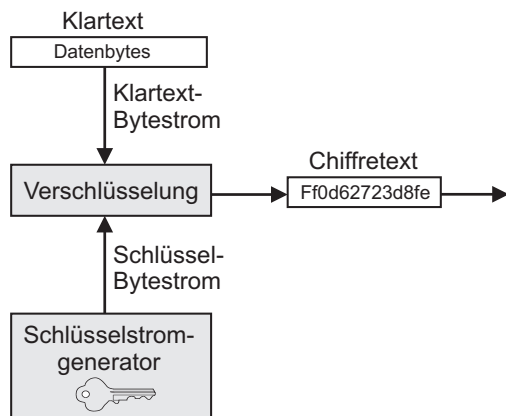


Abbildung 4.11: Die Datenstromverschlüsselung

Die Datenstromverschlüsselung liest den Eingangsdatenstrom und verschlüsselt ihn so lange mit einem Schlüsselstrom, der von einem Schlüsselstromgenerator erzeugt wird, bis der Eingangsdatenstrom vollständig bearbeitet wurde. Ein Padding ist hier offensichtlich nicht notwendig.

Allerdings wird in einigen Fällen, sowohl bei der Block- und Stromverschlüsselung, Padding ganz gezielt eingesetzt, und zwar um die tatsächliche Länge des Klartextes zu verschleiern.

Der Data Encryption Standard (DES)

Der Data Encryption Standard, DES, ist der Klassiker der symmetrischen Datenblock-Verschlüsselungsverfahren schlechthin. Er ist ein Standard des US-amerikanischen National Bureau of Standards (NBS) und ist in der Federal Information Processing Standard Publication (FIPS-Pub) 46-2 beschrieben. In diesem Dokument sind alle Details des Verfahrens beschrieben. Man kann diesen Standard benutzen, um DES zu implementieren, es ist alles vollständig beschrieben. Der Standard wurde im Juli 1977 verabschiedet und in den Jahren 1983, 1988 und 1993 wiederholt überprüft und verlängert.

Er ist also bereits über 30 Jahre alt, und es wurde noch kein Fall bekannt, in dem DES gebrochen wurde, indem eine Kryptoanalyse eingesetzt wurde, die schneller als ein Brute-Force-Angriff ist. Seine Stärke, und auch die des von ihm abgeleiteten Triple DES-Verfahrens, liegt vor allem auch darin, dass ihm die National Security Agency (NSA) seine Qualität nicht nur offiziell bescheinigt hat, sondern angeblich sogar selbst noch Verbesserungen vornahm. Wie kam es zu diesem für die NSA nun wirklich ungewöhnlichen Verhalten?

Die Entwicklung eines Vorläufers vom DES begann bereits Ende der 60er-Jahre, in einem Forschungsprojekt von IBM unter der Leitung von Horst Feistel. Das Verfahren sollte bei Lloyd's of London in den Geldautomaten, die ebenfalls von IBM stammten, eingesetzt werden und erhielt den Namen Lucifer-Chiffre. IBM sah gute Möglichkeiten einer kommerziellen Vermarktung und begann mit weiteren Entwicklungen, die das Ziel hatten, das Verfahren in einem einzigen Chip bzw. einem einzigen Hybridbaustein zu implementieren. In dieser Phase wurde bereits die NSA involviert, die dem Entwicklerteam mit Rat und Tat zur Seite stand – und heraus kam ein Algorithmus, dessen ursprünglicher 128-Bit-Schlüssel um 72 Bit auf nur noch 56 Bit verkürzt war!

Parallel dazu veröffentlichte das NBS, aus dem später das National Institute of Standards and Technology (NIST) hervorging, im Jahre 1973 eine Ausschreibung für einen einheitlichen Verschlüsselungsalgorithmus. Da die Kryptologie bis zu diesem Zeitpunkt hauptsächlich im militärischen und nachrichtendienstlichen Bereich eingesetzt wurde, hielten sich die Menge und die Qualität der eingereichten Verfahren stark in Grenzen – entweder waren die Verfahren als geheim eingestuft oder unbrauchbar. Erst nach einer zweiten Ausschreibung im Jahr 1974 reichte IBM das Lucifer-Verfahren ein. Das NBS konsultierte daraufhin externe Experten, um den Algorithmus zu beurteilen, und zwar die NSA, wie Sie richtig vermutet haben.

Um die genauen Vorgänge damals ranken sich Gerüchte und widersprüchliche Aussagen vor allem dahingehend, ob die NSA an dem Entwurf etwas verändert hatte oder nicht. Tatsache ist jedoch, dass die NSA wohl irrigerweise davon ausgegangen ist, dass DES nur in Chips oder nicht analysierbaren Hardware-Modulen implementiert werden würde. Der 56-Bit-Schlüssel schien für zivile Ansprüche ausreichend lang, aber für die NSA innerhalb von Wochen oder Tagen zu brechen. Die Veränderungen am Algorithmus gegenüber seinem ersten Entwurf, egal ob sie nun von IBM oder der NSA vorgenommen wurden, machten ihn sicher gegen die differenzielle Kryptoanalyse. Interessanterweise wurde die differenzielle Kryptoanalyse aber erst 1990 zum ersten Mal öffentlich beschrieben, fünfzehn Jahre nachdem DES dagegen sicher gemacht wurde.

Wie dem auch sei, die NSA bescheinigte dem DES, dass er ein sicherer Algorithmus sei. Die schnellste Angriffsmethode – auch heute noch – ist ein Brute-Force-Angriff. Das NBS veröffentlichte DES dann in allen Einzelheiten, die notwendig sind, um ihn – auch in Software – zu implementieren. Der Data Encryption Standard wurde in den letzten 30 Jahren zum wohl am weitesten verbreiteten Chiffrieralgorithmus aller Zeiten – trotz aller anfänglichen Kritik. Und die war sehr laut und ist bis heute nicht verstummt. Der nur 56 Bit lange Schlüssel war von Anfang an der hauptsächliche Kritikpunkt. Man hatte ihn schon damals für zu kurz gehalten und die ursprüngliche Länge von 128 Bit gefordert. Allerdings konnten sich die Kritiker damals nicht gegen die NSA durchsetzen.

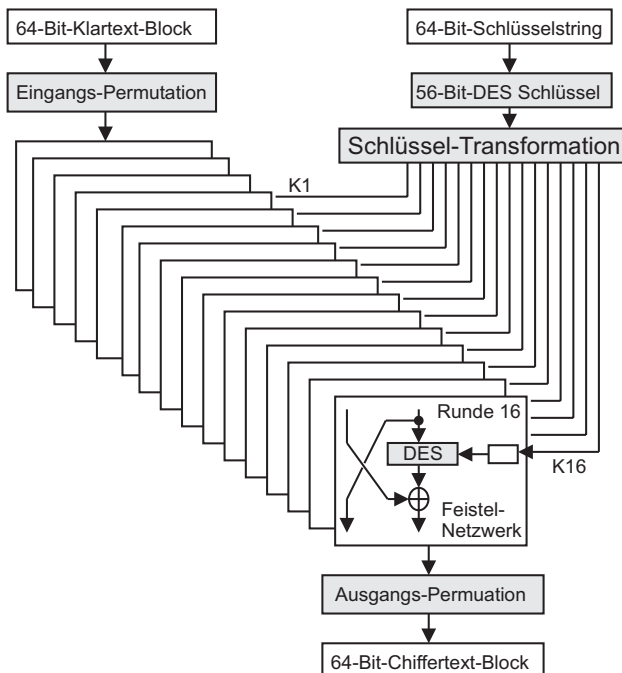


Abbildung 4.12: DES im Überblick

Das Design von DES ist sehr stark hardwareorientiert. Die Entwicklungen begannen bei IBM bereits Ende der 60er-Jahre, und die Rechner waren damals auf einem völlig anderen Leistungsniveau angesiedelt als heutige Systeme. Die Integrationsdichte von Bauelementen war nicht mit dem zu vergleichen, was heute möglich ist. Die Computer arbeiteten zu dieser Zeit nicht mit Mikroprozessoren, sondern mit Hybridelementen, kleinen Baugruppen, auf denen logische Funktionen durch diskrete Dioden und Transistoren ausgeführt wurden. Speichergrößen wurden in Byte und Kilobyte gemessen, und Massenspeicher arbeiteten mit Papier oder Magnetbändern.

In diesem technischen Umfeld, und das ist wirklich bemerkenswert, wurde ein Verschlüsselungsalgorithmus entwickelt, der in heutigen PCs, Workstations und Supercomputern arbeitet und in modernen Chips Durchsatzraten von über 1.000.000.000 Bits pro Sekunde erreicht. Weil er für den Hardware-Einsatz entwickelt wurde, ist DES nicht so leistungsfähig in Software zu implementieren. Jedoch ist dies bei der Leistungsfähigkeit heutiger Rechner kein Problem mehr – und es gibt auch schon die ersten Netzwerkadapter mit integrierten Chips zur Berechnung von DES und anderen kryptografischen Verfahren.

DES transformiert einen 64 Bit großen Klartextblock unter Verwendung eines 56-Bit-Schlüssels in einen 64 Bit großen Chiffretextblock. Der Data Encryption Standard basiert auf dem Einsatz von Feistel-Netzwerken. Er verwendet acht Feistel-Zyklen, also insgesamt 16 Runden, um dieses Ziel zu erfüllen. Bevor der Klartext der ersten Runde zugeführt wird, durchläuft er eine lineare Eingangspermutation, die nach der letzten Runde durch eine Ausgangspermutation wieder aufgehoben wird. Diese beiden Permutationen haben keinen kryptografischen Ursprung, sondern sind Anfang der 70er-Jahre aus Anforderungen heraus entstanden, die aus der damals verfügbaren Hardware resultierten. Die 16 Runden sind alle identisch aufgebaut, die DES-Funktion ist ebenfalls in allen Runden die gleiche. Der einzige Unterschied liegt in den 16 Teilschlüsseln, die als eine der beiden Eingangsvariablen der Runden verwendet werden.

Diese Teilschlüssel werden durch Schlüsseltransformationen aus dem 56-Bit-Schlüssel erzeugt und sind jeweils 48 Bit groß. Die Schlüsseltransformation ist so aufgebaut, dass nach der 16. Transformation der Originalschlüssel wieder hergestellt ist.

Die Schlüsseltransformation ist in Abbildung 4.13 illustriert. Der Schlüssel wird als 64 Bit großer Wert an den DES-Algorithmus übergeben. Dieser entfernt jedes achte Bit daraus und benutzt die verbleibenden 56 Bits als Schlüssel. Im DES-Standard wird zwar permanent (und falsch!) von einem 64-Bit-Schlüssel gesprochen, jedoch finden davon tatsächlich nur 56 Bit Verwendung. Der Grund für diesen Sachverhalt besteht darin, dass die Datenkommunikation zu Anfang der 70er-Jahre nicht die Qualität heutiger Datennetze aufwies und jedes achte Bit in dem 64-Bit-Schlüssel als Paritätsbit benutzt wurde, um Übertragungsfehler zu erkennen. Diese Bits sind natürlich nicht zufälliger Natur, sondern werden aufgrund der sieben Informationsbits berechnet, die links davon stehen. Der DES-Algorithmus selbst benutzt die Paritätsbits nicht, er entfernt sie einfach nur. Aus diesem Grund kann man DES auch beliebige 64-Bit-Schlüssel zuführen, in denen das jeweils achte Bit kein Ergebnis einer Paritätsberechnung ist, wie dies zum Beispiel auch in IPSec und IKE üblich ist, ohne das DES nicht mehr funktionieren würde.

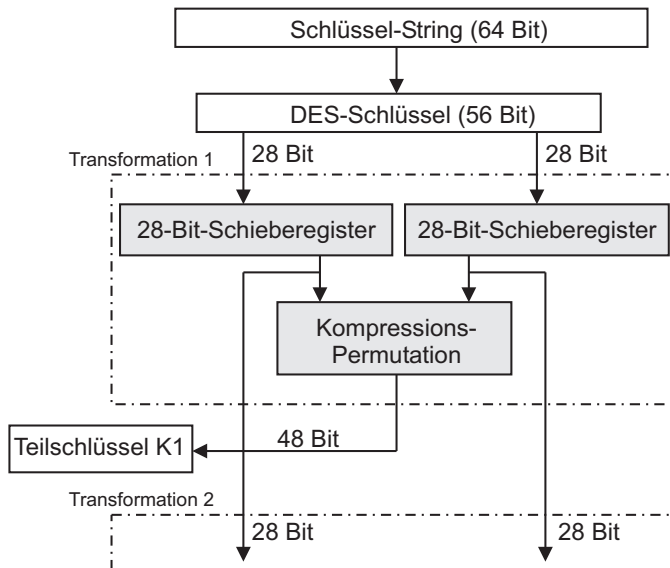


Abbildung 4.13: Die DES-Schlüsseltransformation erzeugt die Teilschlüssel für die 16 DES-Runden.

In den beiden Schieberegistern (Shift), jedes enthält eine Hälfte des 56-Bit-Schlüssels, werden die Bits rundenabhängig um ein oder zwei Bit-Positionen nach links rotiert. Rotiert heißt dabei, dass die Bits, die links „hinausgeschoben“ werden, auf der rechten Seite des Registers wieder auf der niedrigsten Bit-Position erscheinen. Die Summe der Rotationen ist 28, so dass nach 16 Runden wieder das ursprüngliche Bitmuster in den Registern steht.

In Abbildung 4.14 sehen Sie die DES-Funktion im Detail. Die Eingangswerte sind der Teilschlüssel und, im Fall der ersten Runde, die rechte Hälfte des Klartextes oder im Weiteren die rechte Hälfte des Ausgangsblocks der vorangegangenen Runde (R_n). Der Wert R_n wird in einer Expansionspermutation auf 48 Bit erweitert und mit den 48 Bit des entsprechenden Teilschlüssels exklusiv-oder-verknüpft. Das Ergebnis wird in einer nicht-linearen S-Box-Substitution wieder auf 32 Bit reduziert und durchläuft anschließend eine lineare Permutation. Das Resultat ist das Ergebnis der DES-Funktion. Dieses Ergebnis wird im Feistel-Netzwerk mit der linken Hälfte des Klartextes oder der Ausgabe der vorangegangenen Runde exklusiv-oder-verknüpft!

Die S-Boxen der DES-Funktion sind der wichtigste und kritischste Teil des Data Encryption Standards. Allein ihre Funktion entscheidet über Sicherheit oder Unsicherheit des kompletten Verfahrens. Wie die Funktionswerte, in Abbildung 4.15 sehen Sie die fünfte S-Box, erzeugt wurden, ist nicht offen gelegt!

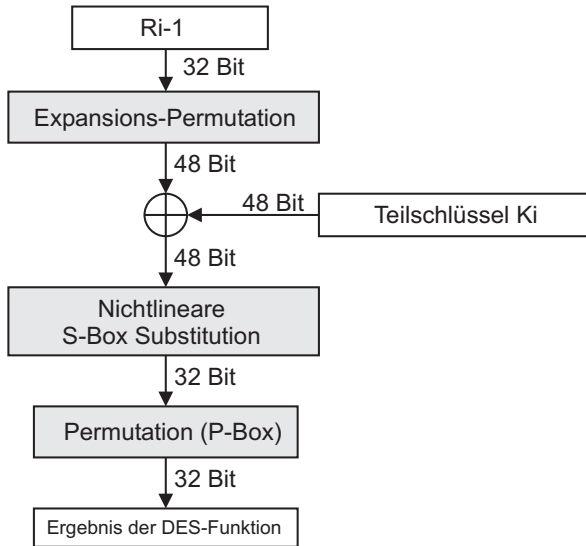


Abbildung 4.14: Die DES-Funktion

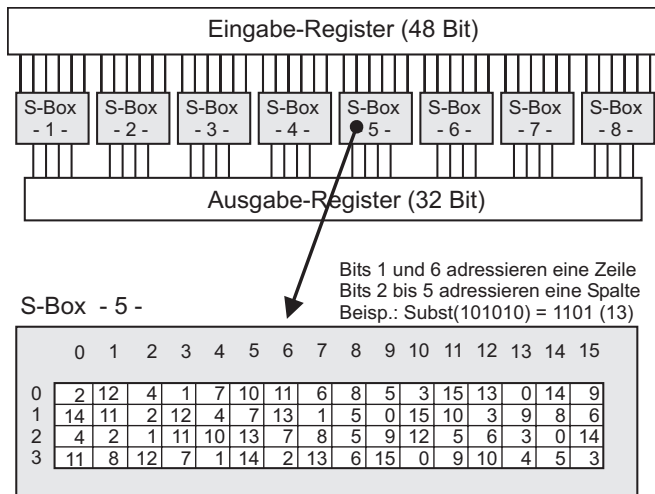


Abbildung 4.15: Die fünfte S-Box der DES-Funktion

Ob die Entwickler einfach nur gewürfelt haben oder wochenlange Berechnungen vorausgegangen sind, bleibt nach wie vor im Dunkeln. Auch heute, 30 Jahre nach der offiziellen Verabschiedung des Standards, wahren noch alle Beteiligten ihr Stillschweigen. Es ist jedoch erwiesenermaßen so, dass eine möglichst gleichmäßige, mehrfache Substitution aller Bits erreicht wurde. Lediglich inoffiziell wurde bekannt, dass die Werte in den S-Boxen, die angeblich von der NSA gegenüber dem originalen IBM-Entwurf geändert wurden, einen zuverlässigen Schutz gegen die differenzielle Kryptoanalyse bieten. Dies ist richtig, zeigt aber auch, dass dieses Verfahren, und vor allem auch der Schutz davor, der NSA schon vor 25 Jahren bekannt gewesen ist!

Durch eine DES-Entschlüsselung wird ein 64 Bit großer Chiffretext mit Hilfe des 56-Bit-Schlüssels, mit dem dieser Chiffretext erzeugt wurde, wieder in einen 64 Bit großen Klartextblock transformiert. Dies geschieht mit dem gleichen Algorithmus. Denn Sie haben im Abschnitt über Feistel-Netzwerke gelesen, dass man dieses sowohl zum Ver- als auch zum Entschlüsseln benutzen kann. Im Fall von DES laufen die Runden dazu einfach nur in umgekehrter Reihenfolge ab. Dies ist einfach dadurch zu realisieren, dass der Prozess in der ersten Runde mit dem Teilschlüssel 16 beginnt und in der letzten Runde mit dem Teilschlüssel 1 endet. Praktisch implementieren kann man dies ganz einfach, indem man die Richtung in den Schieberegistern umkehrt. Somit kann man den Algorithmus sowohl zum Verschlüsseln als auch zum Entschlüsseln benutzen. Genau genommen ist die Entschlüsselung also auch eine Verschlüsselung, denn wenn man statt des Chiffretextes einen Klartext in die Dechiffrierung einspeisen würde, würde man einen Chiffretext erhalten. Dies macht man sich auch beim Triple-DES-Verfahren zunutze, um zum DES-Algorithmus kompatibel zu sein.

Normalerweise könnte ich auf den nächsten Seiten gut 30 Jahre der Geschichte der Kryptoanalyse schildern, denn fast alle Angriffe, die es gibt, wurden gegen DES entwickelt. Aber das Thema bei DES ist gar nicht die Kryptoanalyse, sondern die Tatsache, dass ein DES-Schlüssel mit seinen 56 Bit Länge heute bereits mit Standard-Hardware gebrochen werden kann. Mit einigem Aufwand sogar in ein paar Stunden.

Hände weg von DES, der Schlüssel ist viel zu kurz! Auch wenn DES im kryptologischen Sinn nicht gebrochen ist – eine Brute-Force-Attacke ist mit wenig Aufwand machbar geworden, und das war's.

Was für ein Eigentor die NSA sich mit DES geschossen hatte, wurde eigentlich erst mit der Entwicklung und Einführung von Triple-DES deutlich. Denn dieses Verfahren, um es gleich vorwegzunehmen, kann auch eine National Security Agency nicht knacken. DES ist für die NSA kein Problem, da kommt ein Brute-Force-Angriff in Sekundenschnelle zu einem Ergebnis oder besser gesagt zu einem Schlüssel – mit Triple-DES ist allein schon der Versuch reine Zeitverschwendung.

DES war schon zu Zeiten seiner Einführung 1977 zur Zielscheibe scharfer Kritik einer ganzen Reihe von Kryptologen geworden, unter ihnen auch Persönlichkeiten wie Whitfield Diffie. Deren Kritik richtete sich dabei weniger gegen den Algorithmus selbst als gegen dessen wirksame Schlüssellänge von nur 56 Bit. Diese wurde damals schon als viel zu kurz für ernsthafte Anwendungen angesehen. Diese Kritik hat die NSA aber herzlich wenig interessiert, denn wenn jemand die technischen Ressourcen hatte, DES mit einem Brute-Force-Angriff zu entschlüsseln, dann diese Organisation.

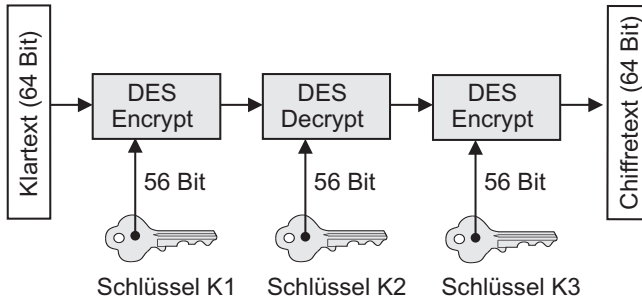


Abbildung 4.16: Triple-DES verkettet drei DES-Verschlüsselungen mit unterschiedlichen Teilschlüsseln miteinander.

Die Zeit arbeitete für die NSA, denn die Rechner wurden immer leistungsfähiger, und damit wurde die Zeit, die zur Berechnung eines DES-Schlüssels benötigt wurde, immer kürzer, während sich an der Sicherheit des Algorithmus wegen seiner festen Schlüssellänge nichts änderte. Denn schnellere Rechner verschlüsseln zwar schneller, aber dadurch nicht sicherer. Außerdem hatte die NSA ja auch noch „Plan B“ in der Schublade, die US-Exportbeschränkungen für kryptografische Produkte. Die National Security Agency hat die Macht, andere Ministerien, wie zum Beispiel das US-Außenhandelsministerium, zu ihrem verlängerten Arm zu machen und die Ausfuhr bestimmter Produkte ganz einfach zu verbieten oder wenigstens per Genehmigungsverfahren regulieren zu lassen.

Aber mit zunehmender Entwicklung der Datenverarbeitung wurde der DES-Schlüssel immer mehr zum Schwachpunkt des ganzen Verfahrens. Denn mittlerweile konnten kleine Organisationen oder sogar gut ausgerüstete Einzelpersonen den Schlüssel mit einem Brute-Force-Angriff brechen. Sie haben in diesem Kapitel erfahren, dass der Schutz gegen einen Brute-Force-Angriff genau genommen recht einfach ist, man braucht nur den Schlüssel zu verlängern. Denn mit jedem Bit mehr Schlüssellänge verdoppelt sich auch die Suchzeit. Allerdings bedingt dies einen Algorithmus mit variabler Schlüssellänge, und genau das ist DES nicht. Der DES-Schlüssel ist 56 Bit lang, und die Erzeugung der Teilschlüssel aus jedem einzelnen dieser Bits ist im Standard genau festgelegt.

Aus dieser Sackgasse schien es auf den ersten Blick keinen praktikablen Ausweg zu geben. Denn den Algorithmus zu ändern, hätte eine Inkompatibilität zu bestehenden Systemen bedeutet. Das Gleiche galt für die Entwicklung eines gänzlich neuen Algorithmus. Der Ausweg, der sich bot, hieß Triple-DES, also eine dreifache Verschlüsselung mit DES, und damit die Möglichkeit, auch Standard-DES benutzen zu können. Triple-DES verkettet, wie in Abbildung 4.16 zu sehen ist, drei DES-Durchläufe miteinander, die jeweils unterschiedliche Teilschlüssel benutzen.

Es erfolgt zuerst eine Verschlüsselung mit dem Schlüssel K1, dann eine Entschlüsselung mit dem Schlüssel K2 und zuletzt wieder eine Verschlüsselung mit dem Schlüssel K3. Somit wurde der Klartext dreifach verschlüsselt, und für einen Brute-Force-Angriff müsste man alle 2^{168} Möglichkeiten ausprobieren, welche die drei Schlüssel zusammen bieten. Und dies ist nicht möglich. Die geforderte Kompatibilität zu DES erreicht man ganz einfach dadurch, dass man den 56-Bit-DES-Schlüssel gleichzeitig allen drei Stufen von Triple-DES zuführt.

An dieser Stelle ist noch eine Sonderversion zu erwähnen, die gelegentlich zu finden ist und nur eine effektive Schlüssellänge von 112 Bit aufweist. In dieser Variante sind der erste und der letzte Schlüssel der drei Triple-DES-Stufen gleich. Somit ergibt sich für einen Brute-Force-Angriff ein zu durchsuchender Schlüsselraum von 112 Bit. Für diesen Wert ergeben sich allerdings Suchzeiten, die in Jahrtausenden gemessen werden, so dass auch hier noch von einem starken Algorithmus gesprochen werden kann.

Die nicht ganz so gute Nachricht vorweg: Der Schlüssel mit seinen 168 Bit (3 Teilschlüssel) hat leider nicht die Wirkung eines Schlüssels mit 168 Bit Länge, sondern (theoretisch) 113 Bit. Der Grund hierfür ist ein cleverer Angriff, der die Komplexität einer Brute-Force-Attacke drastisch reduzieren kann. Allerdings sind 113 Bit mit heutiger Technologie nicht durch einen Brute-Force-Angriff zu knacken.

Die Idee dieser Methode, einer typischen Known-Plaintext-Attacke, ist einfach die, dass die ersten beiden Stufen von Triple-DES mit allen 2^{112} Schlüsseln durchprobiert und die 2^{56} Resultate gespeichert werden. Hierzu werden $2^{56} * 2^{56} = 2^{112}$ DES-Transformationen benötigt. Anschließend lässt man die letzte Stufe rückwärts durchlaufen und vergleicht das Ergebnis jeweils mit den gespeicherten Resultaten. Wenn man eine Übereinstimmung gefunden hat, ist der Schlüssel bestimmt. Damit ist der Gesamtaufwand auf $2^{56} * 2^{56} + 2^{56} = 2^{113}$ DES-Transformationen reduziert. Allerdings sind bei der Theorie dieses Angriffs zwei Dinge vernachlässigt, die in der Praxis aber sehr schwer ins Gewicht fallen: der Speicherbedarf für $2^{56} * 8 = 576.460.752.303.423.488$ Bit und die Rechenzeit für die Vergleiche mit maximal jedem der 72.057.594.037.927.936 Datenblöcke.

Aber auch selbst wenn man den Speicherbedarf außer Acht lässt: Einen Rechner, der 2^{113} Schlüssel in vernünftiger Zeit durchprobieren kann, gibt es in absehbarer Zeit nicht.

Übrigens, dieser Angriff ist auch der Grund, warum es Triple-DES und nicht Dual-DES gibt. Denn hätte man nur zwei Durchläufe anstelle von dreien, wie in Triple-DES, würde sich die Komplexität beim Meet-in-the-Middle-Angriff von 2^{113} auf 2^{57} reduzieren – mit anderen Worten nur eine zweifach anstelle einer 2^{56} -fach höheren Sicherheit gegenüber DES mit 56-Bit-Schlüssel.

Andere Verfahren, wie differenzielle oder lineare Kryptoanalyse, scheitern nach wie vor an den 16 Runden der einzelnen DES-Blöcke von Triple-DES. Wie gesagt, die Entwickler der S-Boxen von DES hatten schon einen Schutz gegen diese Angriffe im Sinn!

Von der Sicherheit her kann man Triple-DES keine Punkte abziehen. Differenzielle und lineare Kryptoanalyse sind nach wie vor komplexer als eine Brute-Force-Attacke. Ein normaler Brute-Force-Angriff muss 2^{168} Schlüssel durchprobieren. Selbst ein Rechner mit 576.460 Tbyte (Tera-Byte) Primärspeicher müsste mit dem Meet-in-the-Middle-Angriff immer noch 2^{113} Schlüssel durchprobieren und 2^{56} Mal einen 8-Byte-Datenblock im über 576.460 Tbyte großen Primärspeicher suchen. Einen Rechner, der das in vernünftiger Zeit kann, wird es in den nächsten Jahren nicht geben.

Wer mit der nicht so üppigen Performance von Triple-DES leben kann, hat keinen Grund, auf einen anderen Algorithmus zu wechseln. DES und damit Triple-DES haben 30 Jahre Kryptoanalyse überstanden, ohne gebrochen zu werden. Diesen Beweis müssen andere Algorithmen erst noch antreten.

Cipher Block Chaining (CBC)

Bei aller Stärke von DES, abgesehen von seiner Grundschlüssellänge, und Triple-DES – es sind monoalphabetische Algorithmen! Das heißt, dass zu einem gegebenen Schlüssel gleiche Klartextblöcke auch gleiche Chiffretextblöcke erzeugen. Damit wäre aber die Möglichkeit eines Angriffs mit statistischen Methoden gegeben, da ein Angreifer in der Praxis sehr schnell erkennen kann, wenn sich bestimmte Muster im Chiffretext wiederholen. Zum Beispiel erzeugen die immer gleichen Signaturen in E-Mails oder auch die Mail-Header in der Regel mehrere Blöcke Chiffretext, die sich untereinander auch gleichen. Ein solcher Angriff vermag jedoch nicht den Schlüssel zu berechnen. Es ist aber möglich, auf den Klartext zurückzuschließen, was auch schon schlimm genug ist.

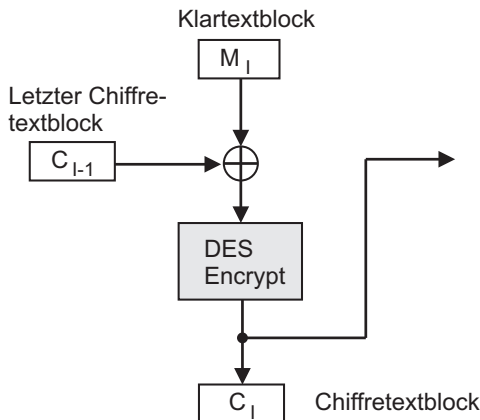


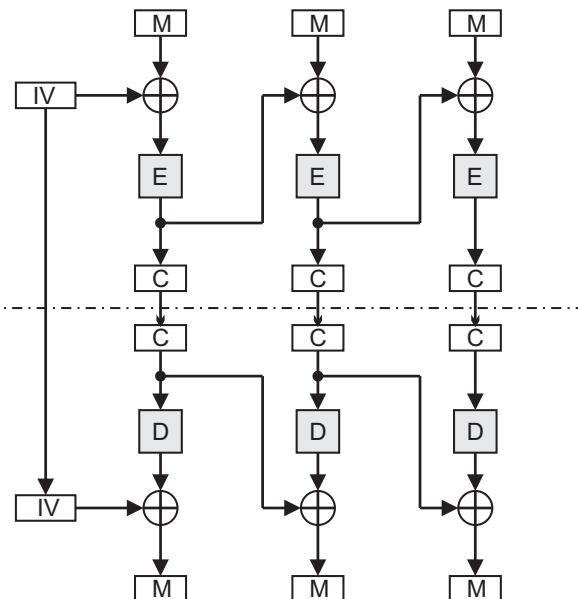
Abbildung 4.17: CBC eliminiert den Monoalphabetismus von DES und anderer Chiffrierverfahren.

Aus diesem Grunde wird DES, wie auch alle anderen Blockchiffreverfahren, meist in einer speziellen Betriebsart, dem CBC-Modus (Cipher Block Chaining) betrieben. Das Ziel dieses Verfahrens ist es, mit Hilfe eines monoalphabetischen Blockchiffreverfahrens eine polyalphabetische Ver- oder Entschlüsselung durchzuführen. Es sollen also gleiche Klartextblöcke zu einem gegebenen Schlüssel verschiedene Chiffretextblöcke erzeugen.

CBC erreicht dies, indem jeder Klartextblock vor der Verschlüsselung mit dem unmittelbar vorher erzeugten Chiffretextblock exklusiv-oder-verknüpft wird. Auf diese Weise ist garantiert, dass sich die Blöcke, die dem DES-Algorithmus als Eingangswert zugeführt werden, von den ursprünglichen Eingangsblöcken unterscheiden. Demzufolge erzeugen gleiche Klartextblöcke verschiedene Chiffretextblöcke, und es liegt praktisch eine polyalphabetische Verschlüsselung vor.

Die Funktionsweise von CBC

Wie in Abbildung 4.17 zu sehen ist, benutzt man zur Exklusiv-oder-Verknüpfung den Chiffretext des unmittelbar vorher verschlüsselten Blocks. Was aber macht man mit dem ersten Klartextblock? Um dieses Problem zu lösen, benötigt man noch einen weiteren Parameter, den so genannten *Initialisierungsvektor* (IV). Dieser Initialisierungsvektor, dessen Länge gleich der Länge der Klartextblöcke sein muss, also 64 Bit im Falle von DES, ist hinsichtlich seiner Erzeugung und Vertraulichkeit recht unkritisch. In der Praxis erzeugt man einfache Pseudo-Zufallszahlen oder benutzt den gespeicherten Chiffretext der letzten Verschlüsselung. In Abbildung 4.18 sehen Sie, dass der Initialisierungsvektor sowohl zum Ver- als auch zum Entschlüsseln benutzt wird. Im Gegensatz zum Schlüssel und zum Klartext braucht der IV aber nicht geheim zu sein. Manche Protokolle wie IP-Security übertragen ihn in jedem Datenpaket unverschlüsselt vor dem Chiffretext.



E=Encrypt, D=Decrypt, M=Klartext, C=Chiffretext, IV=Initialisierungsvektor

Abbildung 4.18: CBC-Ver- und Entschlüsselung mit explizitem Initialisierungsvektor

Wichtig für den Initialisierungsvektor ist, dass er sich während der Verwendungsdauer des Schlüssels nicht wiederholt, dass also bei verschiedenen Paketen oder Übertragungs-sitzungen nicht der gleiche verwendet wird.

Neu: Advanced Encryption Standard (AES)

Obwohl er immer noch nicht „geknackt“ ist und man mit Triple-DES das Problem des kurzen, nur 56 Bit langen Schlüssels entschärft hat, suchte man seit einiger Zeit schon nach einem Nachfolger für den doch schon etwas betagten DES-Standard. Man, das ist das amerikanische National Institute of Standards and Technology (NIST), eine Behörde des US-Handelsministeriums, die sich unter anderem mit Standards für IT-Sicherheit befasst.

Am 2. Januar 1997 begann das NIST mit den Vorbereitungen für die Ausschreibung des Advanced Encryption Standards (AES), der zur Sicherung von Informationen in US-Regierungsbehörden bis weit in das nächste Jahrhundert geeignet sein sollte. Bevor Sie Näheres zum auserwählten Algorithmus erfahren, ist es interessant, das übereinstimmend als vorbildlich bewertete Auswahlverfahren zu beschreiben, um den AES besser beurteilen zu können:

Am 12. September 1997 wurden Industrie und Forschung in aller Welt um Einreichung eines passenden Algorithmus gebeten. Dieser sollte ein nicht als geheim eingestuftes, offen gelegtes und weltweit kostenfreies Verfahren sein, das bestimmten Anforderungen genügen musste:

Der AES muss öffentlich spezifiziert werden und darf daher nicht als geheim eingestuft sein.

Der AES muss ein symmetrischer 128-Bit-Blockchiffrieralgorithmus sein.

Der AES muss eine, bei 128 Bit beginnende, nach oben erweiterbare Schlüssellänge unterstützen. Explizit unterstützt werden müssen Schlüssel mit 128, 196 und 256 Bit.

Der AES muss sowohl in Software als auch in Hardware effizient zu implementieren sein.

Der AES muss entweder frei verfügbar sein oder unter die ANSI-Patentpolitik fallen.

Die eingereichten Algorithmen wurden, sofern sie diesen Bedingungen genügen, intensiv auf folgende Punkte hin geprüft:

Sicherheit des Algorithmus

Performance und Effizienz auf verschiedenen Architekturen

Speicheranforderungen, insbesondere Arbeitsspeicherverbrauch

Eignung zur Hardware- und Software-Implementierung

Eignung zum Einsatz auf Chipkarten

Einfachheit

Flexibilität

Lizenzbedingungen

Im Hinblick auf zukünftige Entwicklungen, insbesondere im Bereich Electronic Commerce, müssen die Algorithmen sowohl sehr performant in Software als auch in Hardware implementiert werden können. Auch an SmartCards wurde gedacht, die Algorithmen sollen auf den verschiedenen Modellen mit einem geringen Arbeitsspeicher von teilweise nur 64 Byte auskommen!

Am 20. August 1998 veröffentlichte das NIST die Beschreibungen von fünfzehn Algorithmen, die von den unterschiedlichsten Organisationen aus zwölf verschiedenen Ländern zur Prüfung eingereicht wurden. Es wurde anschließend weltweit um Kommentare zu den AES-Kandidaten gebeten, um – ebenfalls mit möglichst breiter Mitwirkung – die Sicherheit, Performance und universelle Einsetzbarkeit der Verfahren kritisch zu beurteilen. Dieses Verfahren wurde am 15. April 1999 abgeschlossen.

Aufgrund dieser Kommentare und ihrer weiteren Auswertung blieben letztendlich fünf, nach Ermessen des NIST, ernst zu nehmende Algorithmen übrig. Dies waren MARS, RC6, Rijndael, Serpent und Twofish.

Im Oktober 2000 wurde der Sieger bekannt gegeben: Zur Überraschung vieler wurde der Rijndael-Algorithmus, eine belgische Entwicklung, zum neuen Standard erkoren. Die ursprüngliche Idee, als AES zwei Algorithmen einzusetzen, ein primäres Verfahren und ein Backup-Verfahren für den Fall, dass das primäre irgendwann geknackt würde, warf man wieder. Einerseits war der Auswahlprozess so gut, dass dies nicht notwendig schien, und andererseits argumentierte man damit, dass mit Triple-DES, dass ohnehin schon jeder, der zukünftig auch AES einsetzen würde, implementiert habe, bereits ein Ausweichverfahren existiere.

Und, was viele im Eifer des Gefechtes übersehen haben: Die USA hat sich damit endgültig und verbindlich von der früher praktizierten Zwei-Klassen-Kryptografie verabschiedet, denn es gibt beim AES keine schwache Verschlüsselung mehr. Bestenfalls eines starke und sehr starke Verschlüsselung, denn der kleinste AES-Schlüssel ist 128 Bit lang – ohne dass hier, zumindest zum jetzigen Kenntnisstand, zwischen effektivem Schlüssel und Bitlänge unterschieden werden muss.

An dieser Stelle vielleicht noch ein Hinweis zu den anderen fünf Finalisten: Das NIST betont ausdrücklich, dass sich, insbesondere im Hinblick auf die Sicherheit der Algorithmen, nicht herausgestellt habe, dass die „Verlierer“ schlechter oder unsicherer seien als Rijndael. Sie werden in Zukunft mit Sicherheit auf den einen oder anderen Algorithmus stoßen, da die Unternehmen ihre Entwicklungen sicher vermarkten werden. Diese Verfahren sind sicher und in entsprechenden Umgebungen schnell – in jedem Fall aber schneller als DES und mindestens genauso sicher!

Rijndael

Rijndael ist ein Algorithmus, der von den beiden Belgiern Joan Daemen und Vincent Rijmen entwickelt wurde. Obwohl der Algorithmus auch andere, größere Blocklängen unterstützen kann, wurde vorerst nur die Version mit 128 Bit Blocklänge zum Standard. Die Schlüssellänge kann 128, 192 oder 256 Bit betragen, entsprechend nennt man die unterschiedlichen Verfahren üblicherweise AES-128, AES-192 oder AES-256. Anders als praktisch alle anderen eingereichten Verfahren, beruht AES nicht auf der Struktur von Feistel-Netzwerken. Allerdings werden auch hier Verschlüsselungsrunden benutzt, deren Zahl sowohl von der Blocklänge als auch von der Schlüssellänge abhängt. Die allgemeine Formel zur Ermittlung der Rundenzahl lautet:

$$Nr = \max(Nb, Nk) + 6$$

Mit Nr = Rundenzahl, Nb = Blocklänge in Bit / 32, Nk = Schlüssellänge in Bit / 32

Für den AES mit seinen festen 128 Bit Datenblockgröße bedeutet dies Rundenzahlen von 10, 12 oder 14 bei Schlüssellängen von 128, 192 oder 256 Bit.

Im Weiteren beschreibe ich den Rijndael-Algorithmus in seiner standardisierten (AES-) Version mit 128 Bit Blocklänge. Die Beispiele beziehen sich auf einen Schlüssel mit 128 Bit Länge. Für weitergehende Informationen ist die Lektüre des AES-Drafts von Joan Daemen und Vincent Rijmen zu empfehlen.

Wer sich dieses Dokument einmal anschaut und mit der Spezifikation von DES vergleicht, der wird vermutlich zuerst einmal erschrecken. Denn hier findet sich im Gegensatz zu DES sehr viel Mathematik. Und dazu Mathematik von der Art, die viele im Verlauf ihrer Ausbildung zu Nachrichtentechnikern oder Informatikern wenn überhaupt nur gestreift haben. Andererseits stellt sich bei näherem Hinsehen heraus, dass die ganzen mathematischen Vorbemerkungen letztendlich in extrem simple Operationen wie bitweises Exklusiv-oder (xor) oder die Verkettung ähnlich einfacher Funktionen (shift left, add) münden. Das ist auch eigentlich der springende Punkt, warum sich Rijndael durchsetzen konnte. Denn seine Implementierung auf realen Systemen erlaubt die Benutzung extrem einfacher und sehr schnell auszuführender CPU-Befehle. Damit ist Rijndael auf fast allen Architekturen, angefangen bei Embedded-Microcontroller-System auf Basis der guten alten 8051-CPU bis hin zu hochparallelen Cluster-Systemen, sehr performant implementierbar.

4.8.1 Die Mathematik hinter Rijndael

In Rijndael finden einige Operationen auf Byte-Ebene statt, andere auf Basis von 4-Byte-Wörtern. Betrachten wir zuerst einmal Operationen, die auf der Ebene von Bytes definiert sind. Ein Byte wird hier als endliches Feld (Galois-Feld) mit 256 Elementen, als $GF(2^8)$ definiert. Als Schreibweise wird die Polynominalform benutzt mit Koeffizienten b_i in $\{0,1\}$. Der Körper $GF(2^8)$ wird also folgendermaßen beschrieben:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0$$

Das Byte 11001011b (das Suffix b steht für Binärschreibweise) entspricht also dem Polynom $x^7 + x^6 + x^3 + x^1 + 1$.

Der Körper $GF(2^8)$

Der Körper $GF(2^8)$ muss hinsichtlich Multiplikation und Addition abgeschlossen sein und für jedes Element genau ein inverses Element besitzen. Die Existenz eines neutralen Elements und assoziative und kommutative Eigenschaften runden das Anforderungsprofil ab. Abgeschlossen hinsichtlich Addition und Multiplikation bedeutet, dass das Ergebnis der Addition oder der Multiplikation zweier Elemente aus $GF(2^8)$ wieder ein Element von $GF(2^8)$ sein muss. Nun wird sich mancher fragen, wie das mit einem Byte möglich sein kann, denn wenn die Elemente in $GF(2^8)$ die Bytes 000h (das Suffix h steht für Hexadezimale Schreibweise) bis 0FFh sind, dann liegt das Ergebnis einer normalen Addition von beispielsweise 11001001 mit 11100001 garantiert nicht mehr in $GF(2^8)$!

Nun, ganz einfach, man definiert die Addition als spezielle Addition, die nur für Elemente dieses Körpers mit entsprechenden Regeln anwendbar ist. Und die sehen hier ganz einfach so aus, dass eine Addition ohne Berücksichtigung eines möglichen Übertrags erfolgt. Damit ist der Forderung, dass alle möglichen Ergebnisse der Addition ebenfalls Elemente von $GF(2^8)$ sein müssen, Genüge getan. Entsprechendes gilt auch für die Multiplikation in $GF(2^8)$. Damit erfüllt der Zahlenkörper $GF(2^8)$ alle Bedingungen für eine so genannte Abel'sche Gruppe.

Die Addition auf Byte-Ebene

Die Addition im Körper $GF(2^8)$ wird definiert durch das termweise Addieren der zusammengehörigen Koeffizienten Modulo 2. Modulo 2 bedeutet, dass die Ergebnisse jeweils durch 2 dividiert werden und der Rest der Division das Ergebnis darstellt. Dieser Rest kann nur 0 oder 1 sein. Überträge entstehen somit nicht, und das Ergebnis der gesamten Addition liegt garantiert immer in $GF(2^8)$. Diese Operation ist, wie Sie sicher bemerkt haben, äquivalent zu einer Exklusiv-oder-Verknüpfung (die einer bitweisen Addition ohne Übertrag entspricht) der einzelnen Bits – und genau so wird die Addition in $GF(2^8)$ in Rijndael auch implementiert! Das neutrale Element hinsichtlich der Addition in dem Körper ist 00h. Alle Elemente sind zu sich selbst invers, denn eine Exklusiv-oder-Verknüpfung mit sich selbst hat immer 0 zum Ergebnis. Die Addition in $GF(2^8)$ wird im Weiteren mit dem Symbol $\oplus$ gekennzeichnet.

Die Multiplikation auf Byte-Ebene

Die Multiplikation auf Byte-Ebene mündet ebenfalls in eine Operation, die ähnlich einfach ist wie die Exklusiv-oder-Verknüpfung der Addition in $GF(2^8)$. Allerdings sind je nach Größe des Multiplikators ein paar Operationen mehr notwendig, die allerdings ebenfalls sehr performant implementierbar sind.

In $GF(2^8)$ definiert sich die Multiplikation von Polynomen durch Multiplizieren der einzelnen Polynome Modulo $m(x)$. Das Polynom $m(x)$ ist ein irreduzibles Polynom, das sich nicht weiter zerlegen lässt. Dies ist gegeben, wenn es keine anderen Teiler außer der 1 und sich selbst hat. Für den Rijndael-Algorithmus wurde das Polynom $m(x) = x^8 + x^4 + x^3 + x + 1$ ausgewählt. Dadurch, dass $m(x)$ ein Polynom achten Grades ist, ist das Ergebnis, also der Rest einer Division durch $m(x)$, garantiert von einem Grad kleiner 8, also liegt das Ergebnis damit auch garantiert im Körper $GF(2^8)$.

Implementiert wird das Ganze folgendermaßen: Die Multiplikation eines Bytes mit 2 erfolgt durch Schieben des Byte nach links. Ist das höchstwertige Bit 1 (also der Koeffizient $b_7 = 1$), dann erfolgt eine Exklusiv-oder-Verknüpfung mit $m(x)$. Bei der Multiplikation mit Zahlen größer als 2 erfolgt diese Operation mehrfach unter Addition der Zwischenergebnisse. Diese Operation hat in Rijndael die Syntax $b = \text{xtime}(a)$. $\text{xtime}()$ ist nichts anderes als eine byteweise Schiebeoperation nach links und eine konditionelle (falls ein Übertrag erfolgt) Exklusiv-oder-Verknüpfung mit 00011011b. Die Multiplikation in $GF(2^8)$ wird im Weiteren mit dem Symbol $*$ gekennzeichnet.

Das klingt relativ kompliziert, aber an einem einfachen Beispiel kann man das Ganze vielleicht etwas besser nachvollziehen und wird dabei auch sehen, dass eine Implementierung ebenfalls recht performant möglich ist. Betrachten wir die Multiplikation von 01010111b mit 00010011b. Diese Multiplikation kann zerlegt werden in die Multiplikation von 01010111b mit 00010000b, mit 00000010b und der Addition (bezüglich $GF(2^8)$) der beiden Zwischenergebnisse mit 01010111b:

$$01010111b * 00010011b = 01010111b * (00000001b \oplus 00000010b \oplus 00010000b)$$

Zuerst lassen wir die benötigten vier Durchläufe von `xtime()` berechnen und speichern uns die benötigten Zwischenergebnisse von der Multiplikation mit 00000010b und 00010000b:

$$01010111b * 00000010b = \text{xtime}(01010111b) = 10101110b \text{ (speichern)}$$

$$01010111b * 00000100b = \text{xtime}(10101110b) = 01000111b$$

$$01010111b * 00001000b = \text{xtime}(01000111b) = 10001110b$$

$$01010111b * 00010000b = \text{xtime}(10001110b) = 00000111b \text{ (speichern)}$$

Nun haben wir alle notwendigen Summanden ermittelt und können daraus das Endergebnis berechnen:

$$\begin{aligned} 01010111b * 00010011b &= 01010111b * (00000001b \oplus 00000010b \oplus 00010000b) \\ &= 01010111b \oplus 10101110b \oplus 00000111b = 11111110b. \end{aligned}$$

Das neutrale Element ist bei der Multiplikation in $GF(2^8)$, ebenso wie bei der „normalen“ Multiplikation, die Zahl 1. Zu jedem Element kann in $GF(2^8)$ mit dem erweiterten Euklid'schen Algorithmus das multiplikativ inverse Element $b(x)$ hierzu berechnet werden, so dass gilt $a(x) * b(x) = 1$.

Die Addition auf Wortebene

Auf Wortebene, gemeint sind bei Rijndael 32-Bit-Wörter, lassen sich die vier Byte ebenfalls in Polynomialschreibweise darstellen. Hierbei sind die Koeffizienten a_i (mit i Element aus $\{0...3\}$) die Bytes bzw. die Polynome aus $GF(2^8)$: $(a_3, a_2, a_1, a_0) = a_3x^3 + a_2x^2 + a_1x^1 + a_0$. Die Addition auf der Wortebene erfolgt, indem die zusammengehörigen Koeffizienten addiert werden. Achtung, auch hier gilt wieder die Definition der Addition im Körper $GF(2^8)$! Als Beispiel nehmen wir die Addition von (a_3, a_2, a_1, a_0) mit (b_3, b_2, b_1, b_0) :

$$\begin{aligned} (a_3x^3 + a_2x^2 + a_1x^1 + a_0) \oplus (b_3x^3 + b_2x^2 + b_1x^1 + b_0) = \\ (a_3 \oplus b_3)x^3 + (a_2 \oplus b_2)x^2 + (a_1 \oplus b_1)x^1 + (a_0 \oplus b_0) \end{aligned}$$

Die Multiplikation auf Wortebene

Die Multiplikation auf Wortebene schaut etwas komplexer aus. Auch hier wird wieder ein Polynom definiert, um eine Reduzierung auf einen Grad unter 4 zu erreichen., nämlich $M(x) = x^4 + 1$. Für die gesamte Multiplikation zweier Wörter $c(x) = a(x) * b(x)$ gilt:

$$a(x) * b(x) = c(x) = c_3x^3 + c_2x^2 + c_1x^1 + c_0 \text{ mit:}$$

$$c_0 = a_0 * b_0 \oplus a_3 * b_1 \oplus a_2 * b_2 \oplus a_1 * b_3$$

$$c_1 = a_1 * b_0 \oplus a_0 * b_1 \oplus a_3 * b_2 \oplus a_2 * b_3$$

$$c_2 = a_2 * b_0 \oplus a_1 * b_1 \oplus a_0 * b_2 \oplus a_3 * b_3$$

$$c_3 = a_3 * b_0 \oplus a_2 * b_1 \oplus a_1 * b_2 \oplus a_0 * b_3$$

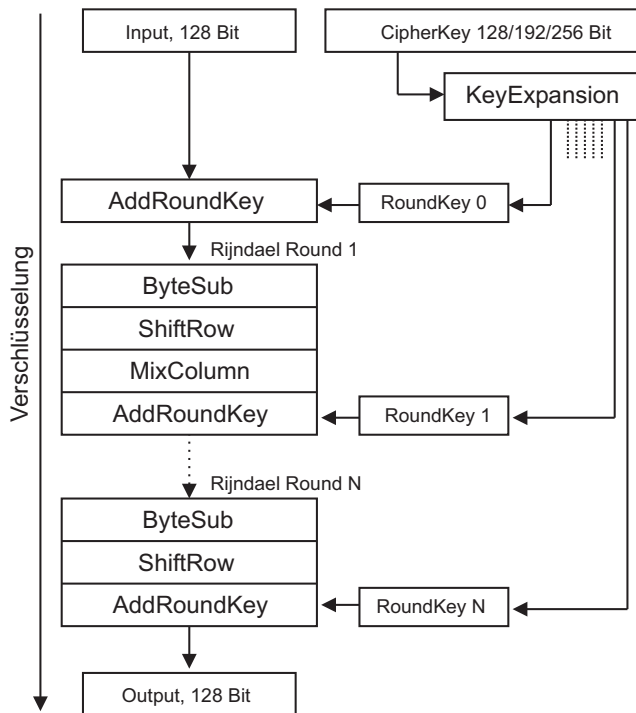


Abbildung 4.19: Die Rijndael-Verschlüsselung im Überblick

4.8.2 Der Rijndael-Algorithmus

Nachdem die zugrunde liegenden mathematischen Grundlagen und Vereinbarungen besprochen wurden, wird nun der Rijndael-Algorithmus im Detail beschrieben. Im Weiteren wird die AES-Variante von Rijndael behandelt, die mit Datenblöcken (Klartext und Chiffretext) von 128 Bit (16 Byte) und Schlüssellängen von 128, 192 oder 256 Bit arbeitet.

Die Bytes des Eingangsdatenblocks werden intern in einer als „State“ bezeichnete Matrix von vier Zeilen und vier Spalten dargestellt. Rijndael benutzt diese Matrix für Operationen auf Byte-, Spalten- oder Zeilenebene. Der Algorithmus basiert nicht auf Feistel-Netzwerken, obwohl auch er mit mehreren Runden arbeitet, um den Klartext in den Chiffretext zu transformieren. Die Runden sind in völlig unterschiedlich arbeitende, so genannte Layer unterteilt, die auf alle Elemente (Byte, Zeilen oder Spalten) des States angewendet werden. Das Ziel der Anwendung dieser verschiedenen Layer ist ein möglichst hoher Schutz gegen die lineare und differenzielle Kryptoanalyse. Interessanterweise findet auch hier, wie in DES oder vielen anderen Algorithmen, eine Kombination aus Substitution und Diffusion in mehreren Stufen (Produktchiffre) Anwendung. Bei Rijndael kommt als dritte Schicht noch die Einbeziehung des Schlüssels hinzu.

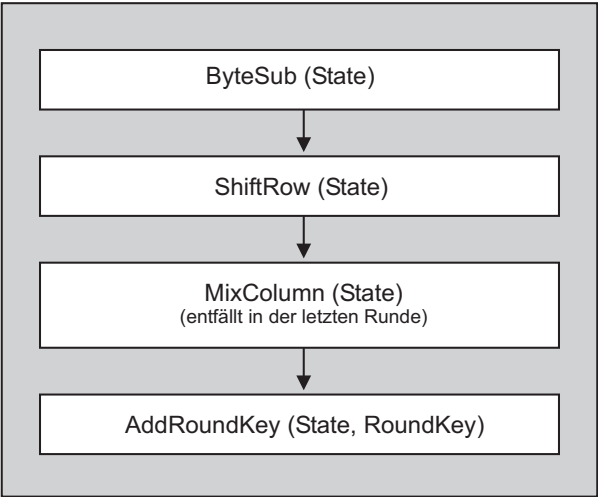


Abbildung 4.20: Eine Runde von Rijndael

Der Linear Mixing Layer soll für eine sehr hohe Diffusion über mehrere Runden hinweg sorgen.

Der Non-Linear Layer arbeitet mit der parallelen Anwendung von S-Boxen, die sehr sorgfältig auf maximale Nichtlinearität optimiert wurden.

Der Key Addition Layer verknüpft den jeweiligen Rundenschlüssel mit dem jeweiligen State exklusiv-oder. Dies ist übrigens die einzige Funktion, in der AES vom Benutzer-schlüssel abhängig ist.

Die Anzahl der Runden in AES leitet sich aus der Größe des Schlüssels ab.

Schlüssellänge:	128 Bit	192 Bit	256 Bit
Anzahl der Runden:			

Tabelle 4.2: Die Anzahl der AES-Runden in Abhängigkeit von der Schlüssellänge

Die Anzahl der Rundenschlüssel ist um eins größer als die Anzahl der Runden, da zusätzlich noch eine initiale Schlüsseladdition erfolgt. Da eine Anwendung der Rundenschlüssel auf den State erfolgt, müssen sie eine Länge von 128 Bit haben, demnach muss ein 128-Bit-Schlüssel beispielsweise auf insgesamt 1408 Bit expandiert werden, die in elf Matrizen mit je vier Zeilen und vier Spalten gespeichert werden. In Implementierungen, in denen ausreichend Arbeitsspeicher zur Verfügung steht, können die Schlüssel alle vorberechnet und gespeichert werden. In Implementierungen, in denen Arbeitsspeicher knapp ist (z.B. SmartCards, MCUs usw.), können die Rundenschlüssel auch on the fly berechnet werden, hier benötigt man nur einen einzigen Buffer mit der Größe des Chiffrierschlüssels.

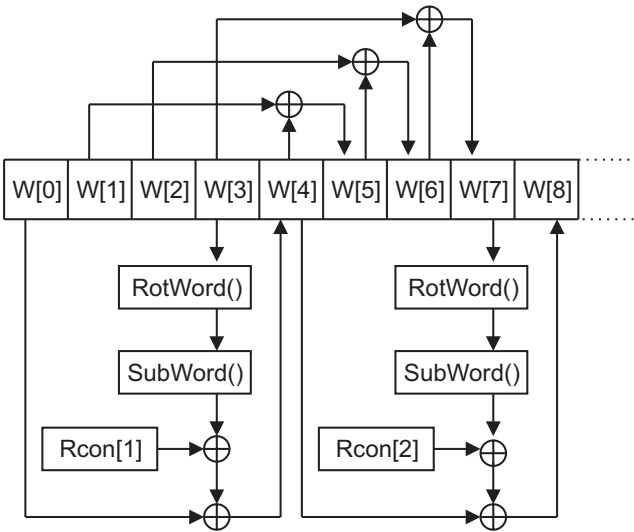


Abbildung 4.21: Die Erzeugung der notwendigen Rundenschlüssel aus dem Chiffrierschlüssel

Grundlegender Ablauf

Für unsere Betrachtung nehmen wir an, dass die elf benötigten Rundenschlüssel im Voraus berechnet werden und zu Beginn jeder Runde zur Verfügung stehen. Zu Anfang werden die Bytes des Eingangsdatenblocks in den State (Input-State) eingelesen. Dabei werden die Bytes wie in Tabelle 4.3 angeordnet.

Tabelle 4.3: Die Anordnung der Bytes im Input-State

Der Schlüssel (hier: 128 Bit) muss auf elf Rundenschlüssel von 128 Bit Größe expandiert werden. Hierzu wird ein so genannter Expanded Key, ein lineares Array aus 4-Byte-Wörtern $w[i]$, erzeugt, aus dem die einzelnen Rundenschlüssel entnommen werden. Der erste Rundenschlüssel ist der Chiffreschlüssel selbst. Die anderen Rundenschlüssel werden gemäß Abbildung 4.21 erzeugt. Jedes Wort, das nach dem Cipher Key ($w[0]...w[3]$) kommt, ist eine Exklusiv-oder-Verknüpfung des Wortes $w[i-1]$ und des Wortes vier Positionen vorher. Für Wörter, deren Position ein Vielfaches von vier ist, wird vor der Exklusiv-oder-Verknüpfung eine Transformation von $w[i-1]$ durchgeführt, und es wird zusätzlich eine Rundenkonstante $rcon[i]$ Exklusiv-Orer verknüpft. Die Transformation besteht aus einem zyklischen Schieben der Bytes eines Wortes um eine Position nach links und einer anschließenden byteweisen S-Box-Substitution.

Die Rundenkonstante $rcon[i]$ wird definiert als Array $(RC[i], 00h, 00h, 00h)$, mit $RC[0] = 01h$ und $RC[i] = xtime(rcon[i-1])$.

Nun werden die Rijndael-Runden nacheinander durchlaufen. Bis auf die letzte Runde, in der die MixColumns-Funktion fehlt, sind alle Runden gleich aufgebaut. Vor der ersten Runde wird zusätzlich der Schlüssel ($w[0]...w[3]$) mit dem State exklusiv-oder verknüpft. Nach der letzten Runde enthält das State-Array den Rijndael-Chiphertext.

Der Aufbau der Runden ist immer gleich, es erfolgen der Reihenfolge nach folgende Operationen auf dem State-Array:

SubBytes(state) ist eine klassische S-Box-Substitution, die auf jedes Byte des State-Arrays angewendet wird. Die S-Box, eine Tabelle aus 256 Byte, ist im Gegensatz zu DES aus einem expliziten, veröffentlichten, mathematischen Algorithmus abgeleitet. Das hat den Vorteil, dass in Systemen mit begrenzten Speicherressourcen die S-Box nicht gespeichert werden muss, sondern die benötigten Werte on the fly berechenbar sind. Für die Entschlüsselung wird eine andere, zu dieser inverse, S-Box benötigt.

ShiftRows(state) verschiebt die Zeilen im State-Array. Die Funktion ist umkehrbar.

MixColumns(state) ordnet jeder Spalte des State-Arrays eine neue Spalte zu. Auch für diese Funktion existiert eine inverse Version.

AddRoundKey(state, roundkey[round+1]) ist eine einfache Exklusiv-oder-Verknüpfung des jeweiligen Rundenschlüssels mit dem State-Array. Die Exklusiv-oder-Funktion ist zu sich selbst invers.

Diese Funktionen, bei denen SubBytes() die einzige nichtlineare Komponente und AddRoundKey() die einzige schlüsselabhängige Operation ist, bieten in ihrer Summe und der hinreichenden Wiederholung in (hier) zehn Runden einen bislang zuverlässigen Schutz gegen alle bekannten, sowohl theoretischen als auch praktisch durchführbaren Angriffe. Alle Angriffe, die bisher beschrieben wurden, sind rein akademischer Natur und brauchen uns bis zur Vorstellung des ersten funktionsfähigen, produktiv einsetzbaren Quantencomputers keine größeren Sorgen zu machen. Und ab diesem Tag kann man den größten Teil der heutigen Kryptografie ohnehin vergessen.

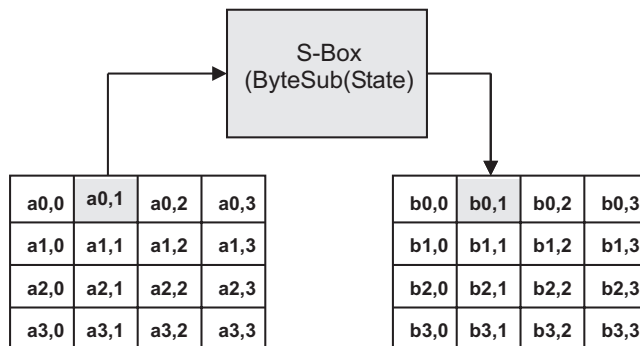


Abbildung 4.22: Die SubBytes()-Funktion

Bevor es an die Entschlüsselung in Rijndael geht, werden die verschiedenen Rundenfunktionen etwas detaillierter erläutert.

Die S-Box in Rijndael kann entweder statischer oder dynamischer Natur sein. In erstem Fall werden für alle 256 möglichen Werte eines Bytes die Werte der S-Box vorberechnet und je nach Hardware-Architektur im RAM gespeichert oder bereits bei der Herstellung im ROM abgelegt. Die dynamische Variante erlaubt es, den Substitutionswert jedes Bytes bei Bedarf zu berechnen. Zu diesem Zweck wurde die Berechnung der Bytes der S-Box offen gelegt. Sie erfolgt in zwei aufeinander folgenden Transformationen, die auf das zu substituierende Byte angewendet werden.

In der ersten Transformation wird das multiplikativ inverse Element im Körper $GF(2^8)$ ermittelt. Multiplikativ invers bedeutet, dass das Produkt beider Elemente das neutrale Element bezüglich der Operation ergibt. Mit anderen Worten, wenn $a(x)$ das multiplikativ inverse Element von $b(x)$ ist, dann ist $a(x) * b(x) = 1 \text{ mod } m(x)$. Der erweiterte Euklid'sche Algorithmus kann zur Ermittlung dieses Elementes eingesetzt werden. Der Sonderfall hier ist das Null-Byte (00h), das selbst-invers ist, also den Wert 00h behält. Diese Transformation bildet den Schutz des Algorithmus gegen die lineare und differenzielle Kryptoanalyse.

Als zweite Transformation findet eine so genannte affine Substitution statt, die das Ergebnis $b(x)$ aus dem Eingangsbyte $a(x)$ wie folgt bildet:

$$b(x) = (x^7 + x^6 + x^2 + x) + a(x) (x^7 + x^6 + x^5 + x^4 + 1) \text{ mod } (x^8 + 1)$$

Die Addition mit $(x^7 + x^6 + x^2 + x)$ soll dafür sorgen, dass ein Null-Byte (00h) nicht noch einmal auf sich selbst abgebildet wird. Die Reduzierung mit $\text{mod } (x^8 + 1)$ garantiert, dass $GF(2^8)$ abgeschlossen bleibt. Diese Transformation dient zum Schutz gegen Interpolationsangriffe. Die S-Box und damit auch `SubBytes()` sind invertierbar. Es ist zudem garantiert, dass die S-Box kein Byte auf sich selbst abbilden kann.

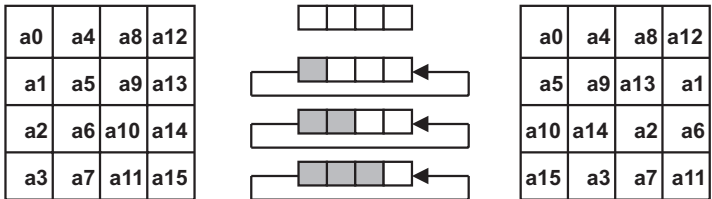


Abbildung 4.23: Die `ShiftRows()`-Funktion

In der Regel wird man die Werte der S-Box vorberechnen oder die bereits vorberechneten Werte fest in der Implementierung benutzen (Code, ROM usw.). Ich habe darauf verzichtet, die insgesamt 512 Bytes der S-Box und ihrer Inversen hier aufzulisten. Jemand, der AES implementieren muss, kann sie aus der einschlägigen Fachliteratur entnehmen oder sich einfach einer der vom NIST veröffentlichten Referenzimplementierung bedienen. Da gibt es den C- und Java-Quelltext auch gleich mit dazu.

Diese Operation besteht aus dem zyklischen Verschieben von drei Zeilen des State-Arrays nach links. Die erste Zeile bleibt unverändert. Zeile 2 wird um 1 Byte-Position, Zeile 3 um zwei Byte-Positionen und Zeile 4 um drei Byte-Positionen nach links verschoben. Zyklisch bedeutet, dass die Bytes, die links aus der Zeile „herausgeschoben“ wer-

den, rechts wieder „hereingeschoben“ werden. Damit ist diese Operation einfach umkehrbar, indem man die Richtung der Schiebeoperation umkehrt.

Hier werden jeder Spalte des State-Arrays neue Spalten(-werte) zugeordnet. Die Spalten werden dabei als Polynome über $GF(2^8)$ behandelt und modulo($x^4 + 1$) mit dem Polynom $c(x)$ mit $c(x) = 03x^3 + 01x^2 + 01x + 02$ multipliziert. Die Werte 03, 02 und 01 sind als Bytes zu interpretieren. Für die vier Byte einer Spalte sieht die Berechnung wie folgt aus:

$$b_0 = (02 * a_0) \oplus (03 * a_1) \oplus a_2 \oplus a_3$$

$$b_1 = a_0 \oplus (02 * a_1) \oplus (03 * a_2) \oplus a_3$$

$$b_2 = a_0 \oplus a_1 \oplus (02 * a_2) \oplus (03 * a_3)$$

$$b_3 = (03 * a_0) \oplus a_1 \oplus a_2 \oplus (02 * a_3)$$

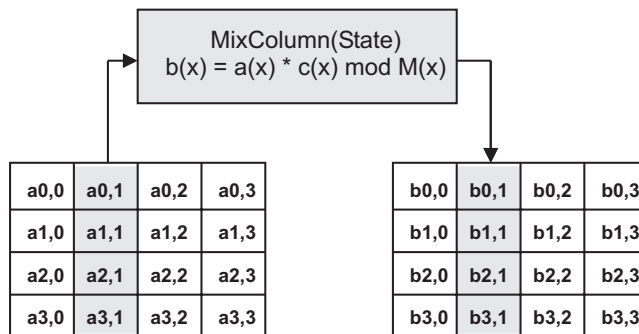


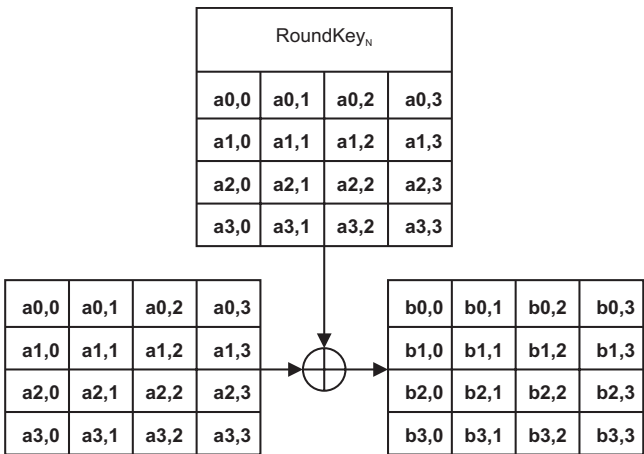
Abbildung 4.24: Die MixColumns()-Funktion

Diese Operation wird gleichermaßen auf alle Spalten angewendet. Auch sie ist umkehrbar, indem anstelle des Polynoms $c(x)$ das Polynom $d(x)$ mit $d(x) = 0Bx^3 + 0Dx^2 + 09x + 0E$ verwendet wird.

In Rijndael ist die Schlüsselabhängigkeit der Verschlüsselung ausschließlich durch die AddRoundKey()-Funktion realisiert. Sie benötigt den jeweils aktuellen Rundenschlüssel und führt eine byteweise Exklusiv-oder-Verknüpfung der Bytes des State-Arrays mit den Bytes des Teilschlüssels aus. Einer Spalte (4 Byte) des State-Arrays entsprechen in aufsteigender Reihenfolge die Bytes eines der vier Wörter (4 Byte) des für diese Runde anzuwendenden Rundenschlüssels. Den genauen Zusammenhang können Sie der Abbildung 4.25 entnehmen. AddRoundKey() ist zu sich selbst invers, da die Funktion ausschließlich aus einer Exklusiv-oder-Verknüpfung besteht.

Rijndael wurde unter anderem auf optimale Performance hin entwickelt. Es ist möglich, sehr viel mit festen Tabellenfunktionen (Table Lookups) und Exklusiv-oder-Verknüpfungen (XOR) zu arbeiten. So ist eine Implementierung möglich, die zwar 4 KByte an Spei-

cherplatz für Tabellen benötigt, aber dafür eine komplette AES-Runde mit ganzen vier Table Lookups und vier XORs realisieren kann – eine immense Geschwindigkeit!



Mit: $b_{0,0} = a_{0,0} \text{ xor } k_{0,0}$, $b_{1,0} = a_{1,0} \text{ xor } k_{1,0}$ $b_{3,3} = a_{3,3} \text{ xor } k_{3,3}$

Abbildung 4.25: Die `AddRoundKey()`-Funktion

Außerdem ist Rijndael sehr gut parallelisierbar und damit bei geeigneter Hardware leicht bis zum Multi-Gigabit-Durchsatz skalierbar – zu bezahlbaren Preisen. Kurz vor Beendigung dieses Buches wurde zum Beispiel von Nortel die weltweit erste Verschlüsselung für optische Übertragungssysteme mit 10 GBit/s mit AES und 256-Bit-Schlüssel demonstriert.

Auch die Entwicklung von AES-Chips dürfte zu deutlichen Performance-Steigerungen gegenüber Software-Lösungen führen. Die elementaren Funktionen sind gut in Hardware zu implementieren. Ob allerdings ein Steigerungsfaktor von Soft- zu Hardware-Lösungen wie bei DES möglich ist, wage ich im Augenblick noch zu bezweifeln. Einige Chipsätze, die AES-Hardware-Support bieten, lassen da Zweifel aufkommen. Allerdings sind die besagten Chips auch keine reinen AES-Chips, sondern Allrounder, die gleichzeitig DES, Triple-DES, RC4, MD5, SHA1 und andere Algorithmen implementiert haben.

Bei der Entschlüsselungsfunktion von Rijndael werden die inversen Versionen der vier Rundenfunktionen eingesetzt. Abbildung 4.26 gibt einen Überblick über die Entschlüsselung. Die Reihenfolge der Rundenschlüssel wird beim Dechiffrieren umgekehrt. Prinzipbedingt ist die Dechiffrierung etwas langsamer als die Chiffrierung.

Tatsache ist, dass heute (Stand Juli 2005) kein Angriff bekannt ist, der schneller als ein Brute-Force-Angriff ist. Somit gilt AES bzw. Rijndael mit Fug und Recht als sicher. Nicht zuletzt wurde Rijndael auch speziell gegen Angriffe wie differenzielle oder lineare

Kryptoanalyse entwickelt. So ist denn auch die Anzahl der Runden so gewählt worden, dass eine hohe Sicherheitsreserve (Security Margin) auch gegen verbesserte Versionen dieser Methoden existiert.

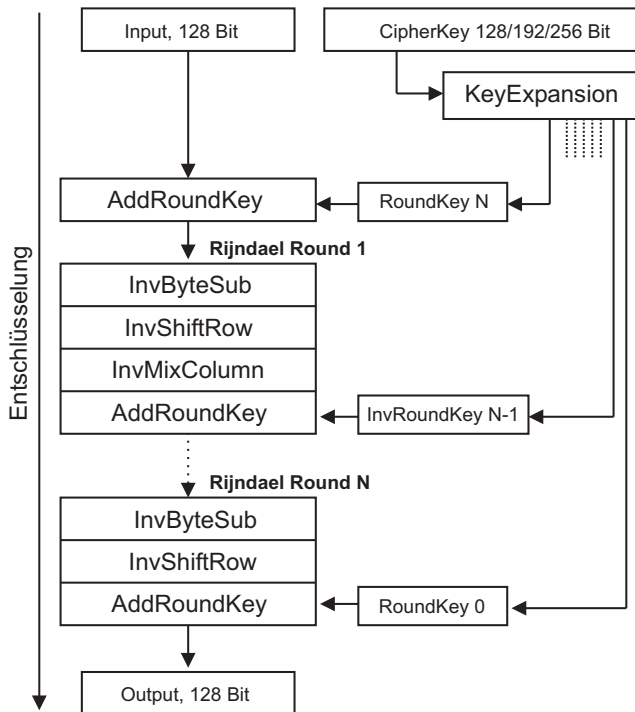


Abbildung 4.26: Die Rijndael-Entschlüsselung im Überblick

Der schnellste, bislang bekannte Angriff auf Rijndael, der Kollisionsangriff (Collision Attack), kann sieben Runden aller Rijndael-Versionen brechen. Aber auch die Collision Attack braucht für die kleinste Schlüssellänge immer noch länger als eine Brute-Force-Attacke. Allerdings nur ganz wenig länger!

AES bzw. Rijndael sind im Vergleich zu anderen Algorithmen noch relativ jung. Das hat zur Folge, dass sich die Kryptologen noch nicht so intensiv und ausführlich mit dem AES beschäftigt haben wie mit anderen Algorithmen.

Algebraische Attacken gegen Rijndael

Es gibt aber einige Punkte, die man zum jetzigen Zeitpunkt doch schon erwähnen sollte. Im Gegensatz zu einigen anderen Algorithmen sind alle Transformationen invertierbar. Die S-Box wird aus einem mathematischen Verfahren abgeleitet. Die Entschlüsselung erfolgt wiederum durch andere, dazu inverse mathematische Verfahren. Hier setzt auch die Kritik, vielleicht in vielen Fällen auch das unbestimmte Bauchweh an, dass zu den mathematischen Funktionen unter Umständen äquivalente Funktionen existieren, die vielleicht viel schneller sind oder andere, für Angriffe auf Rijndael nutzbare Eigenschaften aufweisen.

Die Wissenschaftler Nils Ferguson, Richard Schoepel und Doug Whiting haben im Jahr 2001 den Rijndael-Algorithmus als geschlossenes algebraisches System dargestellt, das in etwa wie ein sehr langer Kettenbruch aussieht. Ausmultipliziert hätte die Version für den AES (128-Bit-Schlüssel, 128-Bit-Blockgröße) etwa 2^{35} Terme. Eine Kryptanalyse mit einem entsprechenden Gleichungssystem als Lösung würde in eine Gleichung mit 2^{26} Unbekannten münden. Für derartige Gleichungen sind noch keine Lösungsmethoden bekannt. Noch nicht.

Rijndael baut aber genau darauf, dass es für solche Formelsysteme keine Lösungsmöglichkeit geben kann. Das allerdings ist mathematisch überhaupt nicht bewiesen.

Für einigen Aufruhr und extrem kontroverse Diskussionen sorgten dann auch Nicolas T. Courtois und Josef Pieprzyk im Jahr 2002 mit ihrem Artikel „Cryptanalysis of block ciphers with overdefined systems of equations“. Die beiden hatten ein von Adi Shamir entwickeltes Dechiffrierverfahren (XL) erweitert und behaupteten, mit ihrer XSL-Methode das Knacken von AES (128 Bit) mit einem Aufwand von 2^{100} anstelle der 2^{128} AES-Transformationen durchführen zu können. Damit wäre der AES wissenschaftlich gesehen gebrochen.

Courtois und Pieprzyk stellten nämlich fest, dass die S-Box von Rijndael durch implizite quadratische boolesche Gleichungen beschrieben werden kann. Um die S-Box zu beschreiben, genügen acht dieser Gleichungen. Allerdings bemerkten die beiden, dass es viel mehr Gleichungen dieses Typs gibt. Und diese Mehrzahl an Gleichungen (overdefined system of equations) soll nun die Komplexität des Angriffs auf AES reduzieren. Allerdings wurde der damit implizierte Behauptung, dass das harte mathematische Problem der Lösung multivarianter quadratischer Gleichungen durch einen bestimmten Algorithmus in weniger als exponentieller Zeit gelöst werden kann, heftig widersprochen. Die Diskussion hierüber dauert noch an, und ein praktisch durchgeführter Angriff scheitert daran, dass die Rechenleistung, um 2^{100} Schlüssel durchzuprobieren, in den nächsten Jahren nicht zur Verfügung stehen wird.

Es gibt zurzeit keinen praktisch durchführbaren Angriff auf Rijndael mit einer Komplexität, die geringer ist als eine Brute Force Attack. Die theoretischen Erwägungen zu algebraischen Angriffen sind mathematisch nicht bewiesen und werden zurzeit heftig diskutiert. Angesichts der Rechenleistung, die heute und in den kommenden Jahren zur Verfügung stehen wird, wären diese Angriffe vorerst ohnehin rein akademischer Natur. Und auch bei Rijndael kann man das tun, was man bei anderen Algorithmen tut, wenn der Schlüssel zu kurz erscheint – einfach einen längeren nehmen, zum Beispiel 256 anstelle 128 Bit.

Seine nach wie vor häufige Verwendung unter anderem in SSL oder WPA (WiFi Protected Access) erfordert es, diesen Algorithmus etwas detaillierter zu besprechen. Außerdem ist RC4 eines der wenigen Stromchiffrierverfahren, das in modernen Netzwerk-Sicherheitsprotokollen Verwendung findet.

Obwohl es von einigen Halbwissenden andauernd wieder behauptet wird: RC4 ist nicht gebrochen. Das Pech von RC4 ist einfach, dass genau diese Art von Halbwissenden RC4 mehrfach in Verschlüsselungssysteme eingebaut haben, ohne sich über das Verfahren selbst, seine Schwächen in bestimmten Fällen oder die Grundlagen der Schlüsselerzeugung tiefere Gedanken gemacht zu haben. Doch dazu mehr im Abschnitt zur Kryptanalyse von RC4.

4.9.1 Geschichtliches

RC4, Ron's Cipher 4, benannt nach seinem Entwickler, dem Krypto-Akkord-Entwickler Ron Rivest, wurde 1987 von diesem für seine Firma RSA Data Security Inc. (heute RSA Security Inc.) ausgetüftelt. Neben der Sicherheit war vor allem eine hohe Performance in Software gefordert, mit Zielplattformen vor allem im PC- und 8-Bit-Microcontroller-Bereich. Obwohl es unter Kryptologen äußerst verpönt ist und ausgerechnet ein Mann wie Ron Rivest es hätte besser wissen müssen, versuchte RSA Data Security allen Ernstes, den Algorithmus geheim zu halten, obwohl er sich sehr schnell verbreitete und vor allem in Echtzeitsystemen oder echtzeitähnlichen Umgebungen gerne eingesetzt wurde.

Natürlich interessierten sich die einschlägig bekannten Hacker für den Algorithmus, und 1994 wurde der Quellcode bzw. ein C-Quellcode, der durch Reverse Engineering ermittelt wurde, anonym in Usenet veröffentlicht. Daraufhin testete ein anderer ganz legal den Quellcode mit dem RC4-Algorithmus, der in RSA's BSAFE-Entwicklungs-Toolkit enthalten war, und die Testvektoren produzierten exakt die gleichen Ergebnisse.

Damit war der Fachwelt die Funktion eines extrem performanten und bei richtigem Einsatz auch sehr sicheren Algorithmus im Detail zugänglich.

4.9.2 Der RC4-Algorithmus

RC4 besteht aus zwei Phasen, der Initialisierung (KSA, Key Scheduling Algorithm) und dem eigentlichen RC4-Algorithmus, einer Art Pseudozufallsgenerator (PRGA, Pseudo Random Generator Algorithm). Der PRGA erzeugt den Schlüsselstrom, eine quasi zufällige, sich nicht wiederholende Folge an Byte-Werten zwischen 0 und 255. Die eigentliche Transformation von Klartext und Chiffretext erfolgt durch Exklusiv-oder-Verknüpfung eines Bytes Klartext mit einem Byte aus der S-Box. RC4 kann 2^{1700} verschiedene Zustände annehmen und stellt damit einen fast idealen Stromchiffrieralgorithmus dar.

Die Initialisierung von RC4

RC4-Schlüssel können eine beliebige Länge haben, üblich sind 40 bis 256 Bit. Der KSA erzeugt aus dem Grundschlüssel eine S-Box mit 256 verschiedenen Bytes. Die Permutation wird ausschließlich vom Schlüssel bestimmt. Am einfachsten stellt sich der Algorithmus in, stark an C angelehntem, Pseudo-Code dar. $K[i]$ ist ein temporäres Schlüsselfeld mit 256 Byte Größe, $S[i]$ die S-Box, ebenfalls mit 256 Byte Größe.

Zuerst wird das temporäre Schlüsselfeld $K[i]$ so lange mit dem Chiffrierschlüssel aufgefüllt, bis alle 256 Byte in $K[i]$ initialisiert sind. Ist der Schlüssel kürzer als 2048 Bit, wird er so lange wiederholt, bis das Feld gefüllt ist. Bei Schlüsseln größer als 2048 Bit (ein Fall, den ich noch niemals erlebt habe) sind nur die ersten 2048 Bit signifikant.

Die Initialisierungsfunktion KSA() definiert sich wie folgt:

```
;Füllen der S-Box mit 256 verschiedenen Bytes
temp, i, j: Byte
S, K: Byte[0..255]
{
    For i = 0 to 255 {
        S[i] := i
    }
;Schlüsselabhängige Permutation der S-Box
;i garantiert, dass alle 256 Positionen der S-Box bearbeitet werden
;j liefert die schlüsselabhängige Zufallskomponente
j := 0
for i = 0 to 255 {
    j := (j + S[i] + K[i]) mod 256
    temp := S[i]
    S[i] := S[j]
    S[j] := temp
}
}
```

Danach ist die S-Box initialisiert und kann benutzt werden. Das Schlüsselfeld K[i] wird nicht mehr benötigt.

Nachdem die S-Box initialisiert wurde, kann RC4 anfangen, seinen Schlüsselstrom erzeugen. Eine absolut sichere Verschlüsselung würde man mit einem Schlüsselstrom aus echten Zufallszahlen erzeugen. Allerdings würde dies auch bedeuten, dass der Empfänger mit dem Chiffretext nichts anfangen kann, da er in diesem Fall keinen passenden Schlüsselstrom erzeugen kann. Das Wesen von echten Zufallszahlen ist ja, dass sie weder vorher bestimmbar noch rekonstruierbar sind. RC4 begnügt sich daher mit 99% Zufälligkeit und erzeugt einen pseudo-zufälligen Schlüsselstrom. Und dieser Schlüsselstrom kann von jedem, der den geheimen Chiffreschlüssel kennt, ebenfalls erzeugt werden.

Der Pseudocode zur Erzeugung des Schlüsselstroms ist äußerst kompakt und arbeitet sehr effizient. In Software ist RC4 gut zehnmal so schnell wie DES und immer noch schneller als AES. Die Schlüssellänge hat in RC4 absolut keinen Einfluss auf die Geschwindigkeit, sondern nur auf die Qualität des Schlüsselstroms und damit auf die Sicherheit.

Jedes Byte des Schlüsselstroms wird auf die folgende Weise durch die Funktion PRGA() erzeugt:

```
Function PRGA(): Byte
Extern S[i]: Byte[0..255]
i, j, k: Byte
{
    i := (i + 1) mod 256
```

```

j := (j + S[i]) mod 256
temp := S[i]
S[i] := S[j]
S[j] := temp
return ((S[i] + S[j]) mod 256)
}

```

Man erkennt, und das ist die Stärke von RC4, dass sich nach jedem erzeugten Byte des Schlüsselstroms die S-Box verändert. Diese Dynamik der S-Box ist verantwortlich für die Pseudozufälligkeit des Schlüsselstroms.

Die Transformation vom Klartext in den Chiffretext erfolgt durch eine einfache Exklusiv-oder-Vernüpfung eines Klartextbytes mit einem durch PRGA() erzeugten Byte so lange, bis der vollständige Klartext verschlüsselt ist.

```

RC4-Encrypt(Input: Byte): Byte
{
    return (input + PRGA()) mod 256
}

```

Die Entschlüsselung von RC4 ist exakt die gleiche Funktion. Durch die Kenntnis des Chiffreschlüssels kann der Empfänger den gleichen Schlüsselstrom erzeugen wie der Sender. Die ankommenden Bytes des Chiffretextes werden mit den Bytes der Funktion PRGA() einer Exklusiv-oder-Verknüpfung unterzogen, und dadurch, dass diese Funktion zu sich selbst invers ist, erhält man wieder den Schlüsselstrom.

```

RC4-Decrypt(Input: Byte): Byte
{
    return (input + PRGA()) mod 256
}

```

Der Pseudocode ist eng an die Programmiersprache C angelehnt, und mit dieser Beschreibung und Grundkenntnissen von C (oder einer anderen Programmiersprache) kann man den kompletten RC4-Algorithmus leicht in weniger als 30 Zeilen gut lesbaren Quellcode implementieren.

Allerdings gibt es einen kleinen Haken: Der Algorithmus ist schnell, klar strukturiert, bestechend einfach und sehr sicher – aber nur wenn er korrekt in ein Gesamtsystem eingebettet ist. Nehmen wir einmal die IP-Welt, in der es gilt, IP-Pakete zu verschlüsseln. Hier steht man vor der Aufgabe, eine Folge von IP-Paketen, nehmen wir an für den Zeitraum einer mehrstündigen Remote Access Session, zu ver- und entschlüsseln. Nehmen wir an, beide Seiten haben die Möglichkeit, sich für die Dauer der Session auf einen geheimen Sitzungsschlüssel zu einigen.

Wir haben gesehen, dass Sender und Empfänger in der Lage sind, damit den gleichen Schlüsselstrom zu erzeugen. Die Applikation des Senders erzeugt Folgen von IP-Paketen, die der Reihe nach RC4 byteweise verschlüsselt und zum Empfänger geschickt werden.

Die Pakete kommen in dieser Reihenfolge an und werden wieder entschlüsselt. Oder? Im Prinzip ja, aber nur, wenn die Pakete auch tatsächlich in der korrekten Reihenfolge ankommen, ist der Schlüsselstrom beim Empfänger synchron. Und das ist bei IP keineswegs garantiert. Im Gegenteil, IP garantiert weder dass sie überhaupt ankommen, noch dass sie in der Reihenfolge ankommen, in der sie abgeschickt wurden.

Die Lösungen, die man für RC4 in Kombination mit Layer-2- oder Layer-3-Paketen in der Praxis bisher angewendet hat, reichen von gut bis katastrophal. Eine Unsitte ist es, sich das Leben einfach zu machen und den Schlüsselstrom bei jedem Paket einfach neu zu starten. Dann ist er garantiert synchron. Sicherheit des Ganzen: gleich null! Denn die Sicherheit von RC4 basiert auf seinen 2^{1700} verschiedenen Zuständen und der Tatsache, dass praktisch ein zufälliger Schlüsselstrom erzeugt wird. Mit obiger Methode erzeugt man zimal den gleichen, also sich wiederholenden, extrem kurzen Schlüsselstrom! Und vor jedem Paket jedes Mal einen neuen Chiffreschlüssel zu verteilen, scheidet aus Machbarkeitsgründen ebenfalls aus.

Etwas, aber nur ganz wenig, entschärft wird das Verfahren, indem man den Schlüssel zweiteilt, in einen geheimen Teil, den eigentlichen Chiffrierschlüssel, und einen öffentlichen Teil, der im Klartext mit jedem verschlüsselten Paket oder Frame mitgeschickt wird. Dieser Teil wird in der Regel als Initialisierungsvektor (IV) bezeichnet. Zu Initialisierung des Schlüsselfeldes $K[i]$ wird dann die Kombination (meist eine einfache Verkettung) von Schlüssel und IV benutzt. Der IV wird auf Zufallsbasis vor jedem Paket neu erzeugt. Auch dieses Verfahren ist noch extrem unsicher, denn ab einer gewissen Anzahl übertragener Pakete kann ein erfolgreicher Ciphertext-only-Angriff auf den Schlüssel geführt werden.

Der beste Ansatz ist der, RC4 einfach aus diesem Bereich des TCP/IP- oder ISO/OSI-Schichtenmodells zu verbannen und dort anzusiedeln, wo es garantiert werden kann, dass die Daten komplett und geordnet ankommen. Also zum Beispiel in oder oberhalb von TCP. Hier kann man RC4 einsetzen, ohne es künstlich unsicher machen zu müssen. Dies wird zum Beispiel in SSL getan, hier kann sich RC4 voll entfalten.

„RC4 geknackt!!! WEP ist unsicher.“ Wieder einmal, als zum ersten Mal praktisch vorgeführt wurde, wie die WEP-Verschlüsselung (Wireline Equivalent Privacy) geknackt wurde indem es gelang, nach einer ausreichend mitgelesenen Menge von Ciphertext den WEP-Schlüssel, egal ob 40 oder 128 Bit, zu berechnen, wurde in diesem Atemzug behauptet, RC4 sei geknackt. Abgesehen davon, dass WEP gar nicht als ernst zu nehmende Verschlüsselung konzipiert war, sondern nur eine zu normaler Datenverkabelung vergleichbare Vertraulichkeit zum Ziel hatte, hatten sich die Angreifer gar nicht an RC4 versucht! Nein, sie nutzten einfach aus, dass sich die Entwickler von WEP offensichtlich nicht hinreichend mit RC4 auseinander gesetzt hatten und sich damit auch keine weiteren Gedanken über die Tatsache gemacht hatten, dass es in RC4, wie in vielen anderen Algorithmen auch, schwache Schlüssel geben kann. Und schwache Schlüssel gilt es eben zu vermeiden. Das Resultat war der hinlänglich bekannte WEP-Sicherheitsgau, denn die Schlüsselerzeugung in WEP produzierte so viele schwache Schlüssel, dass sogar Ciphertext-only-Angriffe mit ein paar Megabyte Chiffretext den geheimen Schlüsselteil berechnen konnten.

Ein anderer Fall ist die Kryptoanalyse von Microsofts PPTP durch Bruce Schneier. Auch hier haben gleich wieder ein paar Leute, die garantiert den Aufsatz von Bruce Schneier nicht richtig gelesen oder, wenn doch, nicht verstanden haben, behauptet, RC4 sei geknackt. Aber auch hier: Das Problem war nicht RC4, sondern alleine die Art, wie Microsoft die symmetrischen Schlüssel erzeugt hatte: aus dem auf LAN-Manager-Verhältnisse zurechtgestutzten Benutzerpasswort! So etwas darf man eigentlich nicht glauben! Auch hier wurde also RC4 gar nicht angegriffen, sondern die Erzeugung der Schlüssel in PPTP, die natürlich mit einfachen Wörterbuchangriffen geknackt werden konnte.

Allerdings gibt es in RC4 tatsächlich so genannte schwache Schlüssel. Im Initialisierungsprozess (KSA) gibt es zwei Schwächen, die hierfür verantwortlich sind. Bei bestimmten Schlüsseln reicht die Kenntnis einiger weniger Bits, um viele Zustands- und Ausgangsbits mit hoher Wahrscheinlichkeit zu bestimmen. Mit Kenntnis dieser Schlüssel sind bestimmte Angriffe nicht nur theoretisch möglich geworden.

Die zweite Schwäche tritt zutage, wenn der RC4-Schlüssel aus einem geheimen Teil, der lange Zeit gültig ist, und einem nicht geheimen Teil besteht, der sich laufend ändern kann. Hier kann ein Angreifer, wenn er genug Chiffretext besitzt, relativ einfach den geheimen Teil des Schlüssels berechnen. Leider ist dies eine verbreitete Betriebsart für RC4 (z.B. WEP).

RC4 ist leistungsfähig und, bei richtig implementierter Zuführung entsprechender Schlüssel, auch sehr sicher. Allerdings führt Letztes zu der richtigen Feststellung, dass RC4 in bestimmten Umgebungen, vor allem wenn er als Blockchiffrierverfahren missbraucht wird, etwas problematisch einzusetzen ist.

Der Geschwindigkeitsvorteil, vor allem gegenüber DES, ist mittlerweile auch kein Schlüsselkriterium mehr, wenn es einen Algorithmus auszuwählen gilt. Einerseits sind die Rechner seit 1987 sehr viel leistungsfähiger geworden, und andererseits bringt der AES eine relativ hohe Performance, auch bei 8-Bit-Systemen.

4.10 Asymmetrische Verschlüsselungsverfahren

In diesem Kapitel wurde bisher stillschweigend davon ausgegangen, dass die symmetrischen Schlüssel zum Chiffrieren und Dechiffrieren bei Sender und Empfänger bekannt sind. Wie sie dorthin gelangt sind, wurde nicht weiter beachtet. Dies kann jedoch ein sehr schwerwiegendes Problem sein, insbesondere dann, wenn sehr viele mögliche Sender und Empfänger existieren oder sich zwei Partner vorher gar nicht kennen. Wo liegt das Problem?

4.10.1 Die kurze Geschichte der Public-Key-Kryptografie

Nehmen wir einmal den Fall an, dass zwei Personen ihre Kommunikation, beispielsweise ihre Internet-E-Mail, verschlüsseln möchten. Da beide sehr sicherheitsbewusst sind, möchten sie das Triple-DES-Verfahren benutzen. In diesem Fall können sich beide persönlich treffen und den Schlüssel austauschen, oder sie können ihn sich per Post auf

Papier oder Datenträger zustellen. Aber in keinem Fall können sie den Austausch über das Internet vornehmen, denn dann könnte ein Unbefugter mitlesen! Selbst das Postverfahren ist nicht hundertprozentig sicher. Wenn beide zudem einige tausend Kilometer voneinander entfernt wohnen, ist ein persönliches Treffen unvorteilhaft, insbesondere dann, wenn es sich nicht nur um zwei Personen, sondern um die 100.000 Mitarbeiter eines Unternehmens handelt. Und was ist, wenn sich beide gar nicht vorher persönlich kennen? Oder wenn Sie, um ein praktisches Beispiel zu nehmen, sich an einem E-Commerce-Webserver anmelden und Ihre Kreditkarteninformationen verschlüsselt übertragen möchten? Wie einigen Sie und der Server sich auf einen symmetrischen Schlüssel?

Dieses Problem war für die Kryptologen in aller Welt eine harte Nuss. Den meisten erschien das Problem praktisch unlösbar, so dass sich in den Sechziger- und Siebzigerjahren fast niemand mehr ernsthaft mit der Thematik der Schlüsselverteilung auseinander setzte. Zum Glück gab es aber noch einige Unentwegte, die sich den Kopf darüber zerbrachen, wie dieses Problem gelöst werden konnte. Zwei dieser Forscher, die Mathematiker Whitfield Diffie und Martin Hellman, begannen im Jahr 1974 eine Zusammenarbeit, deren Resultat (sowie die daraus folgenden Entwicklungen) die Kryptologie revolutionieren sollte.

Ralph Merkle, eine weitere Kryptografie-Größe, stieß noch zu dem Team, und zwei Jahre später, im Jahr 1976, war es so weit. Martin Hellman hatte einen Ansatz gefunden, der auf dem Problem des diskreten Logarithmus (siehe unten) beruhte und als Diffie-Hellman-Merkle-Verfahren Geschichte machte. Bei der Vorstellung ihrer Lösung vor einem verblüfften Fachpublikum anlässlich einer Tagung der National Computer Conference 1976 staunten die Experten über die geniale Einfachheit und gleichzeitige Stärke des Verfahrens, und den ersten Teilnehmern dämmerte die wirkliche Bedeutung dessen, was sie gerade eben gehört hatten, für die weitere Entwicklung unserer Informationsgesellschaft.

Für Whitfield Diffie war das aber genau genommen nur ein halber Durchbruch. Das Verfahren eignete sich zwar hervorragend zum Schlüsseltausch, aber man konnte damit keine Daten verschlüsseln. Ihm schwebte die ganze Zeit über ein gänzlich neues Prinzip vor, bei dem man etwas mit einem speziellen Schlüssel verschlüsseln konnte, das nur mit einem dazugehörigen anderen Schlüssel wieder zu entschlüsseln war – ein asymmetrisches Verschlüsselungsverfahren. Der Schlüssel zum Verschlüsseln konnte bei diesem Konzept jedermann weltweit bekannt sein, denn er konnte ja nur zum Ver-, aber nicht mehr zum Entschlüsseln benutzt werden. Als Methode zur Realisierung schwebte Diffie eine besondere Einwegfunktion vor, zu der es eine so genannte Hintertür- oder Falltür-Funktion (Trap Door Function) gibt, die deren Einwegfunktionalität umgehen kann. Die Kenntnis dieser Hintertür ist die Basis für die Entschlüsselungsfunktion.

Whitfield Diffie stellte dieses Konzept 1975 der Öffentlichkeit vor, und tatsächlich begannen einige Forscher, allen voran natürlich Diffie, Hellman und Merkle, damit, die geeigneten Einwegfunktionen zu suchen – jedoch vorerst ohne Erfolg. Die Idee war super, nur eine passende Einwegfunktion nebst dazugehöriger Hintertür-Funktion war einfach nicht zu finden.

Erst im April 1977 und über 3.000 Meilen von der Wirkungsstätte Whitfield Diffies entfernt, an der Ostküste der USA, im Labor für Computerwissenschaften des Massachusetts Institute of Technology (MIT), fand ein anderes Forscher-Trio eine Lösung des Problems. Dr. Ronald Rivest hatte dort ein Jahr zuvor den Artikel von Diffie und Hellman über die asymmetrische Kryptografie gelesen und unmittelbar begriffen, was für ein Potenzial darin verborgen lag. Er konnte damals auch seine beiden Kollegen Adi Shamir und Leonard Adleman überzeugen, sich mit ihm zusammen auf die Suche nach der Einwegfunktion und ihrer Hintertür zu machen. Nachdem sie ein Jahr lang zunehmend frustrierter eine ganze Reihe unbrauchbarer Ansätze aussortiert hatten, kam Ron Rivest im April 1977 ein Verfahren in den Sinn, das heute als das RSA-Verfahren (siehe unten), benannt nach den Initialen der drei Forscher, weltbekannt und eines der am meisten eingesetzten so genannten Public-Key-Verfahren ist.

RSA machte sich ein anderes mathematisches Problem zunutze als das Diffie-Hellman-Verfahren, nämlich das so genannte Faktorisierungsproblem. Es ist nämlich sehr einfach und schnell möglich, zwei sehr große Primzahlen miteinander zu multiplizieren, aber es ist praktisch unmöglich, das Ergebnis wieder in die ursprünglichen Faktoren zu zerlegen. Die zur Entschlüsselung notwendige Hintertür des Verfahrens leitet sich aus der Kenntnis der beiden Primfaktoren ab.

Beide Verfahren, RSA und Diffie-Hellman, wurden in den USA kurz nach ihrer Veröffentlichung patentiert. Jawohl, Sie haben richtig gelesen, patentiert, denn obwohl es in den USA, wie auch in praktisch allen anderen Staaten, normalerweise nicht möglich ist, mathematische Algorithmen zu patentieren, wurden beide Verfahren geschützt. Jedoch sind beide Patente mittlerweile abgelaufen, und jedermann kann die Verfahren frei von irgendwelchen Gebühren einsetzen.

4.10.2 Das Grundprinzip der Public-Key-Kryptografie

Die Grundpfeiler der Public-Key-Kryptografie sind ungelöste oder so genannte schwierige mathematische Probleme. Besonders geeignet sind mathematische Funktionen, die sich in eine Richtung sehr einfach und schnell berechnen lassen, deren Umkehrung hingegen sehr schwierig und langwierig ist. Langwierig bedeutet hier Zeitspannen zwischen einigen tausend Jahren und dem Alter des Universums. Praktisch gesehen sind solche Einwegfunktionen also nicht umkehrbar. Nicht umkehrbar bedeutet aber auch, dass man etwas, was durch diese Funktionen verschlüsselt wurde, nicht mehr in sinnvoller Zeit entschlüsseln kann. Das ist richtig und bei speziellen Algorithmen, wie dem Diffie-Hellman-Verfahren zur Schlüsselerzeugung, auch essenziell. Andere Public-Key-Algorithmen, die sowohl ver- als auch entschlüsseln sollen, bedienen sich eines speziellen Tricks, der so genannten Falltür-Funktion (Trap Door Function). Diese Funktion kennt eine Art und Weise, wie sie mit einem speziellen Geheimnis, dem privaten Schlüssel, den Chiffretext wieder in den Klartext transformieren kann. Wie dieser Trick funktioniert, werden Sie nach dem folgenden kleinen Überblick über die mathematischen Grundlagen der Public-Key-Kryptografie am Beispiel des RSA-Verfahrens erfahren.

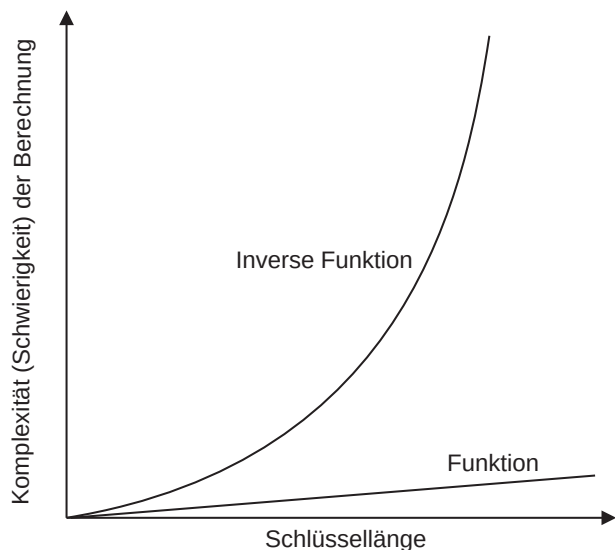


Abbildung 4.27: Die Public-Key-Kryptografie bedient sich Funktionen, deren Umkehrung in sinnvoller Zeit nicht möglich ist.

4.10.3 Mathematische Grundlagen

Dieser Abschnitt soll nur einen recht oberflächlichen, aber zum Verständnis notwendigen Überblick über die Mathematik geben, die in der Public-Key-Kryptografie benutzt wird. Wer sich tiefer in das Gebiet der Zahlentheorie einarbeiten möchte, dem sei die weiterführende Literatur hierzu empfohlen.

Primzahlen

Eine Primzahl ist eine Zahl, die nur durch 1 und sich selbst teilbar ist. Es gibt unendlich viele Primzahlen. Die einzig bekannte gerade Primzahl ist die 2, alle anderen sind ungerade. Jede gerade Zahl größer als 2 ist die Summe zweier Primzahlen. Das ist zwar mathematisch nicht bewiesen, sondern nur eine Vermutung, die so genannte Goldbach'sche Vermutung, trifft aber auf alle bisher untersuchten geraden Zahlen zu.

Die Häufigkeit des Vorkommens von Primzahlen nimmt mit wachsender Größe der Zahlen ab. Im Bereich von Zahlen bis 512 Bit Länge gibt es ca. 10^{151} Primzahlen. Im Vergleich dazu besteht unser Universum aus „nur“ ca. 10^{77} Atomen! Das vereinfacht die Erzeugung von Primzahlen auf Zufallsbasis enorm, denn die Wahrscheinlichkeit, dass zwei unabhängige Systeme zufällig die gleiche Primzahl auswählen, ist kleiner als die Wahrscheinlichkeit, sein Leben lang jeden Tag im Lotto zu gewinnen.

Aber wie erzeugt man auf Zufallsbasis eine große Primzahl? Natürlich sind nicht alle Primzahlen bekannt, und man kann auch vor allem nicht alle 10^{151} Primzahlen im Bereich von 512 Bit berechnen und diese in einem Primärspeicher speichern.

Primzahlerzeugung

Das Problem der Erzeugung von sehr großen Primzahlen lässt sich etwas entschärfen, wenn man sich mit etwas weniger als 100% zufrieden gibt, wenn es also ausreicht, wenn eine Zahl mit 99,999% Wahrscheinlichkeit eine Primzahl ist. Man generiert große, ungerade Zufallszahlen und testet diese mit probabilistisch arbeitenden Testverfahren, die mit einer bestimmten Wahrscheinlichkeit arbeiten. Eines der bekanntesten, schnellsten und am häufigsten implementierten Verfahren ist der Rabin-Miller-Algorithmus. Eine vollständige Rechnerimplementierung zur Primzahlerzeugung auf Basis dieses Algorithmus für 512-Bit-Zahlen arbeitet nach folgendem Algorithmus:

1. Erzeuge eine Zufallszahl p der Länge 512 Bit.
2. Setze das höchst- und das niederwertige Bit von p auf 1, um die Zahl hinreichend groß und garantiert ungerade zu machen.
3. Mache einen ersten Test, ob p durch die bekannten Primzahlen von 3 bis 2000 teilbar ist. Dadurch werden bereits über 95% aller ungeraden, nicht primen Zahlen eliminiert.
4. Falls keine Teilbarkeit besteht, wird der Rabin-Miller-Algorithmus für p sechsmal durchlaufen.

Wenn alle Tests erfolgreich waren, hat man mit hinreichender Wahrscheinlichkeit eine 512 Bit große Primzahl p gefunden. Aufgrund der hohen Wahrscheinlichkeit von 99,999% müssen alle Angreifer (oder etwas freundlicher formuliert: Kryptoanalytiker) in ihren Verfahren davon ausgehen, dass es sich tatsächlich um eine Primzahl handelt – wahrscheinlich ist es ja auch eine.

Modulo primitiv

Eine Zahl g ist modulo n primitiv, wenn es für jedes $b > 0 < n$ eine Zahl z gibt mit $g^z = b \pmod{n}$. Man nennt g auch Generator $\pmod{n}$. In unserem obigen Beispiel von Diffie-Hellmann „erzeugt“ man ein g , das modulo primitiv zu n ist, auf folgende Art und Weise: Man nimmt nacheinander Zahlen, sinnvollerweise fängt man mit sehr kleinen an, und probiert sie aus. Das dauert in der Regel nicht sehr lange, da Generatoren sehr häufig vorkommen.

Relativ prim

Zwei Zahlen sind relativ prim zueinander, wenn sie außer 1 keinen gemeinsamen Teiler besitzen. Die Zahlen 15 und 28 sind relativ prim zueinander, sie sind zwar keine Primzahlen, haben aber außer 1 keinen gemeinsamen Teiler. Aufgrund ihrer Natur ist eine Primzahl zu allen Zahlen, die kein Vielfaches ihr selbst sind, relativ prim.

Rechnen mit Resten (modulo)

In der Kryptografie spielt die modulare Arithmetik, also das Rechnen mit Resten, eine wichtige Rolle. Praktisch kein Public-Key-Verfahren kommt daran vorbei. Ein einfaches Beispiel für modulare Arithmetik ist unsere Armbanduhr, welche die aktuelle Tageszeit „Modulo 12“ anzeigt: $22 \bmod 12 = 10$, d. h., es wird 10 Uhr (22 dividiert durch 12 ist 1 Rest 10) angezeigt, falls es 22:00 Uhr ist. Aber es wird auch 10 Uhr angezeigt, wenn es tatsächlich 10 Uhr ist. Mathematisch ausgedrückt heißt dies: 22 und 10 sind äquivalent modulo 12.

Die Schreibweisen für die Äquivalenz sind verschieden, in der Fachliteratur findet man sowohl die explizite Form mit einem Gleichheitszeichen bestehend aus drei Strichen oder die implizite Form mit einem normalen Gleichheitszeichen. Ich werde mich im Folgenden letzterer Form bedienen.

Allgemein ausgedrückt gilt für alle $a \geq 0$ und $0 < b < n$:

$a = b \pmod{n}$, wenn es eine ganze Zahl k gibt, mit $a = b + kn$

Das Ergebnis von $a \pmod{n}$ nennt man das Residuum, die Operation selbst modulare Reduktion.

Die modulare Arithmetik verhält sich analog zur normalen Arithmetik, sie ist kommutativ, distributiv und assoziativ:

$$(a + b) \pmod{n} = ((a \pmod{n}) + (b \pmod{n})) \pmod{n}$$

$$(a * b) \pmod{n} = ((a \pmod{n}) * (b \pmod{n})) \pmod{n}$$

Speziell im Bereich der Berechnungen in Computern sollte man sich auch folgende Beziehung vergegenwärtigen:

$$a^8 \pmod{n} = (a * a * a * a * a * a * a * a) \pmod{n} = ((a^2 \pmod{n})^2 \pmod{n})^2 \pmod{n}$$

Was macht die modulare Arithmetik für die Kryptografie eigentlich so interessant? Schauen wir uns einmal kurz die Mathematik des weiter unten beschriebenen Diffie-Hellman-Verfahrens an. Hier wird ein Wert A durch die Funktion $A = g^x \pmod{n}$ erzeugt. Entsprechend große x und n vorausgesetzt, handelt es sich dabei um eine Einwegfunktion, denn man kann x nicht mehr aus A zurückberechnen, auch nicht bei Kenntnis von g und n .

Nehmen wir der Einfachheit halber einmal kleine Zahlen für ein kurzes Beispiel; g soll gleich 2 und n gleich 11 sein. Die Zahl n muss eine Primzahl sein. Als Vergleich dazu nehmen wir die Funktion $B = g^x$, also die gleiche Exponentiation, jedoch ohne modulare Reduktion. Diffie-Hellman erlaubt es, alle Werte außer der geheim zu haltenden Zahl x zu veröffentlichen. Somit haben wir zwei Funktionen, $A = 2^x \pmod{11}$ und $B = 2^x$.

Das Ziel eines Angriffs auf das Verfahren ist es, den geheimen Wert x zu ermitteln, alle anderen Werte sind ja bekannt. Das Verfahren, x bei der Funktion $B = g^x$ zu ermitteln, ist einfach. Nehmen wir einmal den Fall, dass $x = 6$ und damit $B = 64$ wäre. Man wählt eine Hilfszahl h und einen möglichen Wert dafür aus, zum Beispiel 4, und rechnet $2^4 = 16$. 16 ist kleiner als 64, also wird h höher gewählt, z.B. 8, und man rechnet $2^8 = 256$. Dieser Wert ist größer als 64, also muss der Wert x dazwischen liegen. Man nimmt einen Zwischenwert, und mit $h = 6$ ergibt sich $2^6 = 64$, und schon man hat den Wert x gefunden. Dies ist durch den stetigen Verlauf der Funktion $B = 2^x$ möglich, denn es gilt immer, dass bei einem $x > y$ auch $2^x > 2^y$ ist. Die maximale Zahl der Schritte dieses Algorithmus lässt sich genau festlegen, denn für eine n Bit lange Zahl benötigt man maximal n Schritte.

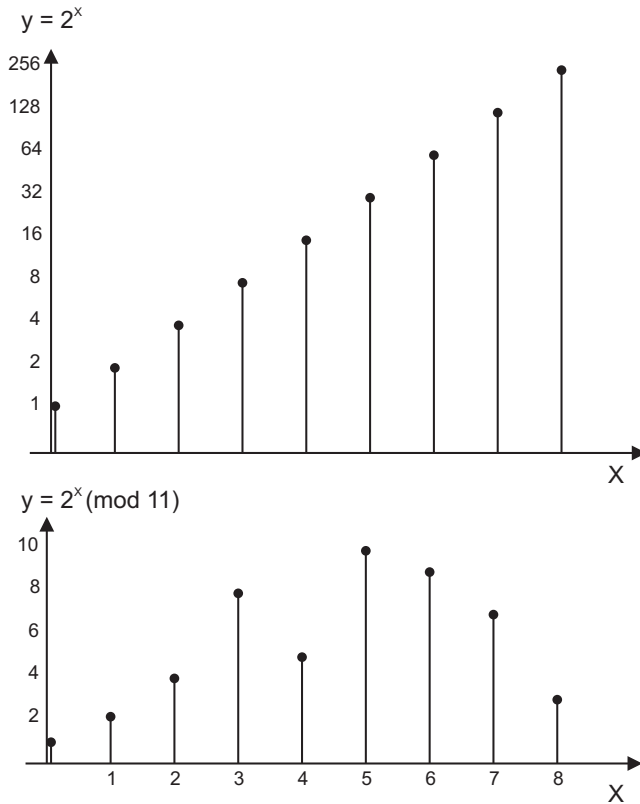


Abbildung 4.28: Die Graphen der Funktion $y=2^x$ und $y=2^x \pmod{11}$ im Vergleich

Mit der modularen Reduktion würde die Funktion Folgendes ergeben: $B = 2^6 \pmod{11} = 9$. Man würde auch hier $h = 4$ wählen und rechnet $2^4 \pmod{11} = 5$. 5 ist kleiner als 9, also wählt man x größer, z.B. gleich 8. $2^8 \pmod{11} = 3$. Das Ergebnis ist jedoch, trotz größerem h , kleiner geworden – was nun? Hier versagt unser einfacher Algorithmus schon, denn man weiß an diesem Punkt nicht mehr, ob man das nächste h jetzt größer oder kleiner wählen soll.

Wo liegt der fundamentale Unterschied beider Funktionen? Dies wird vielleicht am schnellsten klar, wenn man deren Graphen einmal vergleicht. In Abbildung 4.28 sehen Sie, dass der Verlauf der Funktion $g^x \pmod{n}$ sehr unstetig zwischen den Werten 0 und $n-1$ schwankt – ganz im Gegensatz zur stetig ansteigenden Funktion g^x .

Dies ist nur ein einfaches Beispiel, und auch das Beispiel mit der modularen Reduktion wäre noch kein Problem. In der Praxis verwendet man jedoch Zahlen ganz anderer Größenordnung. Das im IKE-Protokoll (siehe Kapitel 8) eingesetzte Diffie-Hellman-Verfahren fordert für n eine streng getestete Primzahl von mindestens 768 Bit Länge (üblich sind 1024 Bit oder mehr) und empfiehlt für x eine Zahl mit mindestens 160 Dezimalstellen mit $x < n$.

Die Euler'sche Totientenfunktion

Das Resultat der Euler'schen Totientenfunktion $j(n)$ ist die Anzahl von positiven ganzen Zahlen kleiner n , die relativ prim zu n sind. Falls die Zahl n eine Primzahl ist, sind alle Zahlen kleiner n relativ prim, also ist in diesem Fall $\varphi(n) = n - 1$.

Falls die Zahl keine Primzahl, jedoch das Produkt zweier Primzahlen p und q ist, dann gilt $\varphi(n) = \varphi(p) * \varphi(q) = (p - 1)(q - 1)$.

Der Satz von Euler, der auf dem Satz von Fermat beruht, besagt, dass für alle x , die relativ prim zu n sind, gilt: $x^{\varphi(n)} = 1 \bmod n$.

Inverse Elemente und Fermats Theorem

Multiplikativ inverse Elemente sind in normalen Zahlensystemen, wie den rationalen Zahlen, leicht zu finden. Das inverse Element zu a ist die Zahl x , die mit a multipliziert 1 ergibt, also $a * x = 1$ oder $x = 1/a$ oder $x = a^{-1}$. Es existiert zu jedem Element ein inverses Element.

Bezüglich der modularen Arithmetik sieht dies etwas anders aus, es kann sogar so sein, dass zu bestimmten Zahlen überhaupt kein inverses Element existiert. Fermats Theorem besagt, dass für alle Primzahlen p und alle Zahlen $a < p$ gilt: $a^p \bmod p = a$ oder $a^{p-1} \bmod p = 1$.

Wenn wir nun das inverse Element x suchen mit $ax \bmod p = 1$, dann kann man, wenn p eine Primzahl ist und $a < p$ ist, Fermats Theorem benutzen. Denn es gilt: $a^{p-1} \bmod p = 1$ und $a^x \bmod p = 1$ zur Bestimmung des inversen Elements. Diese beiden Gleichungen kann man kombinieren zu:

$$\begin{aligned} a^x \bmod p &= a^{p-1} \bmod p \\ \text{und } x &= a^{p-2} \bmod p \end{aligned}$$

4.11 Das Diffie-Hellman-Verfahren

Das Diffie-Hellman-Verfahren ist der Urvater der Public-Key-Kryptografie. Es kann nicht zum Ver- und Entschlüsseln von Daten benutzt werden, sondern nur zur Erzeugung von symmetrischen Schlüsseln. Es basiert auf dem Problem, dass eine Exponentiation mit wenig Aufwand durchführbar ist, die Umkehrung davon, das Auffinden des Logarithmus, jedoch sehr viel länger dauert. Ein schwieriges mathematisches Problem wird daraus, wenn es sich dabei auch noch um den diskreten Logarithmus handelt. Die Funktion $z = g^x \bmod n$ ist schnell zu berechnen, die Rückberechnung von x aus z ist jedoch sehr langwierig. Wieso ist x aus $g^x \bmod n$ sehr viel schwerer zu bestimmen als aus g^x ?

Den gesuchten Wert von x bestimmt man für die Funktion g^x mit Hilfe einer schrittweisen Approximation aus dem Wert z , indem man eine Hilfszahl p mit einem Startwert annimmt, g^p berechnet und mit z vergleicht. Ist z größer, probiert man ein größeres, im anderen Fall ein kleineres p aus und nähert sich auf diese Weise dem gesuchten Wert. Dieser ist gefunden, wenn $g^p = z$ ist, und man benötigt maximal so viele Schritte, wie die Zahl x an Bit-Stellen aufweist.

Im Fall der Funktion $g^x \bmod n$ ist die Sache so nicht zu machen, denn aufgrund der sehr unsteten Funktion kann man nicht sagen, ob man, falls $g^p \bmod n$ größer als z ist, p vergrößern oder verkleinern muss – beides ist gleich wahrscheinlich.

Diese Schwierigkeit macht man sich beim Diffie-Hellman-Verfahren zunutze, denn selbst bei bekanntem z , n und g ist x nicht in sinnvoller Zeit errechenbar. Einen Schlüsseltausch kann man in folgender Weise realisieren:

Die Gegenstellen A und B möchten sich auf einen symmetrischen Schlüssel einigen. Jeder erzeugt für sich zu diesem Zweck eine sehr große Zufallszahl und hält sie geheim. A benutzt zwei Hilfszahlen, die Zahl g und die Zahl n , eine sehr große Primzahl. In der Praxis, zum Beispiel im IKE-Protokoll, sind g und n im Standard festgelegt, somit allgemein bekannt und brauchen nicht mehr explizit übertragen zu werden.

1. A berechnet $a = g^x \bmod n$ und schickt a (sowie gegebenenfalls g und n) zu B.
2. B berechnet $b = g^y \bmod n$ und schickt b zu A.
3. A berechnet $k = b^x = (g^y \bmod n)^x = g^{yx} \bmod n$.
4. B berechnet $k' = a^y = (g^x \bmod n)^y = g^{xy} \bmod n$.
5. Da $g^{yx} \bmod n = g^{xy} \bmod n$, ist auch $k = k'$, und beide Seiten haben somit einen identischen Schlüssel.

Jemand, der die Übertragung von a , b , g und n abgehört hat oder diese Werte bereits kennt, kann trotzdem ohne die Kenntnis wenigstens eines der beiden geheimen Werte x und y den Schlüssel nicht berechnen. Um das Diffie-Hellman-Verfahren sicher zu machen, müssen diese Zahlen gewissen Bedingungen genügen. Die Zahl n muss eine sehr große, zuverlässig getestete Primzahl sein. Üblich sind Längen von 768 oder 1024 Bit; manchmal verwendet man auch noch größere Zahlen. Für g nimmt man meist den Wert 2. Die geheimen Werte x und y sollen auch sehr groß sein, müssen aber stets kleiner als n sein. Sie sollten durch Zufallsgeneratoren oder sehr gute Pseudo-Zufallsgeneratoren erzeugt und nach der Berechnung der symmetrischen Schlüssel vernichtet werden. Die Qualität und Vertraulichkeit von x und y , den privaten Diffie-Hellman-Werten, ist absolut kritisch für den gesamten Schlüsseltausch.

In praktischen Implementierungen des Diffie-Hellman-Verfahrens, zum Beispiel dem IKE-Protokoll, sind die Werte n und g in einem Standard festgelegt und veröffentlicht. Sie müssen aus diesem Grund nicht mehr explizit beim Schlüsseltausch übertragen oder gar von einer Seite vorher noch berechnet werden.

4.11.1 Die Kryptoanalyse des Diffie-Hellman-Verfahrens

Das Diffie-Hellman-Verfahren ist nicht sicher vor Man-in-the-Middle-Angriffen. Das ist nicht weiter schlimm, weil dies von Anfang an bekannt war und deshalb entsprechende Maßnahmen zur Authentifizierung des Schlüsselaustausches implementiert werden können.

Bislang sind noch keine Angriffe gegen Diffie-Hellman derart erfolgt, dass man das Verfahren als gebrochen bezeichnen könnte. Allerdings sollte man ausreichende Schlüssellängen einsetzen. Bei der Erzeugung von Schlüsseln für 128-Bit-Verfahren sollte der Diffie-Hellman-Schlüssel mindestens 1024 Bit lang sein. Näheres hierzu finden Sie in Kapitel 6 (IKE).

4.12 Das RSA-Verfahren

Etwa ein Jahr, nachdem Whitfield Diffie und Martin Hellman durch ihre Arbeit frischen Wind, um nicht zu sagen Sturm, in die Kryptologie-Szene gebracht hatten, gingen drei Forscher an der Ostküste der USA mit einer weiteren bahnbrechenden Arbeit an die Öffentlichkeit. Die Professoren Ronald Rivest, Adi Shamir und Leonard Adleman vom Massachusetts Institute of Technology stellten das nach ihnen benannte RSA-Verfahren vor. Der wesentliche Unterschied zum Diffie-Hellman-Verfahren ist der, dass man mit RSA Daten ver- und entschlüsseln konnte. Denn die drei Wissenschaftler hatten das gefunden, was Whitfield Diffie und Martin Hellman bereits ein Jahr zuvor postuliert hatten: eine Einwegfunktion zur Verschlüsselung und die dazugehörige Falltür-Funktion zur Entschlüsselung.

Die Funktion von RSA beruht auf einem anderen mathematischen Problem als das Diffie-Hellman-Verfahren, nämlich auf der Schwierigkeit, eine große Zahl in ihre Primfaktoren zu zerlegen. Denn es ist einfach und schnell durchführbar, aus den beiden großen Primzahlen p und q die Zahl $n = pq$ zu berechnen. Die Umkehrung, also nur vom Wert n ausgehend die zwei Primzahlen zu finden, die miteinander multipliziert n ergeben, ist ein hartes mathematisches Problem. Wie funktioniert nun das RSA-Verfahren genau?

Das Tupel $\{n, e\}$ bildet den öffentlichen Schlüssel und das Tupel $\{n, d\}$ den privaten Schlüssel, von dem offensichtlich d nur der tatsächlich geheime, private Teil ist, da n auch Teil des öffentlichen Schlüssels ist. Die Transformation des Klartexts M in den Chifretext C erfolgt mit dem öffentlichen Schlüssel $\{n, e\}$ durch:

$$C = M^e \bmod n.$$

Die Entschlüsselung erfolgt mit dem privaten Schlüssel $\{n, d\}$:

$$M = C^d \bmod n.$$

Man kann beide Funktionen auch als $M = (M^e)^d \bmod n$ zusammenfassen. Das Schlüssel-paar $\{n, e\}$ und $\{n, d\}$ wird wie folgt berechnet:

Die Zahl n ist das Produkt zweier großer, möglichst vergleichbar langer Primzahlen p und q :

$$n = pq.$$

Anschließend wählt man e derart, dass es relativ prim zu $(p-1)(q-1)$ ist. Hier kann man sich ohne lange Berechnungen und Tests geeignete Zahlen aussuchen, indem man Primzahlen wählt, die ungleich p und q sind, was für kleine Primzahlen der Fall ist. Für besonders schnelle Berechnungen kann man sehr kleine e wählen, zum Beispiel 3, 17 oder $2^{16} + 1$ (65537). Diese Primzahlen wurden mit Bedacht gewählt, denn sie sind hervorragend geeignet, um mit ihnen binäre Exponentiationen durchzuführen, da sie, aus binärer Schreibweise gesehen, nur aus zwei Einsen und sonst nur Nullen bestehen. Die binäre Schreibweise der Primzahl 65537 ist zum Beispiel 1000000000000001. Allerdings sind sehr kleine Exponenten nicht so extrem sicher.

Der private Schlüsselteil d , der relativ prim zu n sein muss, wird über den erweiterten Euklid'schen Algorithmus bestimmt:

$$ed = 1 \bmod ((p-1)(q-1))$$

Nach der Berechnung von d müssen p und q zuverlässig vernichtet werden. Wenn diese beiden Primzahlen nicht mehr bekannt sind, kann man auch d nicht mehr bestimmen. Man müsste aus der öffentlichen Zahl n ihre beiden Primfaktoren p und q zurückberechnen – was aber ein hartes mathematisches Problem darstellt.

Die mathematischen Zusammenhänge sind beim RSA-Verfahren etwas komplexer als beim Diffie-Hellman-Algorithmus. Die folgende Beschreibung geht nicht sehr ins Detail. Wer tiefer in die Materie einsteigen möchte, dem seien weiterführende Bücher zur Zahlentheorie und Kryptografie empfohlen.

1. Wir haben gesehen, dass folgender Zusammenhang gilt: $C = M^e \bmod n$, $M = C^d \bmod n$ und $(M^e)^d = M \bmod n$. Da $(M^e)^d = M^{ed}$ ist, zerlegt man den Exponenten gewissermaßen in einen öffentlichen Teil e und einen privaten Teil d .
2. Die Zahl n wurde erzeugt, indem man zwei große Primzahlen p und q miteinander multipliziert hat. Da p und q prim sind, gilt für sie die Euler'sche Totientenfunktion $\varphi(n) = (p-1)(q-1)$.
3. Da e und d bezüglich $\bmod \varphi(n)$ invers sein müssen, damit der obige Zusammenhang gültig ist, gilt $ed = 1 \bmod \varphi(n)$, und man kann ein k bestimmen mit $ed = k\varphi(n) + 1$.
4. Nach dem Theorem von Fermat gilt $M^{p-1} = 1 \bmod p$. Da $(p-1)$ einer der beiden Faktoren von $\varphi(n)$ ist, kann man auch $M^{k\varphi(n)} = 1 \bmod p$ schreiben. Wenn wir beide Seiten der Gleichung mit M multiplizieren, erhalten wir $M^{k\varphi(n)+1} = M \bmod p$.
5. Das Gleiche kann man auch für die zweite Primzahl q durchführen und erhält analog dazu $M^{k\varphi(n)+1} = M \bmod q$.
6. Nach (1) gilt $(M^e)^d = M^{ed} = M \bmod n$.
7. Nach (3) gilt weiterhin $ed = k\varphi(n) + 1$. Somit gilt $M^{ed} = M^{k\varphi(n)+1} = M \bmod n$.
8. Nach (4) und (5) gelten $M^{k\varphi(n)+1} = M \bmod p$ und $M^{k\varphi(n)+1} = M \bmod q$. Also gilt auch $M \bmod p = M \bmod q = M^{k\varphi(n)+1} = M \bmod n$.
9. Da wir gesehen haben, dass $ed = k\varphi(n) + 1$ ist, ist auch $M^{ed} = (M^e)^d = M \bmod n$.

Zuerst die gute Nachricht: RSA ist unter richtigen Einsatzbedingungen und mit einer ausreichenden Schlüssellänge sicher. Die schlechte Nachricht: Es weiß niemand, wie lange noch.

Es gibt ein paar mittlerweile bekannte Schwächen bestimmter RSA-Implementierungen, die man umgehen kann. Bei sachkundigem und korrektem Einsatz sind noch keine praktikablen Angriffe gegen RSA bekannt geworden, die einfacher sind als eine Faktorisierung von n . Die ganzen bisher bekannten Angriffe waren gegen (manchmal nicht optimale) Implementierungen von RSA gerichtet und nicht gegen den RSA-Algorithmus selbst. Falls man alle Richtlinien beachtet und entsprechende Programmier-Toolkits, zum Beispiel das von RSA Security, einsetzt, kann nach heutigen Erkenntnissen – beim Einsatz eines hinreichend großen n – nichts schief gehen.

Wichtig zu wissen ist, dass RSA ein monoalphabetischer Algorithmus ist, der mit einem gegebenen Schlüssel aus gleichem Klartext immer gleichen Chiffretext erzeugt und umgekehrt. Also sind, wenn RSA als Datenverschlüsselalgorithmus benutzt wird, was allerdings selten vorkommt, entsprechende Betriebsarten wie z.B. CBC einzusetzen. Bei der Schlüsselübertragung in hybriden Kryptosystemen durch RSA ist dessen Monoalphabetismus kein Problem, da der Klartext zufällig ist und selbst eine (unwahrscheinliche) Wiederholung eines Wertes keine statistischen Angriffe erlaubt.

Einer der möglichen Knackpunkte von RSA ist der, dass man annimmt, den privaten Schlüssel d nur über eine Faktorisierung berechnen zu können, wenn man die beiden ursprünglichen Primzahlen p und q nicht mehr kennt. Das ist allerdings eine überhaupt nicht bewiesene Annahme. Weiterhin nimmt man an, dass die Faktorisierung von N ein hartes mathematisches Problem ist. Aber auch das ist nicht schlüssig bewiesen. Es könnte also bei Entdeckung eines genialen Algorithmus oder bisher nicht erkannter Zusammenhänge möglich sein, d über ein anderes Verfahren als die Faktorisierung zu berechnen. Oder die Faktorisierung selbst könnte so optimiert werden, dass sie in relativ kurzer Zeit zu einem Ergebnis kommt. Das ist bisher nicht geschehen, kann aber.

Ein anderer, allerdings berechenbarer Punkt ist die Faktorisierung mittels der bekannten Methoden, die bisher innerhalb gewisser Grenzen in einem direkten Zusammenhang zur aufgewendeten Rechenleistung stehen. Stünde plötzlich eine Technologie mit um Dimensionen höherer Rechenleistung zur Verfügung, könnten auch längere RSA-Schlüssel in vernünftiger Zeit faktorisiert werden. Solch eine Technologie dürften zum Beispiel entsprechende Quantencomputer sein. Heute sind sie allerdings noch Science-Fiction, aber das wird vermutlich nicht ewig so bleiben.

Aufgeschreckt wurde die Kryptologengemeinde im Jahr 2003, als Adi Shamir, das S von RSA, zusammen mit seinem Kollegen Eran Tromer seine Einschätzung publizierte, dass man mit einer speziellen, hochparallelen Hardware für etwa 10 Mio. Euro einen 1024-Bit-RSA-Schlüssel in einem Jahr berechnen kann. 10 Mio. Euro sind für einen Privatmann viel, aber nur die Hälfte des Jahresbudgets der NSA alleine für ihre Stromrechnung! Trotz der Tatsache, dass diese Hardware (vermutlich) so noch nicht existiert und die Vorhersagen noch nicht verifiziert worden sind, sollte man sich langsam von den allseits verbreiteten, in vielen Systemen und Implementierungen sogar fest kodierten 1024-Bit-RSA-Schlüsseln verabschieden und wenigstens 2048 Bit verwenden.

Echte RSA-Probleme gibt es mit zu kleinen Exponenten e , insbesondere mit dem Wert 3 lässt sich zwar schnell rechnen, aber eine hohe Sicherheit ist damit nicht mehr gegeben. Abhilfe kann hier nur Padding schaffen oder ein hinreichend großes e , was den Algorithmus allerdings verlangsamt.

Weiterhin gibt es verschiedene Angriffe gegen RSA-Implementierungen, die darauf beruhen, dass die Zeit zum Verschlüsseln in einem bestimmten Verhältnis zur Länge des privaten Schlüssels ist. Falls man in der Lage ist, diese Geschwindigkeit zu bestimmen, kann man den Schlüsselraum abschätzen und die Suche auf diesen Bereich konzentrieren.

Insbesondere SmartCards, auf die ein Angreifer vorübergehend Zugriff hat, können solche Zeitmessungen erlauben. Abhilfe schafft hier eine künstliche Verlängerung der Rechenzeit durch die RSA-Implementierung. Aber SmartCards haben in den Händen von Angreifern trotzdem kein leichtes Leben: Sie werden mit Über- oder Unterspannung

versorgt, über- oder unterkühlt, extrem erhitzt oder abgekühlt und auf alle möglichen anderen Arten malträtet. Ziel dabei ist es, die Hardware in einem Grenzbereich zu betreiben, in dem sie nicht mehr ganz korrekt funktioniert und man Rückschlüsse auf die „Innereien“ ziehen kann. Es ist schon beeindruckend, wenn einem ein Kryptoanalytiker in seinem Labor zeigt, wie man an der Kurve der Stromaufnahme einer SmartCard die 16 DES-Runden erkennen kann! Hier sind die Hardware-Hersteller gefordert, entsprechende Maßnahmen zu ergreifen!

RSA ist heute sicher, wenn es richtig implementiert ist und man ausreichend lange Schlüssel, mindestens 2048 Bit, verwendet. Falls ein Schutz sehr lange (zehn Jahre oder länger) dauern soll, was für VPNs in der Regel nicht zutrifft, sollte man wenigstens 4096 Bit einsetzen.

ECC ist eine Public-Key-Technologie, die in letzter Zeit vermehrt in VPN, insbesondere im IKE-Protokoll in Verbindung mit dem Diffie-Hellman-Schlüsseltausch eingesetzt wird. ECC ist sehr universell und kann sowohl zur Schlüsselerzeugung, zum Schlüsseltausch oder für digitale Signaturen verwendet werden. Uns interessiert hier aber hauptsächlich die Verwendung zur Erzeugung von symmetrischen Sitzungsschlüsseln.

Der Hauptvorteil von ECC ist, dass es, je nach Implementierung, etwa vier- bis zehnmal schneller sein kann als RSA oder Diffie-Hellman auf Basis modularer Exponentierung. Das liegt unter anderem daran, dass ECC mit vergleichsweise sehr kurzen Schlüsseln, meist weniger als 300 Bit, eine extrem hohe Sicherheit bieten kann – bei entsprechend geringerem Rechenaufwand.

Auch hier ist wieder ein hartes mathematisches Problem für die Sicherheit von ECC verantwortlich, nämlich der diskrete Logarithmus in einer geeigneten Menge oder einem geeigneten Zahlenkörper. Damit existiert eine gewisse Verwandtschaft zum Diffie-Hellman-Algorithmus, und ECC kann in diesem auch hervorragend eingesetzt werden.

Die Basis von ECC bildet die Tatsache, dass man auf einer elliptischen Kurve eine abelsche Gruppe abbilden kann und darin das ECDLP (ECDLP, Elliptic Curve Discrete Logarithm Problem) sehr schwer zu lösen ist, man in der Praxis also von einer Einwegfunktion sprechen kann. Der benutzte Zahlenkörper spielt in ECC eine wesentliche Rolle, üblich sind Galois-Felder über eine große Primzahl p , $\text{GF}(p)$, oder Galois-Felder über einen Körper mit n Elementen, $\text{GF}(2^n)$.

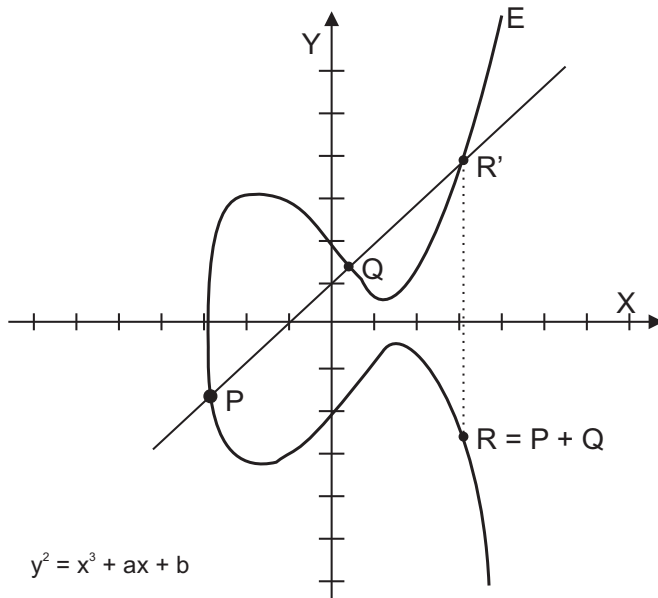


Abbildung 4.29: Die grafische Addition der Punkte P und Q auf der elliptischen Kurve E

Im Bereich reeller Zahlen kann man die Verknüpfung recht anschaulich grafisch (Abbildung 4.29) darstellen. Eine Addition, Schreibweise „+“, zweier Punkte P und Q definiert sich wie folgt:

1. Man zeichnet eine Gerade durch P und Q.
2. Diese Gerade schneidet die Kurve E im Punkt R*.
3. Der Punkt R* wird an der x-Achse gespiegelt, und man erhält den Punkt R.
4. Der Punkt R ist als Ergebnis der Addition definiert, es gilt $R = P + Q$.

Ein Sonderfall liegt vor, wenn zum Beispiel P die Spiegelung von Q an der x-Achse ist. Dann würde die Gerade durch P und Q keinen weiteren Schnittpunkt mit der Kurve haben. Damit wäre die Addition dieser beiden Punkte nicht möglich und die Bedingung für die Gruppe nicht mehr erfüllt. Also definiert man einen speziellen Punkt O, der auch gleichzeitig das neutrale Element bildet. Dieser Punkt O bildet den fehlenden Schnittpunkt. Daraus ergibt sich, dass die Spiegelung eines Punktes an der x-Achse dessen inverses Element bezüglich der Addition darstellt. Es gilt $P + P^* = P + (-P) = 0$. Mit O als neutralem Element bezüglich der Addition gilt auch $P + O = 0$. „Grafisch gesehen“ liegt der Punkt O in unendlicher Entfernung entlang der y-Achse.

Die Addition eines Punktes mit sich selbst erfolgt, indem eine Tangente an den Punkt angelegt wird, welche die Kurve dann in nur einem weiteren Punkt schneidet. Ist die Tangente senkrecht, dann tritt wieder der Sonderfall ein, dass der Punkt O als zweiter Schnittpunkt genommen wird.

Eine Multiplikation ist als mehrfache Addition definiert. Der Multiplikator gibt an, wie oft der Punkt P zu sich selbst addiert wird. Es gilt z.B. $3 * P = P + P + P$.

Die tatsächlichen mathematischen Berechnungen erfolgen aufgrund der verwendeten Gleichung mit den Koordinaten der Punkte.

Diese Zusammenhänge bleiben auch dann gültig, wenn die elliptische Kurve nicht mehr über den reellen Zahlen, wie in unserem obigen Beispiel, sondern über einen diskreten Zahlenkörper, zum Beispiel $GF(p)$, definiert wird. Damit verlassen wir allerdings die schöne durchgehende Kurve und bekommen ein Feld von Punkten als grafische Darstellung. Die Additionsoption erscheint aus dieser Sicht wie ein absolut zufälliges Springen von Punkt zu Punkt, das mit der einfachen grafischen Addition wie im obigen Beispiel nicht mehr darstellbar ist. Der Effekt ist im Prinzip der gleiche, wie er in der Abbildung 4.28 zum Problem des diskreten Logarithmus veranschaulicht ist. Eine vergleichbare Komplexität bzw. Schwierigkeit bei der Lösung des Logarithmusproblems erreicht man hier aber schon bei viel kleineren Schlüsselwerten.

Die Verwendung der ECC zum Erzeugen von symmetrischen Sitzungsschlüsseln auf Basis des Diffie-Hellman-Algorithmus sieht wie folgt aus:

Vorbedingung: Beide Seiten, A und B, müssen sich auf bestimmte Faktoren im Voraus geeinigt haben. Dies sind die benutzte Kurve und der benutzte Zahlenkörper sowie ein fester Punkt P auf der elliptischen Kurve. Die Kurve wird in der Regel so bestimmt, dass auf ihr eine Punktgruppe hoher Primzahlordnung n existiert.

1. A wählt eine streng geheim zu haltende Zufallszahl x aus mit $1 < x < n$.
2. B wählt eine streng geheim zu haltende Zufallszahl y aus mit $1 < y < n$.
3. A berechnet seinen öffentlichen Schlüssel Q_A mit $Q_A = xP$.
4. A berechnet seinen öffentlichen Schlüssel Q_B mit $Q_B = yP$.
5. Die beiden öffentlichen Schlüssel werden ausgetauscht.
6. A berechnet nun $R_A = xQ_B$ und B berechnet $R_B = yQ_A$.
7. Da $R_A = xQ_B = xyP = yxP = yQ_A = R_B$ gilt, haben beide Seiten den identischen Wert $R = R_A = R_B$ berechnet..

Ein potenzieller Angreifer, selbst wenn er alle Parameter außer x und y kennt, steht vor dem Problem, R aus P , Q_A und Q_B berechnen zu müssen. Ohne Kenntnis wenigstens eines der beiden Geheimnisse x oder y ist dies in vernünftiger Zeit nicht möglich. Das ECDLP (Elliptic Curve Discrete Logarithm Problem) wird für signifikant schwerer als das Problem des diskreten Logarithmus (DLP, Discrete Logarithm Problem) gehalten. Dadurch kann man bei vergleichbarer Sicherheit wesentlich kürzere Schlüssel benutzen. So ist ein Diffie-Hellman-Schlüssel der Länge 1024 Bit gleich sicher wie ein ECDH-Schlüssel mit 163 Bit Länge, und ein ECDH-Schlüssel mit 283 Bit entspricht einem DH-Schlüssel von 3072 Bit Länge!

Es sind noch keine erfolgreichen Angriffe gegen gut entworfene ECC-Implementierungen publiziert worden. Jedoch ist die Mathematik hinter ECC komplexer als die mit dem normalen DLP verbundene, und dadurch ist eine korrekte Wahl der Parameter und Eigenschaften der verwendeten Kurve und des Zahlenkörpers nicht trivial.

So wurden denn auch im ursprünglichen IKE-Standard, dem RFC2409, zwei ECC-Gruppen im Körper $GF(2^n)$ definiert, die Gruppen 3 und 4, die auf den Körpern $GF(2^{155})$ und $GF(2^{185})$ beruhen. Die Zahl n (155 und 185) ist eine natürliche Zahl, die nicht prim ist. Forschungen, die 2001 von der Waterloo Universität in Kanada von M. Jacobsen, A. Menezes und A. Stein mit Unterstützung der Firma Certicom, einem ausgewiesenen ECC-Spezialisten, veröffentlicht wurden, zeigten erfolgreiche Lösungen des ECDLP über den Körpern $GF(2^{62})$, $GF(2^{93})$ und $GF(2^{124})$. Es wurde ebenfalls postuliert, dass auch der Körper $GF(2^{155})$, der bislang jedem anderen Angriff widerstehen konnte, ebenfalls anfällig für den beschriebenen Angriff (weil Descent Attack) ist. Glücklicherweise wurde in der Arbeit ebenfalls festgestellt, dass Körper, die für die Zahl n Primzahlen verwenden, diesem Angriff nicht ausgesetzt sind. Dies gilt sowohl für $GF(n)$ als auch für $GF(2^n)$.

So wurden denn auch Erweiterungen für IKE veröffentlicht, die neue Gruppen, unter anderem die Gruppen 7 und 8, enthalten, die gegen wirklich alle bisher bekannten Schwächen gewappnet sind. Die Körper dieser Gruppen sind $GF(2^{163})$ und $GF(2^{283})$, die, zusammen mit anderen, nach aktuellen Sicherheitsaspekten ausgewählten, Parametern ein hohes Maß an Schutz bieten. Man darf nicht vergessen, dass diese Algorithmen an den sensibelsten Punkten der Sicherheitsprotokolle arbeiten, nämlich dort, wo die Sitzungsschlüssel erzeugt werden.

Das Thema Zufallszahlen ist Gegenstand einer Reihe von Büchern und auch von RFCs der Internet Engineering Task Force (IETF). Gemeinhin ist es nicht schwer, eine zufällige Zahl zu erzeugen, man kann zum Beispiel würfeln oder Roulette spielen. Es ist aber weitaus schwieriger, auf einem Computer eine echte oder der sehr nahe kommende Zufallszahl ausreichender Größe zu erzeugen. Das liegt daran, dass ein Computer aufgrund seiner Programmierung nur vorherbestimmte Dinge tut, auch wenn einem das angesichts des Verhaltens moderner PC-Betriebssysteme manchmal recht unglaublich erscheint. Aber es ist eine Tatsache, dass Pseudo-Zufallszahlen – so nennt man solche Größen, die durch Computerprogramme erzeugt wurden – sehr schwach sind und oft mit einem überschaubaren Aufwand rekonstruiert werden können.

Solche Programmfunktionen – fast jede Programmiersprache oder Programmierumgebung hat eine RND-Funktion (Random Function) – basieren auf einem sich wiederholenden Muster, das auf einem jedes Mal anderen Startwert basiert. Kennt man diesen Startwert oder kann man ihn ungefähr abschätzen, so kann man auch die Zufallszahl(en) mit einigem Aufwand rekonstruieren.

Leider ist die Systemzeit eine Größe, die praktisch auf jeder Plattform zur Verfügung steht und sich laufend ändert. Leider deshalb, weil sie in vielen Software-PRNGs (Pseudo Random Number Generator) als eine Eingangsgröße eingeht. Das Problem ist, dass man oft ziemlich genau ermitteln kann, wann eine Zufallszahl erzeugt wurde, und damit zumindest den Bereich der möglichen Zeiten abschätzen kann. Ist die Zeit jedoch nur einer von mehreren, voneinander unabhängigen, veränderlichen Faktoren, so kann man für gewisse Anwendungen auch mit Software-PRNG leben.

Aber auch in einem Computer gibt es Dinge, die nicht so einfach rekonstruierbar sind und eine starke Zufallskomponente haben. Die mittlere Zugriffszeit von Festplattenlaufwerken, der Inhalt des Arbeitsspeichers und vieles mehr kann schon eine gute Grundlage bilden. Nimmt man von verschiedenen derartigen Erzeugern gelieferte Werte, fügt sie zusammen und benutzt sie als Eingangsvariablen für verkettete Hash-Funktionen oder einen sehr guten PRNG, so ist das Ergebnis schon eine sehr gute Pseudo-Zufallszahl. Da man in der Regel nicht alle paar Millisekunden neue Zufallszahlen benötigt, ist der Aufwand einer solchen Methode durchaus vertretbar. Zudem werden in Systemen, die viele Zufallszahlen benötigen, oft im Hintergrund solche Berechnungen durchgeführt und die Zahlen zur späteren Verwendung gespeichert.

Allerdings ist es oft wichtig, dass die Zufallszahlen relativ gleichmäßig über einen gewissen Bereich verteilt sind. Schlecht wäre es zum Beispiel, wenn sich eine volle Festplatte immer im Rahmen sehr langer Zeiten bewegen würde und damit eine Häufung bestimmter Eingangswerte für den PRNG stattfände.

Ein weiteres Problem, das den Grad der Zufälligkeit verringern kann, kann entstehen, wenn man Zufallszahlen in bestimmten Wertebereichen benötigt, die von der Quelle (z.B. Festplattenzugriffszeit) so nicht geliefert werden können. Die Methode, wie man die Werte expandiert oder reduziert, ist sehr sorgfältig auszuwählen.

Am sichersten sind solche Zufallsgeneratoren, die ihre Ausgabe durch Messung von wirklich zufälligen Naturphänomenen erzeugen, zum Beispiel anhand von Halbleiter-rauschen oder der kosmischen Strahlung. Dafür benötigt man natürlich spezielle Hardware, die solche Ereignisse in digitale Werte umwandelt. Diese ist in Standardrechnern wie PCs, Workstations usw. bisher werksseitig nicht eingebaut und kann nur in Form eines Zusatzmoduls eingesetzt werden. Jedoch ist es gut möglich und wäre auch sehr wünschenswert, dass in wenigen Jahren ein Hardware-Zufallsgenerator-Chip in jedem Chipsatz von PC-Motherboards enthalten ist, zusammen mit RSA-, Diffie-Hellman- und AES-Chips. Aber dieser Wunsch ist zum Zeitpunkt dieser Auflage immer noch nicht in Erfüllung gegangen.

Vom Standpunkt der Sicherheit aus gesehen sind echte Zufallszahlen, die von spezieller Hardware aus natürlichen Phänomenen (z.B. thermisches Rauschen von Halbleitern) abgeleitet werden, unbedingt zu empfehlen. In normalen Rechnern, Routern und fast allen VPN-Gateways sind hierfür, wenn für die jeweilige Plattform überhaupt verfügbar, allerdings meist Zusatzmodule notwendig.

Echt schlimm sind die portablen Software-Implementierungen, die hardwareunabhängig Pseudo-Zufallszahlen erzeugen. Durch ihre Plattformunabhängigkeit können keine spezifischen Hardware-Informationen (Festplatte, RAM usw.) herangezogen werden, und irgendwie schafft es die Systemzeit doch immer wieder, sich als fortwährend ändernde Größe in den PRNG einzuschleichen.

Hash-Funktionen

Hash-Funktionen sind keine Funktionen, die sich zum Ver- oder Entschlüsseln eignen. Sie werden manchmal auch als Einweg-Kompressionsfunktionen bezeichnet, weil sie aus einem großen, beliebig langen Eingangswert einen kurzen Ausgangswert fester Größe berechnen. Man kann solche Werte benutzen, um Datenbankindizes oder Prüfsummen zu erzeugen. Hash-Funktionen, vor allem solche, die in der Sicherheitstechnologie Verwendung finden, sollen im Allgemeinen folgenden Eigenschaften genügen:

Es soll einfach und somit auch schnell möglich sein, aus einem beliebig langen Eingangswert M einen Hash-Wert h der endlichen Größe n zu erzeugen.

Es soll unmöglich sein, aus diesem Hash-Wert h den zugehörigen Eingangswert M zu berechnen.

Es soll unmöglich sein, zu einer gegebenen Nachricht M eine Nachricht M' zu berechnen, die den gleichen Hash-Wert erzeugt, den auch die Nachricht M erzeugt.

Dies sind die grundsätzlichen Designkriterien. Die ersten beiden sind leicht zu implementieren, beim letzten jedoch drängt sich Ihnen bestimmt die Frage auf, ob dies überhaupt möglich ist. Hierüber streiten sich die Gelehrten, jedoch so viel sei gesagt: Eine Berechnung als solche ist nicht möglich, es gibt bislang keine Formel, keinen Algorithmus, der aus einer beliebigen Nachricht M eine andere Nachricht M' berechnet, so dass beide den gleichen Hash-Wert haben. Jedoch kann man eine Nachricht M' finden, die den gleichen Hash-Wert der Nachricht M hat. Dies ist auch offensichtlich, denn die Nachricht M kann beliebig lang sein und unendlich viele Werte annehmen. Der Hash-Wert hat eine endliche Größe, in der Praxis findet man Werte zwischen 128 und 512 Bit. Somit ist die Menge der Nachrichten M' , die den gleichen Hash-Wert erzeugen, auch unendlich groß – man muss nur eine finden. Ein solches Ereignis nennt man Kollision. Für den unten beschriebenen MD5-Algorithmus und den Secure Hash Algorithm 1 (SHA1) wurden schon Kollisionen gefunden, für andere Verfahren wie RIPEMD noch nicht.

Somit ist noch ein anderes Designkriterium von Wichtigkeit, nämlich dass kleinste Differenzen in der Nachricht M sehr große Änderungen im Hash-Wert h erzeugen müssen. Meist fordert man, dass sich im Hash-Wert 50% aller Bits ändern, falls sich in der Nachricht M nur ein Bit verändert hat. Ändern sich jedoch sehr viele Bits in der Nachricht oder ändert sich auch ihre Länge, dann kann das Ausmaß der Änderung im Hash-Wert ruhig kleiner werden, da sich in diesem Fall die Natur und der Inhalt der Nachricht erheblich geändert haben. Diese Änderungen sind auf anderen Ebenen wie der Applikations- oder Darstellungsschicht im Fall von Datenübertragungen leicht erkennbar.

4.15.1 Hash-Algorithmen

Es gibt eine ganze Reihe von Hash-Algorithmen, von denen hier zwei vorgestellt werden. Die Auswahl fiel deshalb auf die beiden im Folgenden beschriebenen Algorithmen, weil sie im IPSec-Standard als obligatorische Verfahren festgelegt wurden und sich ebenfalls im TLS- und SSL-Standard wiederfinden.

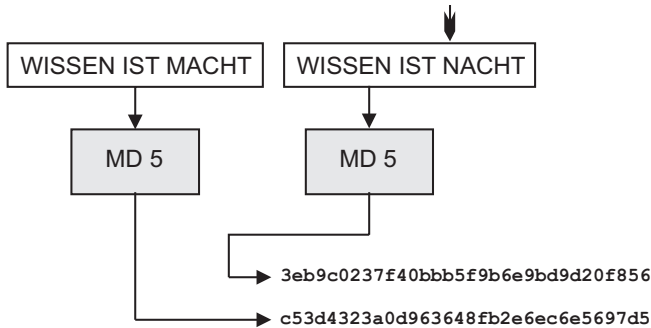


Abbildung 4.30: Die Hash-Werte der MD5-Funktion von zwei leicht unterschiedlichen Eingabegrößen

Message Digest 5 (MD5)

Dieses Verfahren kennen die meisten, zumindest vom Namen her. Es ist nicht nur in IPSec zu finden, sondern auch in allen möglichen anderen Protokollen wie CHAP, OSPF, L2TP, und es gibt sogar PC-Versionen davon, um Dateien auf Veränderungen zu überprüfen. MD5 wurde vom Kryptologie-Guru Ron Rivest entwickelt und ist im RFC1321 inklusive seines C-Quellcodes beschrieben. Der Aufbau von MD5 soll hier nicht zu sehr vertieft werden. Das Verfahren erzeugt aus einem Eingangswert beliebiger Länge einen 128-Bit-Hash-Wert. Neben dem Eingangswert selbst geht auch die Länge des Wertes in die Berechnung des Hash-Werts mit ein. In Abbildung 4.30 sehen Sie das Ergebnis der MD5-Funktion, die auf die beiden Texte angewendet wurde, die sich nur um einen Buchstaben unterscheiden. Die beiden Hash-Werte h und h' sind sehr unterschiedlich, während man den Unterschied der beiden Texte beim ersten Hinsehen vielleicht gar nicht bemerkt.

Secure Hash Algorithm 1 (SHA1)

Der Secure Hash Algorithm (SHA) ist ein neueres Verfahren, das einen größeren Hash-Wert als MD5 erzeugt. Es wird sehr oft als besser als MD5 bezeichnet. Dies liegt vor allem an seinem internen Aufbau, seiner größeren Anzahl an Verarbeitungsrunden und dem höheren Lawineneffekt, der durch verbesserte Rückkopplungsfunktionen erreicht wurde. Auch sind bisher keine Kollisionen bekannt geworden. Andererseits misstrauen viele Anwender dem Algorithmus, da seine Entwicklung maßgeblich von NSA beeinflusst wurde und seine Entwurfskriterien nicht offen gelegt wurden. SHA ist im FIPS-PUB 180-1 (Secure Hash Standard) beschrieben. Er erzeugt einen 160 Bit langen Hash-Wert aus einem beliebig langen Eingangswert.

Keyed Hash

Hash-Verfahren bieten sich an, um die Integrität von Nachrichten zu schützen. Wenn der Sender den Hash-Wert einer Nachricht berechnet und der Empfänger dies mit der von ihm erhaltenen Nachricht tut, dann kann er beide Hash-Werte miteinander vergleichen, um zu prüfen, ob die Nachricht während des Transports verändert wurde. Aber wie kommt der Empfänger zum Hash-Wert des Senders? Eine elegante Art wäre es, den Hash-Wert einfach mit der Nachricht mitzuschicken. In diesem Fall wäre aber kein Schutz mehr gegeben. Denn ein Angreifer könnte sowohl die Nachricht ändern als auch den Hash-Wert neu berechnen.

Dies kann man verhindern, indem man so genannte Keyed-Hash-Funktionen verwendet. Solche Funktionen bilden ihren Hash-Wert aus der Nachricht M und einem Schlüssel K . Dieser Schlüssel ist sowohl dem Sender als auch dem Empfänger bekannt. Jetzt kann man den Hash-Wert mit der Nachricht zusammen verschicken, denn ein Angreifer kann zwar immer noch die Nachricht verändern, aber ohne den passenden Schlüssel kann er den Hash-Wert nicht mehr neu berechnen, und der Empfänger würde sofort anhand der unterschiedlichen Hash-Werte die Änderung bemerken.

Hash-based Message Authentication Code (HMAC)

Nachdem nun die Hash-Funktionen selbst durch symmetrische Schlüssel sicher gegen Änderungen bzw. Neuberechnungen geworden sind, fehlt nur noch eine Verbesserung, um sie resistenter gegen Kollisionen zu machen. Zu diesem Zweck wurde das HMAC-Verfahren entwickelt. Es ist kein Hash-Algorithmus, sondern es kombiniert existierende Verfahren wie MD5 oder SHA mit symmetrischen Schlüsseln und einem speziellen Padding, um auf diese Weise ein starkes Verfahren zur Authentifizierung und Integritätsprüfung zu erhalten. In Abbildung 4.31 ist das Verfahren in Verbindung mit dem MD5-Algorithmus zu sehen. Die HMAC-Verfahren gelten in der Fachwelt als sicher vor Kollisionen, auch wenn das zugrunde liegende Hash-Verfahren dies nicht ist. Auf diese Weise kann man auch Verfahren wie MD5 oder SHA1, für die schon Kollisionen gefunden wurden, weiterhin verwenden und deren Geschwindigkeitsvorteil gegenüber anderen Algorithmen ausnutzen.

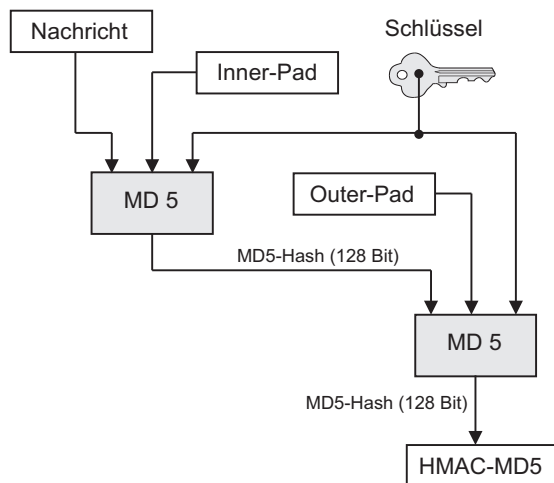


Abbildung 4.31: HMAC-Funktionen schützen zuverlässig vor Kollisionen und bieten die Möglichkeit der Authentifizierung per symmetrischem Schlüssel.

4.15.2 Die Kryptoanalyse von MD5 und SHA1

Es ist schon interessant, wie aufgeregt sich Fachpresse und etliche Internetforen über die Tatsache gezeigt haben, dass es einigen chinesischen Forschern gelungen ist, Kollisionen für MD5 und vor allem für den SHA1 zu erzeugen. Kryptologiegrößen wie Bruce Schneier und etliche seiner Kollegen haben zwar den verdienten wissenschaftlichen Respekt gezollt, sind dann aber wieder weitgehend zur Tagesordnung übergegangen und machen ihr Online-Banking vermutlich immer noch mit SSL und SHA1.

Nun, die Leistung der Kryptoanalytiker ist zwar beeindruckend und anerkennenswert, aber was genau haben sie getan, und was bedeutet es für die Praxis sicherer VPNs?

Was sie getan haben, ist einfach und beruht unter anderem auf dem Geburtstagsparadoxon. Sie haben nämlich zwei Eingangsgrößen nach einem bestimmten Verfahren so lange in Abhängigkeit von den beiden beliebigen Hash-Werten gezielt verändert, bis die beiden unterschiedlichen Größen irgendwann den gleichen Hash-Wert geliefert haben. Dass zwei solche Größen existieren müssen, ist klar, es gibt sogar unendlich viele Paare von Eingangsgrößen, die eine Kollision in jedem bekannten Hash-Verfahren erzeugen. Das Problem ist nur, diese Kollisionen auch zu finden, und hier haben diese Wissenschaftler einiges geleistet. Das Ganze nennt man einen Kollisionsangriff, der bei SHA1 eine theoretische Komplexität von 2^{80} Operationen hat, die aber hier auf etwa 2^{69} Operationen reduziert wurde. Damit ist SHA1 wissenschaftlich gesehen gebrochen.

Beim Finden von Kollisionen gilt es, grundsätzlich zwei Fälle mit jeweils zwei Varianten zu unterscheiden:

1. Man ist völlig frei bei Auswahl der beiden Eingangsgrößen M und M' , die, obwohl unterschiedlichen Inhalts, zwei identische Hash-Werte h und h' erzeugen sollen. Die schwierige Variante liegt dann vor, wenn M und M' gewissen inhaltlich formalen Bedingungen genügen müssen, also gewisse Formate wie IP-Pakete, X.509-Zertifikate, Word-Dokumente usw. einhalten müssen. Das schränkt die Entropie von M und M' extrem ein.
2. Für die nicht veränderliche Nachricht M gibt es einen einzigen Hash-Wert h . Man versucht nun, eine Nachricht M' zu erzeugen bzw. zu finden, die einen Hash-Wert h' erzeugt, der identisch zu h ist.

Die schwierige Variante ist auch hier gegeben, wenn M' gewissen inhaltlich formalen Bedingungen genügen muss, die auch für M gelten. M' muss dann beispielsweise ein gültiges IP-Paket, X.509-Zertifikat oder Word-Dokument sein. Das schränkt die Entropie von M' extrem ein.

Für die VPN-Praxis, nehmen wir einmal IPSec, sind die Forschungen irrelevant, denn hier gibt es eine feste Eingangsgröße, deren Inhalt durch eine meist starke Verschlüsselung verborgen ist. Man muss hier also eine Eingangsgröße M' suchen, die erstens den gleichen Hash-Wert erzeugt und zweitens nach der Entschlüsselung der erzeugten Eingangsgröße beim Empfänger wieder als ein formal korrektes, sinnvolles IP-Paket herauskommt! Alleine das Finden eines gleichen Hash-Wertes zu einer festen, nicht veränderbaren Eingangsgröße besitzt eine Komplexität von 2^{160} Operationen, das Knacken der Verschlüsselung einmal ganz außer Acht gelassen. Vergessen wir das Ganze!

In vielen Protokollen, so auch IPSec, ist außer der Verschlüsselung, die bei IPSec ja optional ist, noch eine andere Barriere gegen Kollisionen eingebaut: Denn hier werden Verfahren wie MD5 oder SHA1 in einen HMAC-Algorithmus eingebunden, der eine sehr hohe Sicherheit vor Kollisionen auch bei schwachen Hash-Algorithmen bietet.

Der Grund, warum vorerst noch keine akute Gefahr bei IPSec oder SSL besteht und warum die chinesischen Wissenschaftler erfolgreich Kollisionen gefunden haben, liegt im Geburtstagsparadoxon begründet: Angenommen Sie veranstalten eine Geburtstagsparty. Um mit einer Wahrscheinlichkeit von 50% jemanden unter Ihren Gästen zu finden, der am gleichen Tag wie Sie Geburtstag hat, müssen Sie mindestens 253 Leute einladen. Wenn sie aber nur zwei beliebige Leute suchen, die an einem beliebigen gleichen Tag Geburtstag haben, dann müssen Sie dafür nur 23 Leute einladen! Mit anderen Worten: Sind beide Größen variabel, dann ist der Aufwand niedrig (23 Personen), ist eine der Größen fest, dann ist er sehr hoch (253 Personen).

Ersteres haben die Chinesen gemacht: Man suche zwei beliebige Eingangsgrößen, die ein gleichen beliebigen Hash-Wert besitzen. Die Integrität von IPSec- oder SSL-Paketen mit ihrem festen Hash-Wert und ihrer festen Eingangsgröße zu kompromittieren ist formal um den Faktor 2^{80} komplexer und real aufgrund der Verschlüsselung der Pakete nicht durchführbar.

Einzig wenn man digitale Signaturen auf Basis von MD5 oder SHA1 benutzt, ist nunmehr allerhöchste Vorsicht geboten. Denn wenn man zum Beispiel ein größeres Word-Dokument elektronisch signiert, kann dieses von einem Angreifer im Voraus so modifiziert werden (Pre-Imaging Attack), dass es den gleichen Hash-Wert und damit auch die gleiche Signatur produziert wie das gefälschte Dokument, das untergeschoben werden soll. Denn die elektronische Signatur wird auf die Bytes der Word-Datei angewendet und nicht auf das, was am Bildschirm angezeigt oder ausgedruckt wird. In der Datei können durch den Angreifer nicht druckbare und sichtbare Sonderzeichen eingefügt, Tabs durch Leerzeichen ersetzt und etliche andere Modifikationen vorgenommen werden, die in ihrer Summe eine riesige Anzahl verschiedener Dateien erzeugen, die sich auf Bildschirm und Drucker aber nicht unterscheiden! Das gefälschte Dokument wird ebenfalls in diesen Prozess eingebunden, und man erhält irgendwann zwei völlig verschiedene Dateien mit einem identischen Hash-Wert. Es kann also vorkommen, dass man arglos ein Dokument signiert, dessen gefälschte Version schon existiert und in das der Angreifer nur noch die elektronische Signatur einfügen muss.

In Protokollumgebungen wie IPSec oder SSL sind derartige Pre-Imaging-Angriffe so extrem schwierig und in den meisten Fällen praktisch undurchführbar, dass hier keine Gefahr besteht.

Fazit

Obwohl MD5 und SHA1 formal gebrochen sind, indem ein Verfahren vorgestellt wurde, das schneller als eine Brute-Force-Attacke ist, besteht für die praktische Anwendung von MD5 und SHA1 in IPSec oder SSL gleich aus mehreren voneinander unabhängigen Gründen zurzeit keinerlei Gefahr. Erstens ist kein Pre-Imaging-Angriff möglich, zweitens sind die Eingangsgrößen für den Hash-Wert stark verschlüsselt, und drittens arbeiten IPSec und SSL mit HMACs und nicht mit den reinen Hash-Algorithmen.

5 IP Security (IPSec)

Beim Thema Netzwerksicherheit, denkt heute jeder zuerst an IP Security. Diese Protokolle, eigentlich für IP Version 6 entwickelt und erst später auch für IPv4 spezifiziert, bieten einen umfassenden Schutz für den Datenverkehr auf IP-Ebene. Das war nicht immer so, aber selbst eine Marktmacht wie Microsoft musste ihre proprietäre Technologie durch standardisierter Protokolle wie L2TP und IPSec ergänzen. Aber trotz der Allgegenwart von IPSec gibt es noch sehr viele Defizite, was IPSec kann und was nicht. Dieses Kapitel soll einen vollständigen technischen Überblick über IPSec geben, um die Möglichkeiten und gegebenenfalls Einschränkungen dieser Protokollfamilie beurteilen zu können.

5.1 IP Security im Überblick

IP-Security bietet einen weit reichenden Schutz von IP-Datagrammen. Insbesondere werden folgende Bereiche abdeckt:

- Paketintegrität
- Paketauthentifizierung
- Paketvertraulichkeit
- Verkehrsflussvertraulichkeit
- Schutz vor Replay-Angriffen
- Schutz vor weiteren Denial-of-Service-Angriffen

5.1.1 Paketintegrität

IPSec schreibt bestimmte Verfahren vor, mit denen die Paketintegrität, also der Schutz vor unerlaubter Veränderung der Pakete auf dem Transport, gewährleistet werden kann. Diese Funktionalität ist ein Muss, es gibt nur eine einzige Ausnahme, bei der diese Funktionalität nicht durch spezielle Funktionen sichergestellt werden muss, sondern bei der andere Dienste diesen Schutz mit zur Verfügung stellen. Diese Ausnahme liegt bei einer Verschlüsselung der Daten vor, denn dann kann eine wirkungsvolle Änderung, also ein Angriff, nur erfolgen, wenn der Angreifer den Schlüssel kennt, was aber in IPSec praktisch unmöglich ist (vgl. Kapitel 6, IKE).

Die Paketintegrität wird sichergestellt, indem in das IPSec-Paket ein so genannter Hash-based Message Authentication Code (HMAC) eingefügt wird. Dabei handelt sich um einen kryptografisch durch einen symmetrischen Schlüssel abgesicherten Hash-Wert, der über das zu versendende IP-Paket gebildet wird. Die korrekte Berechnung des HMAC ist nur dem Sender und dem Empfänger des Pakets möglich, da nur diese über den notwendigen Schlüssel verfügen.

5.1.2 Paketauthentifizierung

Die Paketauthentifizierung, also die Gewährleistung, dass das Paket auch vom authentischen Absender und von niemand anderem kommt, der sich mit gefälschten Absenderadressen in die Kommunikation einklinkt, ist praktisch ein Abfallprodukt der Paketintegritätsprüfung. Denn diese bildet ihren HMAC unter anderem aus einem symmetrischen Schlüssel, der ausschließlich dem Sender und Empfänger oder, genauer formuliert, dem Initiator und Responder der IPSec-Sicherheitsassoziation (SA) bekannt ist. Wenn der Empfänger eines IPSec-Pakets den HMAC überprüft und keine Änderungen festgestellt hat, dann sind zwei Dinge sichergestellt: Erstens wurde das Paket auf dem Transport nicht verändert, und zweitens kommt das Paket auch definitiv vom anderen Partner der SA, denn außer dem Empfänger selbst kennt nur noch dieser den symmetrischen Schlüssel, der zur HMAC-Berechnung benötigt wird.

5.1.3 Paketvertraulichkeit

Die Vertraulichkeit der Daten oder – im Falle von IPSec im Tunnelmodus – der vollständigen, privaten IP-Pakete wird durch deren Verschlüsselung erreicht. Die Verschlüsselung der Daten kann auch benutzt werden, um die Paketintegrität und die Paketauthentizität in gewissem Maße sicherzustellen. Denn die in IPSec eingesetzten Algorithmen sind symmetrische Verfahren, deren Sicherheit zusammen mit anderen Faktoren darauf basiert, dass der Schlüssel nur dem Sender und dem Empfänger bekannt ist. Ein verschlüsseltes Paket ändern kann somit auch nur jemand, der den Schlüssel ebenfalls kennt, was man durch den Einsatz des IKE-Protokolls zum Erzeugen der notwendigen Schlüssel wirkungsvoll verhindern kann.

5.1.4 Verkehrsflussvertraulichkeit

IPSec kann man in zwei verschiedenen Modi betreiben, dem Transport-Modus und dem Tunnel-Modus. Letzterer kapselt ein privates IP-Paket in den Datenbereich eines neu erzeugten IP-Pakets ein. Sobald dieser Datenbereich verschlüsselt wird, kann sich ein Angreifer, der das neue Paket auf dem Transport mitliest, keine Informationen mehr über Verkehrsbeziehungen im privaten Netzwerk verschaffen. Denn das komplette private Paket, inklusive Adressen, Protokollnummern usw., ist verschlüsselt. Die einzige Information, die ein Angreifer ermitteln kann, ist die Gesamtmenge an Paketen, die zwischen IPSec-Gateways oder -Hosts verschickt werden.

5.1.5 Schutz vor wiederholtem Senden von Paketen (Replay-Angriff)

Ein Replay-Angriff ist eine Art von Angriff, mit der ein Angreifer trotz Schutzmechanismen wie Integritätsprüfung, Authentifizierung und zusätzlicher Datenverschlüsselung seine Pakete in eine IPSec-Verbindung einschleusen und den Empfänger zu umfangreichen, ressourcenintensiven, aber völlig sinnlosen Berechnungen veranlassen kann. Die Sinnlosigkeit würde aber erst von der Applikation erkannt, die das private IP-Paket anschließend weiterverarbeitet. Der Trick des Angreifers besteht einfach darin, dass er

IPSec-Pakete aufzeichnet und diese nochmals zum Empfänger schickt. Dieser verarbeitet das Paket auch ganz normal, denn es ist mit einem korrekten HMAC versehen, lässt sich entschlüsseln und sieht von der Sicherheitsseite aus betrachtet normal aus. Wenn nun eine ausreichend große Anzahl von solchen Paketen wiederholt zum Empfänger geschickt wird, muss er erhebliche Ressourcen für die kryptografische Verarbeitung aufwenden, die ihm dann für die Verarbeitung ordnungsgemäßer Pakete schließlich fehlen.

Um sich vor derartigen Angriffen zu schützen, wurden in IPSec spezielle Maßnahmen eingeführt. IP-Security überprüft hierbei optional durch einen Anti-Replay Service (ARS) jedes eingehende Paket zuallererst daraufhin, ob es schon einmal angekommen ist.

5.1.6 Schutz vor weiteren Denial-of-Service-Angriffen

Es gibt mittlerweile eine Reihe teilweise recht ausgeklügelter Denial-of-Service-Angriffe, also Attacken, die darauf abzielen, ein Gerät in einen Zustand zu versetzen, in dem es seine ursprünglichen Aufgaben nicht oder nur noch in unzureichendem Maße ausführen kann. Solche Angriffe zielen oft auf den TCP-Stack ab oder versuchen, rechenintensive Funktionen in einem System anzustoßen. Einen Schutz hiervoor kann man erreichen, indem man auf viele Funktionen eines normalen IP-Stacks verzichtet und einen so genannten gehärteten Stack implementiert. Dies ist allerdings nur bei dedizierten VPN-Gateways möglich, die beispielsweise auf ihrem öffentlichen Interface ausschließlich Tunneling-Protokolle wie IPSec und L2TP terminieren.

Ein gehärteter Stack ist so programmiert, dass er nur die Protokolle 50 (IPSec ESP), 51 (IPSec AH) und 17 (UDP) mit den Portnummern 500 oder 4500 in die Schicht 3 durchlässt. Eine TCP-Verarbeitung gibt es auf dieser Ebene nicht, dadurch fällt schon eine ganze Reihe möglicher Angriffe weg. Die IPSec-Protokolle selbst haben bereits eine ganze Reihe eingebauter Schutzmechanismen, ebenso das IKE-Protokoll, das mit UDP-Paketen der Portnummer 500 oder 4500 arbeitet. Alle anderen Pakete werden bereits vor der Ebene 3 ohne weitere Meldung (Silent Discard) verworfen und können keinen weiteren Schaden mehr anrichten.

5.2 Kryptografische Verfahren in IPSec

Diese ganzen Anforderungen an die Sicherheit der Datenübertragung bedingen, dass eine Reihe von kryptografischen Verfahren eingesetzt wird, flankiert von gewissen Richtlinien zur Implementierung in IPSec, um die verschiedenen möglichen Angriffe abzuwehren.

5.2.1 Datenverschlüsselung

Die Datenverschlüsselung in IPSec wird mit symmetrischen Verfahren durchgeführt, die im Cipher-Block-Chaining-Modus (CBC) mit explizitem Initialisierungsvektor (IV) betrieben werden. DES, der Data Encryption Standard, ist als Algorithmus zwingend von jeder Implementation zu unterstützen, andere Verfahren wie Triple-DES, IDEA, Cast, Blowfish, RC5 usw. können optional implementiert werden.

Alle diese symmetrischen Verfahren bedingen, dass in jedem Endpunkt einer IPSec-Verbindung (Sicherheitsassoziation) die gleichen, ausreichend langen Schlüssel vorliegen. Wie diese Schlüssel dorthin gelangen, wird in Kapitel 8 näher beschrieben. Wichtig zu wissen ist an dieser Stelle, dass diese Schlüssel vor dem Einsatz des IPSec-ESP-Protokolls, das die Datenverschlüsselung implementiert, den beiden Kommunikationspartnern zur Verfügung stehen müssen. Je nach Algorithmus werden Schlüssel verschiedener Länge benutzt.

Die am häufigsten in IP-Security eingesetzten Algorithmen zur Verschlüsselung und die von diesen verwendeten Schlüssellängen sind:

DES, mit einem 56 Bit langen Schlüssel

Triple-DES, das einen aus drei 56-Bit-Teilschlüsseln bestehenden 168-Bit-Schlüssel benötigt

IDEA benutzt einen 128 Bit langen Schlüssel.

Blowfish arbeitet mit einem 448 Bit großen Schlüssel.

AES mit 128 oder 256 Bit großen Schlüsseln

Der Einsatz dieser Verfahren, die in der Regel monoalphabetisch sind, muss im CBC-Modus erfolgen. Der Initialisierungsvektor für diese Betriebsart wird in jedem Paket mit übertragen und dem Chiffretext unmittelbar vorangestellt. In Abbildung 5.1 ist eine Verschlüsselung mit dem DES-Algorithmus in IPSec schematisch dargestellt. Man kann hier auch erkennen, wie die Verschlüsselung die Paketgröße verändert, da der Initialisierungsvektor und eventuell notwendige Füllzeichen (Padding) in das Paket mit aufgenommen werden müssen. In diesem Fall, einem zugegebenermaßen sehr ungünstig konstruierten, werden die 25 Byte Originaldaten auf 40 Byte vergrößert, also um mehr als 60%.

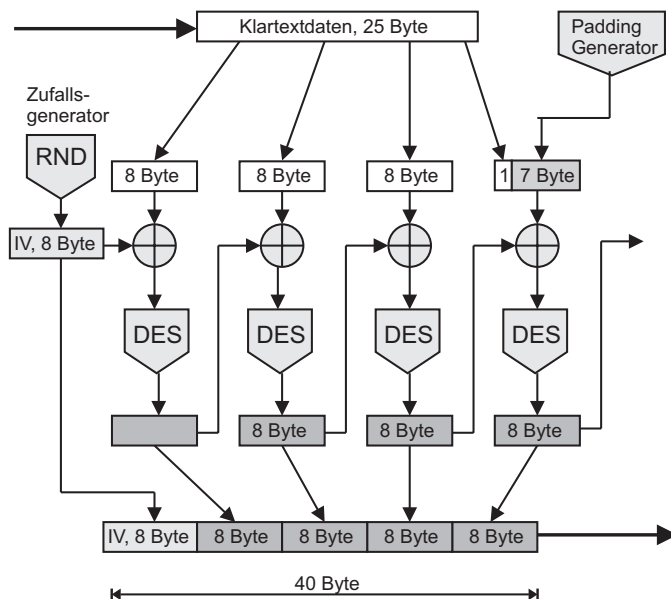


Abbildung 5.1: IPSec-Datenverschlüsselung mit DES-CBC mit explizitem Initialisierungsvektor (IV)

Der Initialisierungsvektor sollte zufälligen Ursprungs sein und möglichst nicht aus dem verwendeten Schlüssel des Verschlüsselungsprogramms abgeleitet werden. Er kann ungesichert übertragen werden und wird dem Chiffretext in einem IPSec-ESP-Paket direkt vorangestellt. Für nachfolgende Pakete kann der letzte Chiffretextblock als IV verwendet werden.

Im vorliegenden Beispiel ist weiterhin zu sehen, wie der Klartext in für den jeweiligen Chiffrieralgorithmus passende Teile zerlegt wird. Im Fall von DES oder auch Triple-DES müssen diese Blöcke genau 64 Bit groß sein. Falls, wie in Abbildung 7.1, der letzte Block keine 64 Bit (8 Byte) lang ist, muss er mit Füllzeichen auf die erforderliche Länge aufgefüllt werden.

Da das DES-Verfahren als mittlerweile nicht mehr ausreichend angesehen wird, um sensible Daten zu schützen, bieten praktisch alle für dieses Buch untersuchten IPSec-Implementierungen auch Triple-DES als optionalen Algorithmus an, die meisten zusätzlich noch AES.

5.2.2 Integritätsprüfung und Authentifizierung

Zum Schutz der Datenintegrität und zur Authentifizierung greift man auf Hash-Algorithmen zurück, die in entsprechenden Betriebsarten sicher vor so genannten Kollisionen sind. Die in IPSec vorgeschriebenen Verfahren sind HMAC-MD5-96 und HMAC-SHA1-96, also Algorithmen, die auf den bekannten HashFunktionen MD5 und SHA1 aufsetzen. Die Funktionsweise dieser Verfahren ist in Kapitel 4 beschrieben.

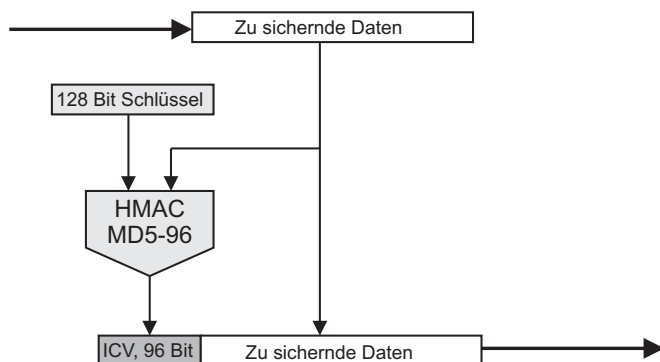


Abbildung 5.2: In IPSec kann man HMAC-MD5-96 zur Authentifizierung und Integritätsprüfung von Datenpaketen einsetzen.

Diese Verfahren arbeiten ebenfalls mit einem symmetrischen, 128 Bit langen Schlüssel, der ausschließlich den beiden Kommunikationsendpunkten bekannt sein darf. Dieser Schlüssel wird ebenfalls vom IKE-Protokoll generiert und muss vorhanden sein, sobald IPSec-Pakete übertragen werden. In Abbildung 5.2 ist die generelle Funktion innerhalb IPSec skizziert, wobei die Position des ICV (Integrity Check Value) im ausgehenden Paket vom jeweiligen IPSec-Protokoll abhängig ist. Es hängt auch vom jeweiligen Protokoll ab, welche Bereiche des Pakets gesichert werden. Näheres dazu ist in den Kapiteln zum AH- und ESP-Protokoll zu finden.

Die Hash-Verfahren werden nicht direkt eingesetzt, sondern arbeiten im HMAC-Modus, der die Verfahren sicher vor so genannten Kollisionen macht. Eine Kollision liegt dann vor, wenn zwei verschiedene Eingabewerte den gleichen Hash-Wert erzeugen. Für das MD5-Verfahren hat man bereits Kollisionen demonstriert, jedoch waren diese eher akademischer Natur. Es gibt zum jetzigen Zeitpunkt keine bekannte mathematische Methode, solche Werte zu berechnen, die einen gleichen Hash-Wert produzieren, insbesondere dann nicht, wenn sich diese Werte ähneln sollen und gleich lang sein müssen. Nichtsdestoweniger hat man das Kollisionsproblem in Fachkreisen zur Kenntnis genommen und lässt als Schutzmaßnahme dagegen die Hash-Verfahren generell im HMAC-Modus arbeiten; aus dem MD5-Verfahren wird z.B. HMAC-MD5-96. Die Zahl 96 bedeutet, dass der Hash-Wert auf 96 Bit verkürzt wird, indem man die niederwertigen 32 Bit aus dem Wert entfernt. In den RFCs 2403 und 2404 sind die Benutzung der Hash-based Message Authentication Codes auf Basis von SHA1 und MD5 spezifiziert, das HMAC-Verfahren selbst ist im RFC2104 beschrieben.

IPSec erlaubt auch die optionale Verwendung anderer Verfahren zur Integritätssicherung, jedoch sind in heutigen Implementierungen fast ausschließlich HMAC-MD5-96 und HMAC-SHA1-96 zu finden.

5.2.3 Schutz vor Replay-Angriffen

Der Schutz vor wiederholtem Senden von Paketen (Replay-Angriff) wird in IPSec durch den ARS (Anti Replay-Service) gewährleistet. Zu diesem Zweck ist in jedem IPSec-Paket, egal mit welchem Protokoll es erzeugt wurde, ein 32 Bit großes Feld enthalten, in dem eine monoton steigende Sequenznummer enthalten ist. Sobald eine IPSec-Sicherheitsassoziation aufgebaut wird, wird dieser Zähler mit 0 initialisiert. Obwohl der ARS in IPSec optional ist, wird in ausgehenden Paketen der Zähler immer um eins erhöht. Falls auf ARS verzichtet wird, betrifft dies nur den Empfänger, er wertet das Feld in diesem Fall einfach nicht aus.

Da die Pakete aufgrund der Wegefindung in IP-Netzen nicht in der Reihenfolge ankommen müssen, in der sie abgeschickt wurden, reicht es nicht aus, im Empfänger zu prüfen, ob die Sequenznummer des ankommenden Pakets um eins höher ist als die des vorhergehenden Pakets der entsprechenden Sicherheitsassoziation. Stattdessen verwaltet der Empfänger ein so genanntes Empfangsfenster. Dieses Fenster ist meist 64 Bit, mindestens jedoch 32 Bit groß. Pakete, die innerhalb dieses Fensters ankommen, werden verarbeitet. Pakete, die links davon liegen, sind entweder schon einmal eingetroffen oder kommen schlicht zu spät an und werden verworfen. Falls ein Paket eintrifft, das rechts vom aktuellen Fenster liegt, wird es der IPSec-Integritätsprüfung unterworfen. Falls es diese besteht – und nur dann –, wird das Fenster so weit verschoben, bis das Paket an der rechten Grenze des Fensters steht.

In Abbildung 5.3 wird die Funktionsweise an einem 32 Bit breiten ARS-Fenster demonstriert. Das linke Element im Fenster hat den Wert N, wobei N dem Wert des Sequenzzählers des Pakets entspricht, das an dieser Stelle angekommen ist. Das Paket mit der Sequenznummer N-6 fällt nicht in das Fenster, sondern liegt links davon. Dies bedeutet, dass das Paket entweder zu spät eintrifft oder bereits schon einmal eingetroffen ist, also wird es verworfen. Das Paket N+7 liegt zwar im Bereich des Fensters, ist aber schon einmal angekommen und wird somit auch verworfen. Das Paket N+21 liegt im Fenster, ist auch noch nicht vorher angekommen und kann demnach verarbeitet werden.

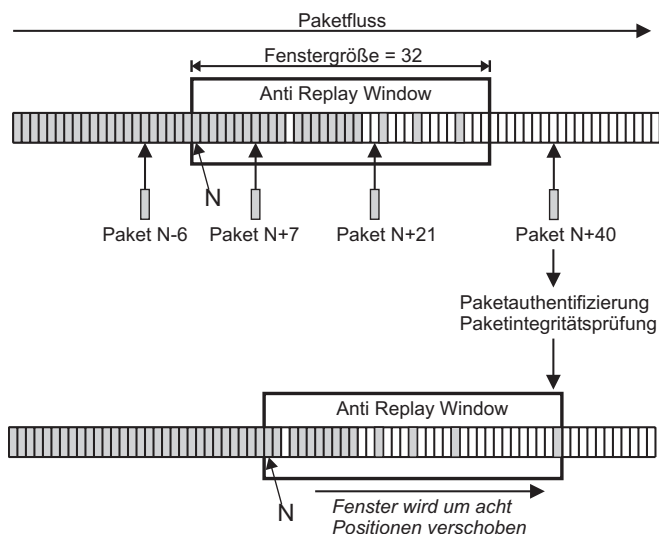


Abbildung 5.3: Der IPsec Anti Replay Service (ARS)

Das Paket N+40 hingegen liegt rechts des Fensterbereichs. Es wird auf Integrität und Authentizität geprüft, und falls diese Prüfung erfolgreich war, wird das Fenster so weit nach rechts verschoben, bis das neue Paket darin erscheint.

An dieser Stelle vielleicht ein Tipp zur Konfiguration von IPsec-Sicherheitsassoziationen: Viele Implementierungen bieten die vom Standard ausdrücklich zugelassene Möglichkeit an, beim Einsatz der Verschlüsselung im ESP-Protokoll auf die explizite Prüfung der Integrität und Authentizität durch HMAC-MD5-96 oder HMAC-SHA1-96 zu verzichten. In diesem Fall besteht aber leider die Möglichkeit, dass bestimmte Denial-of-Service-Angriffe trotz aktivem ARS ausgeführt werden. Denn wenn ein Angreifer den Verkehr mitliest und ständig selbst Pakete mit der aufgezeichneten verschlüsselten Nutzlast erzeugt, deren Sequenznummern sehr weit rechts außerhalb des Fensters liegen, passieren folgende zwei fatalen Dinge: Erstens wird das Fenster sehr weit nach rechts verschoben, so dass die ordnungsgemäß eintreffenden Pakete links außerhalb des Empfangsbereichs liegen und daher verworfen werden. Zweitens werden ausgerechnet die gefälschten Pakete weiterverarbeitet und durchlaufen die rechenintensive Entschlüsselung.

IPsec selbst kann nicht mehr erkennen, dass es sich um unzulässige Pakete handelt, das können nur noch die höheren Anwendungsschichten, also UDP, TCP oder IP, falls es sich um IPsec im Tunnel-Modus handelt. Erst diese Protokolle stellen dann fest, dass bei der Entschlüsselung der Pakete ein ziemlicher Unsinn herausgekommen ist, da die Daten bereits schon einmal angekommen sind. Bis zu diesem Punkt wurde aber schon einiges an Ressourcen aufgewendet, die an anderer Stelle fehlen.

Man sollte also in jedem Fall die explizite Paketintegritätsprüfung und Paketauthentifizierung benutzen, auch bei der Verschlüsselung der Daten. Die verwendeten Algorithmen und Systeme sind mittlerweile recht schnell, so dass hierdurch keine nennenswerten Leistungseinbußen entstehen.

5.3 Die IPSec-Standards

Die Standardisierung von IPSec ist recht weit fortgeschritten. Vor allem in den RFCs 2401 bis 2412 und einigen anderen, darin angesprochenen Dokumenten ist alles Wissenswerte über IP-Security nachzulesen. In letzter Zeit neu hinzugekommen sind vor allem Erweiterungen für neue Kryptoalgorithmen, NAT-Traversal und Dead-Peer-Detection.

5.3.1 Die IPSec-Architektur

Die Architektur von IPSec ist in einer Reihe von RFCs zusammengefasst, die im folgenden Kapitel beschrieben sind. Die IPSec-Architektur selbst können Sie als Übersicht in Abbildung 5.4 sehen.

Ausgehend von dieser Architektur und den jeweiligen Anwendungsstrategien ergeben sich folgende typische Kombinationen von IPSec-Diensten:

ESP mit Verschlüsselung, Integritätsprüfung, Authentifizierung und ARS

ESP mit Verschlüsselung, Integritätsprüfung und Authentifizierung

ESP mit Verschlüsselung und ARS

ESP mit Verschlüsselung

ESP mit Integritätsprüfung, Authentifizierung und ARS

ESP mit Integritätsprüfung und Authentifizierung

AH mit Integritätsprüfung, Authentifizierung und ARS

AH mit Integritätsprüfung und Authentifizierung

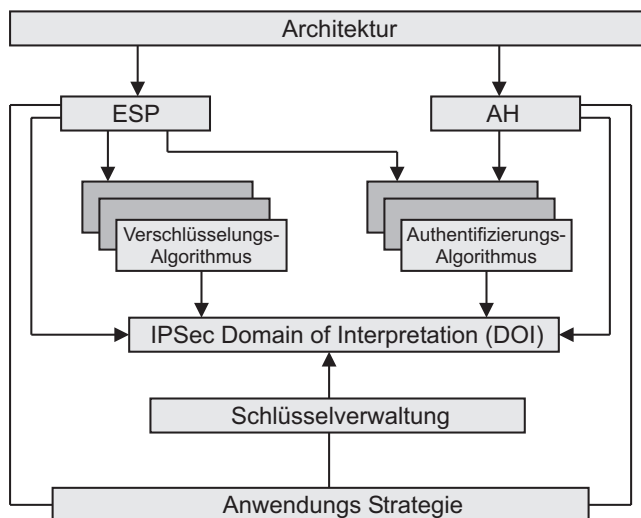


Abbildung 5.4: Die Architektur von IPSec

Weiter differenzieren kann man dies noch durch die verschiedenen Algorithmen, die für die verschiedenen Dienste zur Verfügung stehen. Im Bereich des Schlüsselmanagements

kann die Anwendungsstrategie festlegen, wie dieses zu erfolgen hat und auf welche Art sich die Kommunikationspartner gegenseitig authentifizieren. Diesem Teil, dem Internet Key Exchange (IKE), ist das nächste Kapitel gewidmet.

5.3.2 Die aktuelle IPSec-Standardisierung

Hier folgt ein kurzer, nicht vollständiger Überblick über die wichtigsten RFCs der IPSec-Arbeitsgruppe der Internet Engineering Task Force:

- RFC 2401: Security Architecture for the Internet Protocol
- RFC 2402: IP Authentication Header
- RFC 2403: The Use of HMAC-MD5-96 within ESP and AH
- RFC 2404: The Use of HMAC-SHA-1-96 within ESP and AH
- RFC 2405: The ESP DES-CBC Cipher Algorithm with explicit IV
- RFC 2406: IP Encapsulating Security Payload (ESP)
- RFC 2407: The Internet IP-Security Domain of Interpretation for ISAKMP
- RFC 2408: Internet Security Association and Key Management Protocol
- RFC 2409: The Internet Key Exchange
- RFC 2412: The OAKLEY Key Determination Protocol
- RFC 2451: The ESP CBC-Mode Cipher Algorithms
- RFC 2857: The Use of HMAC-RIPEMD-160-96 within ESP and AH
- RFC 3526: More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)
- RFC 3566: The AES-XCBC-MAC-96 Algorithm and Its Use With IPSec
- RFC 3602: The AES-CBC Cipher Algorithm and Its Use with IPSec
- RFC 3664: The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)
- RFC 3686: Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)
- RFC 3706: A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers
- RFC 3715: IPsec-Network Address Translation (NAT) Compatibility Requirements
- RFC 3947: Negotiation of NAT-Traversal in the IKE
- RFC 3948: UDP Encapsulation of IPsec ESP Packets

5.4 Die IPSec Security Association

Eine Sicherheitsassoziation (SA) ist das Fundament von IPSec. Sie bestimmt das Verhalten dieses Sicherheitsprotokolls, also welche Verschlüsselungsalgorithmen verwendet werden, die Lebensdauer der SA, das Authentifizierungsprotokoll usw. Die verschiedenen SAs werden in der Security Association Database (SAD) gespeichert. Eine IPSec-

Sicherheitsassoziation ist immer unidirektional. Dies bedeutet, wie in Abbildung 5.5 gezeigt, dass in der Praxis für bidirektionalen Datenverkehr immer mindestens zwei SAs zwischen den Kommunikationsendpunkten existieren müssen. In diesem Fall hat jedes der beiden Systeme eine eingehende (Inbound-SA) und eine ausgehende (Outbound-SA) Sicherheitsassoziation. Es können aber durchaus auch mehrere SAs über eine Verbindung existieren, wenn Pakete mit unterschiedlichen Sicherheitsstufen transportiert werden müssen.

Der Security Parameter Index (SPI), der in jedem IPSec-Protokoll-Header eingetragen ist, ordnet jedes Paket eindeutig einer SA zu. Denn falls innerhalb einer IP-Verbindung mehrere SAs mit dem gleichen Protokoll existieren, die sich z.B. nur in der Stärke der Verschlüsselung unterscheiden, dann ist der SPI der einzige eindeutige Wert, denn die Quell- und die Zieladresse sowie Protokollnummer wären in jedem IP-Paket gleich.

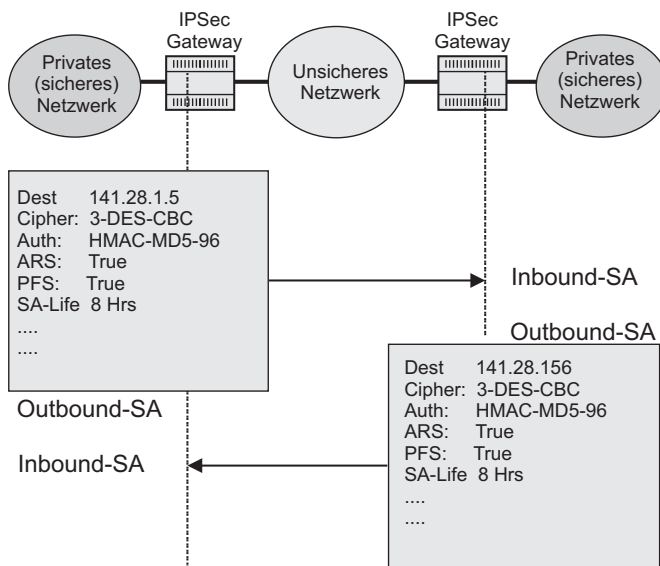


Abbildung 5.5: Eine Security Association in IPSec ist immer unidirektional.

Eine weitere, allerdings in keinem Standard festgelegte Verwendung des SPI findet man bei kleinen SOHO-Routern (Small Office Home Office) verschiedener Hersteller, die ihre Verbindung zum Internet über Wählverbindungen mit dynamischer Zuweisung von IP-Adressen herstellen. Ohne den Einsatz von IPSec wurden die verschiedenen Rechner hinter diesem Router per NAT/PAT (Network Address Translation/Port Address Translation) auf die vom ISP zugewiesene IP-Adresse abgebildet, indem man ihre TCP- oder UDP-Portnummern ändert und der Router eine Tabelle pflegt, in der die privaten Adressen den Portnummern zugewiesen werden.

Mit IPSec ist dies nicht möglich. Erstens besitzen ESP oder AH überhaupt keine Portnummern, und zweitens werden die Pakete auf Integrität geprüft. Mit AH funktioniert dies demnach gar nicht. Mit ESP, das den äußeren Header nicht auf Integrität prüft, kann man eine andere Verteilungsstrategie der Pakete im privaten Netz einsetzen. Da der SPI eindeutig ist, kann man im Router eine Tabelle pflegen, in der die Pakete aufgrund des

SPI dem richtigen Rechner zugeordnet sind. Allerdings funktioniert dies nur, wie in Kapitel 6 erklärt wird, wenn die Rechner IKE im so genannten Aggressive Mode einsetzen, was die meisten PC-Implementierungen aber können.

SAs werden in der Regel durch das IKE-Protokoll erzeugt. Falls die IPSec-Sicherheitsstrategie ein Paket prüft, das IPSec-Diensten zugeführt werden muss, wofür aber noch keine Sicherheitsassoziation existiert, dann startet IPSec das IKE-Protokoll, um diese zu generieren. Sobald dieser Prozess abgeschlossen ist, kann das Paket ordnungsgemäß verarbeitet und anschließend übertragen werden. Die manuelle Erzeugung von SAs wird im Standard ebenfalls berücksichtigt, ist aber wegen der Fehleranfälligkeit und der immanenten Sicherheitsprobleme in fast keinem System implementiert.

Das Löschen von Sicherheitsassoziationen erfolgt durch das Protokoll, das sie erzeugt hat, meist IKE. Die Gründe hierfür liegen entweder in der Sicherheitsstrategie, wenn die Lebensdauer der SA abgelaufen ist oder wenn die maximale Anzahl von Bytes, die innerhalb der SA übertragen wurden, überschritten ist. Oder die Gegenstelle verlangt, dass die SA gelöscht wird, beispielsweise weil ein neuer Schlüssel ausgetauscht wird.

Eine Sicherheitsassoziation pflegt eine Datenbank, in der Felder bzw. Strukturen enthalten sind, die für den Betrieb der SA notwendig sind. Diese Felder sind:

Sequence Number Dies ist der Sequenzzähler für ausgehende Pakete.

Anti Replay Window Dies ist das ARS-Fenster. Es wird für ankommende Pakete verwendet.

Sequence Number Overflow Dieses Feld gibt an, ob nach einem Überlauf des Sequenzzählers noch weiter übertragen werden darf, damit die SA weiter existieren kann.

Lifetime Dieses Feld enthält die Lebensdauer der SA durch Angabe der zeitlichen Länge, der Anzahl der maximal zu übertragenden Bytes oder beides. Es gibt darüber hinaus auch zwei Arten der Lifetime – eine, welche die SA sofort nach Eintritt der Terminierungskondition beendet (hard), und eine andere (soft), die dem IPSec-Kernel eine Nachricht schickt, damit dieser eine neue SA erzeugen kann und die Kommunikation nicht unterbrochen wird.

Mode Hier wird angezeigt, ob IPSec innerhalb dieser SA im Tunnel- oder Transportmodus arbeitet.

Tunnel Destination Hier steht, falls es sich um einen IPSec-Tunnel-Modus handelt, die IP-Zieladresse des äußeren Headers.

PMTU Parameters Bei IPSec im Tunnel-Modus werden die PMTU-Informationen, die man zur Fragmentierung benötigt, in dieser Struktur gespeichert.

5.5 Die IPSec Security Policy

5.5.1 Die Security Policy in IPSec

Die Sicherheitsstrategie oder Security Policy in IPSec legt fest, welche Dienste auf ein- oder ausgehende Pakete anzuwenden sind. Die Regeln, aus denen sich diese Strategie zusammensetzt, sind in einer speziellen Datenbank, der Security Policy Database (SPD), abgelegt. Diese Strategie wählt eine von drei Arten aus, wie die Pakete behandelt werden. Die Pakete werden entweder verworfen (discard), nicht durch einen IPSec-Dienst transformiert und sofort weiterbearbeitet (bypass) oder durch IPSec umgeformt (apply) und dann weiterverarbeitet.

Für ausgehende Pakete liefert die SPD im letzten der drei Fälle (apply) einen Verweis auf einen Eintrag in der Security Association Database, in der die spezifischen Optionen und Parameter hinterlegt sind, mit denen eine Sicherheitsassoziation aufgebaut oder bereits unterhalten wird. Falls die ausgewählte SA noch nicht aktiv ist, wird ein Key-Management-Protokoll, in der Regel IKE, gestartet, um sie zu etablieren. Erst dann wird das Paket übertragen.

Die Security Policy Database ist eine Datenstruktur im Kernel (Kernbereich) des Betriebssystems, auf dem der IPSec-Stack implementiert ist. Auf Applikationsebene müssen entsprechende Werkzeuge zum Ändern, Löschen und Hinzufügen von Einträgen existieren. Zunehmend werden auch Schnittstellen zu netzwerkweiten Verzeichnisdiensten wie LDAP eingesetzt, um eine übergreifende Sicherheitsstrategie einsetzen zu können.

5.5.2 Die IPSec-Selektoren

Die Security Policy Database enthält verschiedene Selektoren, mit denen festgelegt wird, welche Dienste auf bestimmte Pakete angewendet werden. Folgende Informationen aus dem IP-, TCP- und UDP-Header der zu inspizierenden Pakete können als Selektoren benutzt werden:

- Zieladresse
- Quelladresse
- Protokoll
- Portnummer (des Transportprotokolls)
- Name

Zieladresse

Die IP-Zieladresse kann in Form von Adressbereichen, Netzwerken, Hostadressen oder ohne weitere Spezifikation (Wildcard) angegeben werden. IPSec-Gateways benutzen in der Regel keine Hostadressen und bieten daher allen dahinter liegenden privaten Systemen Schutz.

Quelladresse

Die IP-Quelladresse kann ebenfalls in Form von Adressbereichen, Netzwerken, Hostadressen oder als Wildcard angegeben werden.

Protokoll

Dieser Selektor bezieht sich, sofern vorhanden, auf das Protokollfeld des Transportprotokolls. Falls kein Transportprotokoll verwendet wird, z.B. beim Verschachteln von IPSec-SAs, wird das Feld nicht verwendet.

Portnummer (des Transportprotokolls)

Dieser Selektor bezieht sich auf das Tupel aus Ziel- und Ursprungsportnummer des Transportprotokolls (TCP, UDP ...). Auch hier können Wildcards benutzt werden.

Name

Dieses Feld wird im Augenblick von IPSec nicht benutzt. Es wird momentan nur vom IKE-Protokoll als Selektor verwendet.

5.6 Die IPSec-Betriebsmodi

5.6.1 Tunnel-Modus

Der Tunnel-Modus wird eingesetzt, um komplette IP-Pakete einschließlich Header zu schützen. Dies wird erreicht, indem ein neues IP-Paket erzeugt wird, in dessen Datenbereich das zu übertragende Paket eingefügt wird. Auf diese Weise erreicht man auch eine weitgehende Vertraulichkeit des Verkehrsflusses, da ein Angreifer beim Einsatz des ESP-Protokolls keinerlei Informationen über das Originalpaket erhält. Alle Adressen und Protokollnummern des Originalpakets sind verschlüsselt.

Der Tunnel-Modus kann in allen drei IPSec-Einsatzszenarien verwendet werden, also in Gateway-zu-Gateway-, Host-zu-Gateway- und Host-zu-Host-Verbindungen.

Im Gegensatz zu anderen Tunneling-Protokollen, die es erlauben, auch Pakete anderer Netzwerkprotokolle, wie z.B. IPX, einzukapseln, ist mit IPSec nur IP-in-IP-Tunneling möglich.

5.6.2 Transport-Modus

Der IPSec-Transport-Modus wird benutzt, um höhere Protokollschichten zu schützen, die im Datenbereich der IP-Pakete eingekapselt sind. Der Paket-Overhead ist im Gegensatz zum Tunnel-Modus etwas geringer, da kein zusätzlicher IP-Header konstruiert werden muss.

Der Transport-Modus kann ausschließlich in IPSec-Host-zu-Host-Verbindungen benutzt werden, da der Kommunikationsendpunkt auch gleichzeitig der Sicherheitsendpunkt ist. Weiterhin können sich Angreifer Informationen über die Netzwerkstruktur und internen Verkehrsbeziehungen verschaffen – selbst beim Einsatz der Verschlüsselung im ESP-Modus, die in diesem Fall nur auf das höhere Protokoll angewendet wird. Protokollnummern geben weiterhin Aufschluss über eingesetzte Anwendungen.

5.7 IPSec-Einsatzszenarien

Abhängig davon, wo die IPSec-Verbindungen initiiert und terminiert werden, kann man die möglichen Einsatzszenarien in drei Gruppen unterteilen, die in vielen praktischen Fällen auch gemischt eingesetzt werden können:

Gateway-zu-Gateway

Host-zu-Gateway

Host-zu-Host

5.7.1 Gateway-zu-Gateway

Hier erfolgt der Schutz der IP-Datagramme durch IPSec zwischen zwei Gateways. Es sind meist VPN-Gateways oder Router. Der Haupteinsatzzweck ist die Verbindung von privaten Netzen über unsichere Verbindungswege.

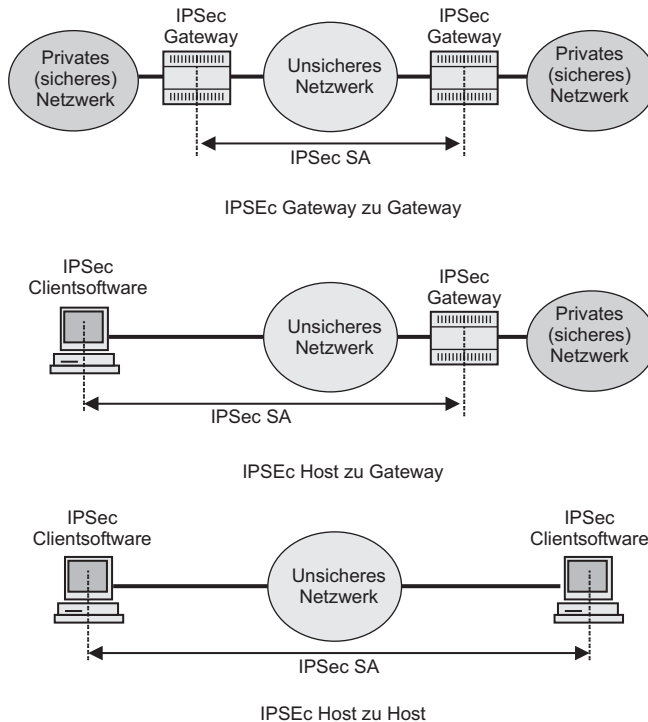


Abbildung 5.6: Die verschiedenen IPSec-Einsatzszenarien

5.7.2 Host-zu-Gateway

Hier erfolgt der Schutz der IP-Pakete auf dem Transport von einem Endgerät (Host) zu einem Gateway. Ein typisches Einsatzgebiet ist Remote Access über unsichere Transportnetze.

5.7.3 Host-zu-Host

Dieses Szenario, in dem zwischen Endsystemen IP-Datagramme übertragen werden, trifft man selten an. Ein mögliches Einsatzgebiet sind Intranet-VPN.

5.8 Die IPSec-Protokolle

5.8.1 Die Paketverarbeitung in IPSec

Ausgehende Pakete

Ausgehende Pakete werden von der höheren Schicht (Transportebene) an das IP-Protokoll weitergegeben. Es erfolgt eine Abfrage in der Security Policy Database, was mit dem betreffenden Datagramm geschehen soll. Abhängig von den Informationen über das Paket gibt es folgende drei Möglichkeiten:

Discard (Fallenlassen), das Datagramm wird verworfen.

Bypass (Umgehen), das Datagramm umgeht IPSec und wird verschickt.

Apply (Anwenden), es wird ein Verweis auf eine SA zurückgeliefert, das Datagramm wird entsprechend verarbeitet und anschließend verschickt.

Falls IPSec feststellt, dass für das betreffende Paket im dritten Fall (Apply) noch keine SA existiert, wird das IKE-Protokoll aufgerufen, um die notwendige Sicherheitsassoziation zu erzeugen. Das Paket wird erst übertragen, nachdem die SA erzeugt wurde.

Diese Verarbeitung unterscheidet sich grundsätzlich von der ausgehender Pakete. Bei eingehenden Paketen prüft die IPSec-Schicht zuerst, ob es sich um ein IPSec-Paket handelt und ob dafür bereits eine Sicherheitsassoziation aufgebaut wurde. Falls keine SA existiert, wird das IPSec-Paket sofort fallen gelassen. Im anderen Fall wird aufgrund der Informationen in der Security Association Database die notwendige Transformation des Pakets durchgeführt.

Falls das Paket kein IPSec-Paket ist, wird geprüft, ob auf das Paket eine Security Policy anwendbar ist. Wenn dies der Fall ist, aber keine Sicherheitsassoziation dafür besteht, wird das Paket sofort verworfen. Wenn keine Security Policy anwendbar ist, wird das IP-Paket an die nächsthöhere Verarbeitungsschicht weitergereicht.

Dieses Verhalten gilt für normale IP/IPSec-Stack-Implementierungen; spezielle „gehärtete“ Implementierungen verwerfen sofort alle Nicht-IPSec-Pakete, außer UDP-Paketen mit der Portnummer 500, die vom IKE-Protokoll benutzt werden. Standardkonform ist dieses Verhalten – allerdings mit höherem Verarbeitungsaufwand und geringerer Sicherheit – dadurch erreichbar, dass eine Sicherheitsstrategie existiert, die alle möglichen IP-Pakete umfasst.

Authentication Header ist ein IPSec-Protokoll, das außer der Datenvertraulichkeit alle in IPSec geforderten Schutzmechanismen bietet. Die detaillierte Spezifikation ist im RFC2402 nachzulesen. Die Hauptaufgabe des AH besteht in der Authentifizierung der Datenpakete und der Sicherung ihrer Integrität. Optional kann man den Schutz vor wiederholtem Senden von Datagrammen (Replay-Angriffen) aktivieren.

Der Schutz der Datenintegrität erstreckt sich über das vollständige IP-Paket, also auch über den IP-Header. Somit kann auch keinerlei Form von NAT (Network Address Translation, Umsetzung von IP-Adressen) auf Pakete angewendet werden, die mit dem Authentication Header gesichert wurden. Falls dies in bestimmten Fällen ein Problem sein sollte, muss man das ESP-Protokoll einsetzen.

Da sich einige Werte im IP-Header während des Transports ändern können, dürfen die betreffenden Felder natürlich nicht in den Prüfwert eingehen und werden von der Berechnung mit 0 bewertet. Diese fünf Felder sind: Type of Service, Flags, Fragment Offset, Time to Live und Header Checksum.

Das Authentication-Header-Protokoll wird im IP-Header durch die Protokollnummer 51 gekennzeichnet. Der eigentliche Header, der zwischen IP-Header und Nutzdaten eingefügt wird, ist recht einfach aufgebaut. Es werden auch keine Trailer oder Füllzeichen benötigt, wodurch die Pakete nicht so sehr vergrößert werden.

In IP Version 4 hat der Authentication Header meist eine Länge von 24 Byte, die sich allerdings bei Einsatz anderer optionaler HMAC-Verfahren unter Umständen vergrößern kann. Der Authentication Header setzt sich aus den im Folgenden erläuterten Feldern zusammen (vgl. Abbildung 5.8):

Dieses Feld zeigt an, welche Art von Daten auf den Authentication Header folgt. Im Tunnel-Modus ist dies immer der Wert 4 (IP). Im Transport-Modus hängt dies von den Nutzdaten ab. Dies sind meist TCP- oder UDP-Pakete.

Dieses Feld gibt die Länge des Headers an. Da, wie eingangs erwähnt, IPSec ein Protokoll aus IP Version 6 ist, erfolgen die Berechnungen auch entsprechend dem RFC2460 in Vielfachen von 64-Bit-Werten.

Dieses Feld ist für zukünftige Erweiterungen reserviert und hat den Wert 0.

Der Security Parameter Index ist eine innerhalb eines IPSec-Systems eindeutige 32-Bit-Zahl, die zusammen mit der Zieladresse des äußeren IP-Headers eindeutig eine Sicherheitsassoziation identifiziert.

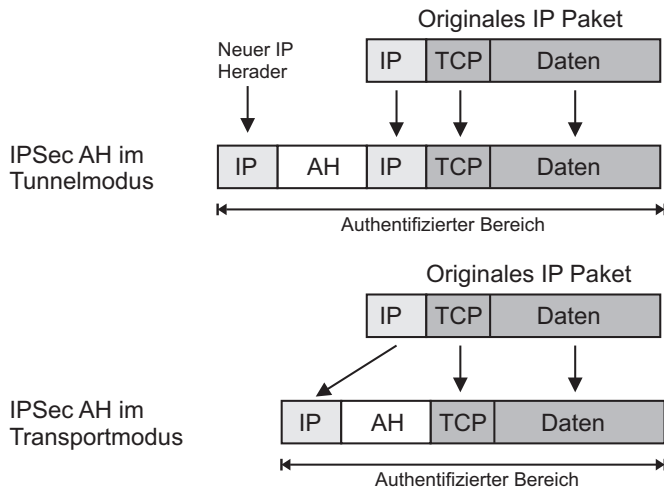


Abbildung 5.7: Das IPsec Authentication Header Protocol (AH)

In diesem Feld ist der Wert eines 32-Bit-Zählers enthalten, der bei jedem neuen IP-Paket um eins erhöht wird. Er dient auf der Empfängerseite zum Schutz vor wiederholtem Senden von Datagrammen.

Integrity Check Value (ICV), Authentication Data Hier stehen die eigentlichen Informationen zur Authentifizierung und Integritätsprüfung, der so genannte ICV (Integrity Check Value). Mit welchem Verfahren die Werte erzeugt wurden, wird nicht im AH-Protokoll festgelegt, sondern vorher vom IKE-Protokoll oder manuell spezifiziert. Vom Standard werden HMAC-MD-96 und HMAC-SHA-96 vorgeschrieben, andere Verfahren können optional implementiert werden.

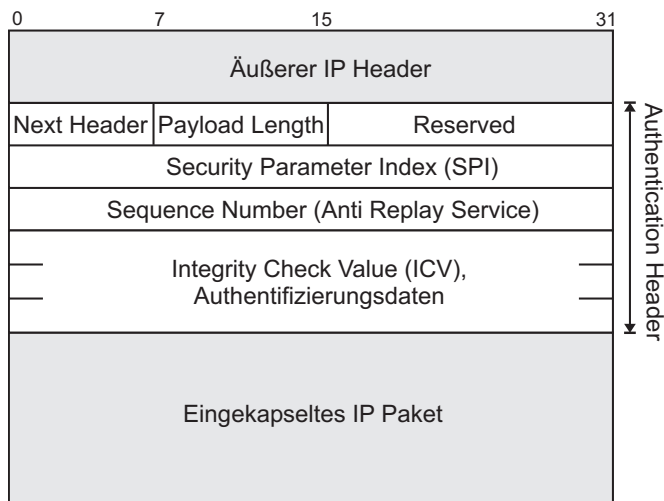


Abbildung 5.8: Das IPsec-AH-Paketformat

Die Verarbeitung ausgehender AH-Pakete

Im Transport-Modus wird der AH-Header direkt nach dem IP-Header in das IP-Paket eingefügt. Im Next-Header-Feld wird die Nummer des Protokolls eingetragen, das im Nutzdatenbereich eingekapselt ist. Das SPI-Feld erhält die Nummer aus der entsprechenden SA, ebenso wird der Wert des Sequenzzählers auf einen gültigen Wert gesetzt, also um eins gegenüber dem letzten Paket erhöht, das innerhalb dieser SA übertragen wurde. Das Protokollfeld im IP-Header wird auf 51 (AH) gesetzt.

Im Tunnel-Modus erhält das Feld Next Header den Wert 4, und der AH wird zwischen dem inneren und dem äußeren IP-Header eingefügt. Die restlichen Felder werden wie im Transportmodus behandelt. Dann wird ein neuer IP-Header mit der Senderadresse des Geräts und der Zieladresse, die in der SA festgelegt wurde, erzeugt und das Protokollfeld auf 51 gesetzt.

Die restliche Verarbeitung ist bei Tunnel- und Transport-Modus gleich. Das Paket wird mit dem in der Sicherheitsassoziation spezifizierten Authenticator authentifiziert, und der Wert wird in das Authentication-Data-Feld eingetragen. Dabei ist zu beachten, dass im Gegensatz zu ESP das vollständige IP-Paket berechnet wird, inklusive des neuen IP-Headers. Dies bedeutet, dass AH mit keiner Art von NAT (Network Address Translation) eingesetzt werden kann! Von der Berechnung des ICV werden nur die Felder ausgenommen, die sich während der Übertragung ändern können.

Als Letztes wird die Paketprüfsumme berechnet und in den IP-Header eingetragen. Nun kann das Paket verschickt werden.

Die Verarbeitung eingehender Pakete

Die Verarbeitung eingehender AH-Pakete erfolgt in umgekehrter Reihenfolge. Bevor die eigentliche Verarbeitung erfolgen kann, muss aufgrund von SPI und IP-Quell- und Zieladressen in der Security Association Database (SAD) ein entsprechender Eintrag mit den notwendigen Optionen und Parametern gefunden werden.

Anschließend wird überprüft, ob das Paket schon einmal eingetroffen ist. Hierzu wird, falls diese Option (ARS) im Empfänger aktiviert ist, der Sequenzzähler im AH-Header ausgewertet und geprüft, ob sich der Zählerwert im Empfangsfenster befindet.

Falls dem so ist, wird der HMAC des Pakets vom Empfänger berechnet und mit dem Wert im Paket verglichen. Sind beide gleich, so wurde das Paket nicht verändert, es ist also authentisch, und aufgrund des Next-Header-Feldes wird geprüft, wie das Paket weiterverarbeitet werden soll. Ist in dem Feld eine 4 (IP) eingetragen, so handelt es sich um ein Paket im Tunnel-Modus, und das eingekapselte Paket wird entpackt und der nächsthöheren Verarbeitungsschicht zugeführt. Steht in diesem Feld ein anderer Wert, dann wird das vorliegende Paket sofort zur entsprechenden Verarbeitungsschicht weitergeleitet.

5.8.3 Encapsulating Security Payload (ESP)

Das Encapsulating-Security-Payload-Protokoll bietet alle Sicherheitsfunktionen, die auch AH bietet, plus einem zusätzlichen Schutz der Datenvertraulichkeit. Um dies zu erreichen, wird ein Teil des IP-Pakets verschlüsselt und ein Header sowie ein Trailer in das Datagramm eingefügt. Datenintegritätsprüfung und Authentifizierung erstrecken sich im Gegensatz zum Authentication Header nicht über das vollständige Paket, sondern der IP-Header wird von der Berechnung ausgenommen. ESP ist ebenfalls ein IP-Protokoll, ihm wurde die Nummer 50 zugeordnet.

Ebenso wie beim Authentication Header sind die Algorithmen, die zur Integritätsprüfung und zur Verschlüsselung benutzt werden, nicht im ESP-Protokoll spezifiziert, sondern bereits vom IKE-Protokoll oder manuell festgelegt worden. ESP muss mindestens einen der beiden Sicherheitsmechanismen Verschlüsselung oder Integritätsprüfung unterstützen. Es ist also möglich, ESP mit Verschlüsselung, ESP mit Integritätsprüfung oder ESP mit Verschlüsselung und Integritätsprüfung zu betreiben, nicht jedoch ESP ohne eines dieser beiden Verfahren. Die erste Variante ohne Integritätsprüfung und Authentifizierung birgt aber gewisse Sicherheitsrisiken und sollte möglichst vermieden werden.

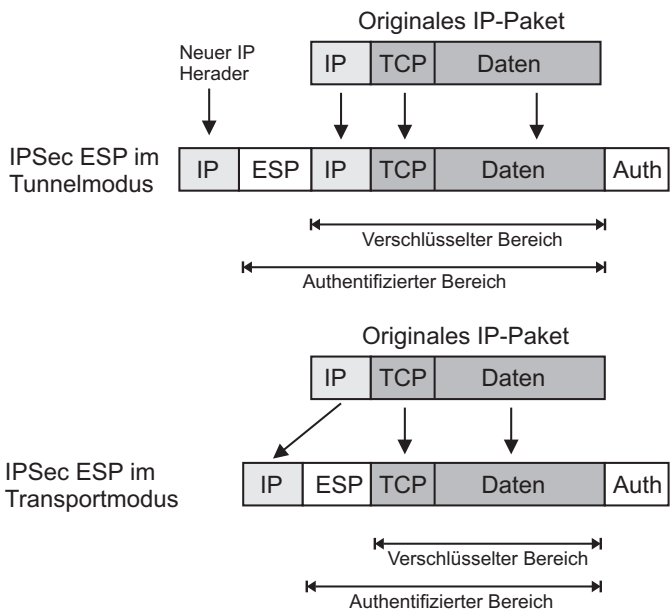


Abbildung 5.9: IPSec Encapsulating Security Payload (ESP)

Encapsulating Security Payload fügt dem originalen Datagramm unter Umständen mehr an zusätzlicher Länge hinzu als das AH-Protokoll, mindestens jedoch 4 Byte mehr.

Der ESP-Header folgt direkt auf den IP-Header, und der ESP-Trailer mit den Authentifizierungsdaten wird direkt an die Nutzdaten angehängt. Es sind folgende Felder spezifiziert (siehe Abbildung 5.10):

Security Parameter Index (SPI) Der Security Parameter Index ist eine innerhalb eines IPSec-Systems eindeutige 32-Bit-Zahl, die zusammen mit der Zieladresse des äußeren IP-Headers eindeutig eine Sicherheitsassoziation identifiziert.

Sequence Number (Anti Replay Service) In diesem Feld ist der Wert eines 32-Bit-Zählers enthalten, der bei jedem neuen IP-Paket um eins erhöht wird. Er dient auf der Empfängerseite zum Schutz vor dem wiederholten Senden von Datagrammen. In abgehenden Paketen wird der Zähler immer hochgezählt, so dass die Entscheidung, ob er benutzt wird, ausschließlich auf der Empfängerseite liegt.

Initialisation Vector Der Initialisierungsvektor ist eine 64 Bit große Zufallszahl, die für den Cipher-Block-Chaining-Modus (CBC) benötigt wird. Dieser Modus ist im RFC2405 als DES-Betriebsart vorgeschrieben.

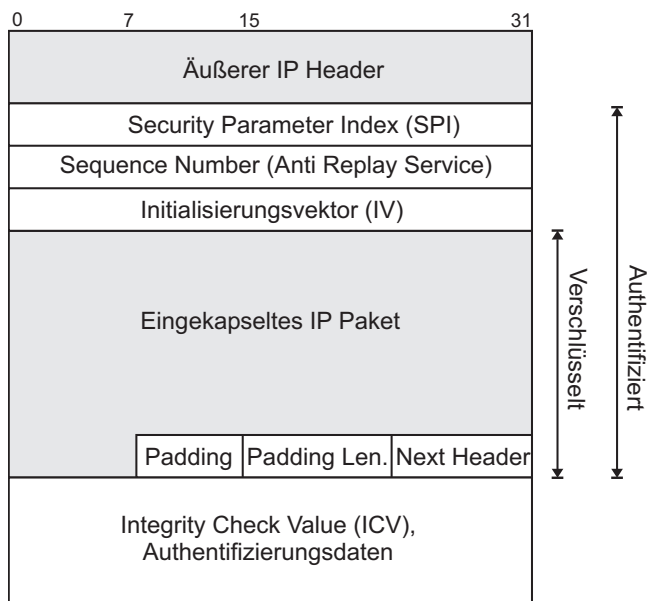


Abbildung 5.10: Das IPSec-ESP-Paketformat

Padding Das Padding (Auffüllen) wird im ESP-Protokoll für verschiedene Zwecke verwendet. Bestimmte Verschlüsselungsalgorithmen setzen voraus, dass die zu verschlüsselnden Daten ein Vielfaches einer bestimmten Blockgröße sind. Beim DES-Verfahren zum Beispiel ist diese Blockgröße 64 Bit. Nutzdaten, die kein Vielfaches davon sind, werden auf die erforderliche Länge aufgefüllt.

Eine weitere Verwendung besteht darin, auch wenn gar keine Verschlüsselung in ESP spezifiziert wurde, die beiden Felder *Padding Length* und *Next Header* an ihre richtige Position zu bringen.

In Verbindung mit einem Verschlüsselungsalgorithmus kann man durch das Padding auch die Länge der Originaldaten verschleiern und damit ESP ohne Integritätsprüfung vor so genannten Cut-and-Paste-Angriffen schützen.

In diesem Feld ist festgelegt, wie lang das Padding-Feld ist. Eine 0 zeigt an, dass kein Padding verwendet wurde.

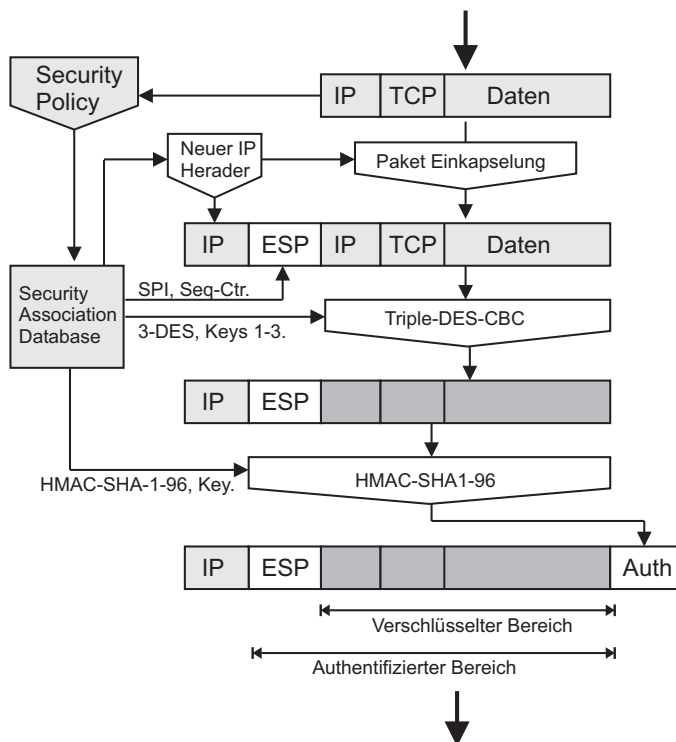


Abbildung 5.11: Die Verarbeitung eines ausgehenden IPSec-ESP-Paketes

Dieses Feld gibt an, welche Art von Daten im Nutzdatenfeld enthalten ist, im Tunnel-Modus ist dies immer der Wert 4 (IP), im Transport-Modus meist 6 (TCP) oder 17 (UDP).

Hier stehen die eigentlichen Informationen zur Authentifizierung und Integritätsprüfung, der so genannte ICV (Integrity Check Value). In ESP ist die Authentifizierung optional; falls keine ausgewählt wurde, entfällt dieses Feld.

Im Transport-Modus wird der ESP-Header direkt nach dem IP-Header in das IP-Paket eingefügt und der ESP-Trailer an das Paket angehängt. Im Next-Header-Feld wird die Nummer des Protokolls eingetragen, das im Nutzdatenbereich eingekapselt ist. Das SPI-Feld erhält die Nummer aus der entsprechenden SA, ebenso wird der Wert des Sequenzzählers auf einen gültigen Wert gesetzt, also um eins gegenüber dem letzten Paket erhöht, das innerhalb dieser SA übertragen wurde. Falls notwendig, werden Füllzeichen eingefügt und das Feld Padding Length auf den entsprechenden Wert gesetzt. Das Protokollfeld im IP-Header wird auf 50 (ESP) gesetzt.

Im Tunnel-Modus umschließen ESP-Header und -Trailer das eingekapselte IP-Paket. Das Feld Next Header erhält bei IP Version 4 den Wert 4. Die restlichen Felder werden wie im Transport-Modus behandelt. Dann wird ein neuer IP-Header mit der Senderadresse des Geräts und der Zieladresse, die in der SA festgelegt wurde, erzeugt und das Protokollfeld auf 50 gesetzt.

Die restliche Verarbeitung ist bei Tunnel- und Transport-Modus identisch. Es wird zunächst alles zwischen dem Initialisierungsvektor und dem ESP-Authentication-Data-Feld mit dem Verschlüsselungsalgorithmus verschlüsselt, der für die angewendete SA festgelegt wurde. Daraufhin wird das Paket vom ESP-Header bis zum ESP-Trailer mit dem in der Sicherheitsassoziation spezifizierten Authenticator authentifiziert und der Wert in das Authentication-Data-Feld eingetragen.

Als Letztes wird die Paketprüfsumme berechnet und in den IP-Header eingetragen. Nun kann das Paket verschickt werden.

In Abbildung 5.11 ist die Verarbeitung eines ESP-Pakets im Tunnelmodus skizziert. Die IPSec-Security-Policy prüft zuerst anhand ihrer Datenbank, ob für das zu versendende Paket eine Regel zutrifft, und liefert im Erfolgsfall einen Verweis auf einen Eintrag in der Sicherheitsassoziationsdatenbank (SAD) zurück. In dieser Datenbank sind alle Parameter und Optionen hinterlegt, die IPSec benötigt, um eine Sicherheitsassoziation aufzubauen bzw. zu unterhalten. Im Falle unseres Beispiels existiert ein entsprechender Eintrag, und es ist auch bereits eine Sicherheitsassoziation aktiv, innerhalb derer das Paket transportiert werden kann. In der SAD ist genau festgelegt, wie das zu übertragende Paket bearbeitet werden muss (Tunnelparameter, Verschlüsselungsalgorithmus, Schlüssel, Authentifizierungsalgorithmus, SPI, letzter Wert des Sequenzzählers usw.).

Die Verarbeitung eingehender Pakete erfolgt in genau umgekehrter Reihenfolge. Bevor die eigentliche Verarbeitung erfolgen kann, muss aufgrund von SPI und IP-Quell- und Zieladressen in der SAD ein entsprechender Eintrag mit den notwendigen Optionen und Parametern gefunden werden. Dann wird überprüft, ob das Paket schon einmal eingetroffen ist. Hierzu wird, falls diese Option im Empfänger aktiviert ist, der Sequenzzähler im ESP-Header ausgewertet und nachgeschaut, ob sich der Zählerwert im Empfangsfenster befindet.

Falls das Paket einmalig ist, wird der HMAC des Pakets vom Empfänger berechnet und mit dem Wert im Paket verglichen. Sind beide gleich, so wurde das Paket nicht verändert und ist authentisch.

Nachdem also festgestellt worden ist, dass das Paket zum ersten Mal eingetroffen, integer und authentisch ist, kann es der rechenintensiven Entschlüsselung zugeführt werden. Sobald das Paket entschlüsselt ist, wird aufgrund des Next-Header-Feldes geprüft, wie das Paket weiterverarbeitet werden soll. Ist in dem Feld eine 4 eingetragen, handelt es sich um ein Paket im Tunnelmodus, und das eingekapselte Paket wird entpackt und der nächsthöheren Verarbeitungsschicht zugeführt. Steht in diesem Feld ein anderer Wert, wird das vorliegende Paket sofort zur entsprechenden Verarbeitungsschicht weitergeleitet.

Die IPSec-Funktionalität kann auf verschiedene Arten implementiert werden. Dies hängt von einigen Faktoren ab, zum Beispiel davon, ob gemischte, also IP- und IPSec-Funktionalität auf dem gleichen Interface notwendig ist oder nicht.

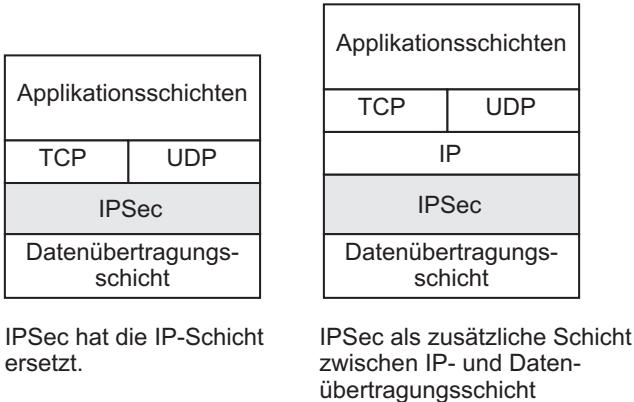


Abbildung 5.12: Die verschiedenen Arten einer IPSec-Implementierung

Betriebssystemebene, IPSec-Stack

In dieser Variante wird der ursprüngliche IP-Stack durch einen neuen Stack ersetzt, der eine zusätzliche IPSec-Funktionalität aufweist. Dies ist eine leistungsfähige Implementierung, die allerdings nur in bestimmten Systemumgebungen einsetzbar ist, z.B. in Routern oder dedizierten VPN-Gateways.

Falls es sich bei dem Gateway um ein reines VPN-System handelt, kann man auch ganz spezielle, gehärtete Stacks einsetzen, die zusätzliche Sicherheit bieten können. Diese Implementierungen verarbeiten ausschließlich diejenigen Protokolle, die für ihre Funktionalität benötigt werden. Ein spezieller IPSec-Stack würde zum Beispiel alle Pakete außer Protokoll 17 mit Portnummer 500 (IKE), Protokoll 50 (ESP) und Protokoll 51 (AH) ohne weitere Aktivitäten verwerfen (Silent Discard) und wäre dadurch vor einer ganzen Reihe von Denial-of-Service-Angriffen sicher.

5.9.2 Bump-in-the-Stack (BITS)

Insbesondere für PC-Implementierungen, in denen die IPSec-Funktionalität nicht vom Hersteller des Betriebssystems eingebaut ist, muss man eine andere Implementierung wählen. Denn anderenfalls würde man massiv in die Netzwerkfunktionalität eingreifen, und bei jedem Update des Betriebssystems würde IPSec wieder vom herstellereigenen IP-Stack ersetzt werden. Als Lösung fügt man IPSec als zusätzliche Verarbeitungsschicht zwischen IP-Ebene und Datenübertragungsschicht ein. Hier wird zwar die Netzwerkschicht zweimal durchlaufen, was nicht der leistungsfähigste Ansatz ist, aber dafür ist diese Methode sehr flexibel und sowohl vom IP-Stack als auch von der Übertragungstechnologie fast unabhängig.

5.10 Betrachtungen zur IPSec Performance

Über die so genannte Performance von IPSec ist in der letzten Zeit ein derart großer – man muss es leider wirklich so drastisch formulieren – Unsinn veröffentlicht worden, dass diesem Thema hier ein eigener Abschnitt gewidmet ist. Mit dem Begriff „Performance von IPSec“ fangen die Missverständnisse bereits an, denn IPSec ist eine Protokolldefinition; IPSec weist daher weder einen bestimmten Datendurchsatz auf, noch kennt es Verzögerungszeiten. Auch Aussagen wie die, dass man sich, anstatt IPSec einzusetzen, Gedanken über eine Hardware-Verschlüsselung machen sollte, sindbarer Unsinn, da sich IPSec und Hardware-Verschlüsselung nicht ausschließen – im Gegenteil: Eine ganze Reihe von Herstellern setzt Hardware-Bausteine zur Verschlüsselung von IPSec ein. Es wurden wohl, aus welchen Gründen auch immer, nicht gut gelungene Implementierungen von IPSec dazu benutzt, das Protokoll als solches als schlecht einzustufen – ein unzulässiger Vorgang, wie Sie im Folgenden auch sehen werden.

Ich möchte an dieser Stelle versuchen, einmal pragmatisch an das Thema heranzugehen, und zeigen, was die Performance von IPSec-Implementierungen tatsächlich beeinflusst und welchen praktischen Effekt dies haben kann. Ich habe mit Bedacht, obwohl die technische Durchführung kein Problem gewesen wäre, auf Messungen verzichtet, da ich am Sinn bestimmter Messungen erhebliche Zweifel hege und wirklich ganz selten einen Messaufbau gesehen habe, der mit einer echten Netzwerkumgebung mit all ihren verschiedenen Applikationen, Protokollen und fehlskonfigurierten Komponenten auch nur annähernd vergleichbar wäre.

Die meisten Messungen, die veröffentlicht werden, stammen von Herstellern, „unabhängigen“ Reports (die Anführungszeichen stehen hier für eine Einschränkung, denn nicht wenige solcher Vergleichstests werden von einem der Testkandidaten finanziert – der dann auch prompt gewinnt) oder Fachzeitschriften. Letztere sind oft der rettende Strohhalm, denn diese haben zumindest die Möglichkeit, wirklich unabhängig zu testen oder testen zu lassen. Aber letztendlich sind Labormessungen niemals identisch mit dem Verhalten der Systeme in der realen Welt unserer heutigen Sprach- und Datennetze.

Die Performance von IPSec-Implementierungen hängt von einer Reihe von Faktoren ab. Zuerst einmal legt die Security Policy selbst den Aufwand fest, den eine IPSec-Implementierung treiben muss, um die notwendigen Transformationen durchzuführen. Es ist offensichtlich, dass AH mit HMAC-MD5-96 im Transportmodus weitaus weniger Verarbeitungsressourcen benötigt als ESP mit Triple-DES und HMAC-SHA1-96 im Tunnelmodus – um gleich einmal zwei Extremfälle zu betrachten. Zum anderen sind natürlich die Ressourcen des Systems, auf dem IPSec implementiert ist, dafür entscheidend, ob eine ausreichende Leistung zur Verfügung steht.

Ein anderer Faktor ist die zusätzliche Paketgröße, die von IPSec erzeugt wird. Auch hier ist der Extremfaktor wieder der ESP-Tunnel-Modus mit Authentifizierung und Verschlüsselung. Der Overhead fällt hier umso mehr ins Gewicht, je kleiner die Nutzlastpakete sind. Damit wird zunehmend die Nutzbandbreite verkleinert, und eine Steigerung der Übertragungskosten ist die Folge.

Das IKE-Protokoll kann für eine IPSec-Sicherheitsassoziation auch eine Datenkomprimierung aushandeln, falls beide Gegenstellen dies unterstützen. Dies ist auch sehr wichtig, da zum einen die Pakete kleiner werden und zum anderen verschlüsselte Pakete

nicht mehr komprimierbar sind. Verschlüsselungsverfahren erzeugen nämlich ein Bitmuster, in dem praktisch keine sich wiederholenden Muster mehr vorkommen. Darauf bauen allerdings die Komprimierungsverfahren auf. Die gut gemeinte Idee, auf tieferen Schichten, z.B. auf PPP-Ebene, eine Datenkomprimierung zu aktivieren, bringt gar nichts außer der Verschwendung von Rechenleistung und Speicherplatz und, je nach eingesetztem Verfahren, sogar zusätzlicher Paketlänge. Also sollte die IPSec-Komprimierung aktiviert werden, wodurch die Daten vor der Verschlüsselung komprimiert werden. Allerdings sollte man sich in diesem Fall vor Augen halten, dass dies auch wieder ressourcenintensiv ist und die Gesamtleistung des Systems nachteilig beeinflussen kann.

Zusammenfassend kann man sagen, dass heutige moderne Rechner mit ausreichend Arbeitsspeicher und hinreichend leistungsfähigen CPUs der Pentium-Klasse mit IPSec-Clients keine Probleme haben. Manche Hersteller lassen sogar bei der Installation ihrer Client-Software eine Testroutine ablaufen, die testet, ob es sich um ältere CPUs handelt, und die Installation dann nicht mehr zulässt.

Bei IPSec-Gateways oder VPN-Remote-Access-Konzentratoren auf IPSec-Basis hängt es sehr stark vom Design des Systems ab, wie hoch seine Leistungsfähigkeit ist. Allgemein kann man sagen, dass ein speziell für Tunneling und IPSec entwickeltes System ohne spezielle Hardware-Beschleuniger durchaus Leistungen im Bereich von 50 bis 100 MBit/s bei mehreren tausend SAs mit AES-Verschlüsselung bieten kann. Viele Systeme bieten die Option, Hardware-Beschleuniger einzubauen, um die Leistung weiter zu erhöhen. Bei der Auswahl solcher Karten ist allerdings Vorsicht geboten. Einige Karten bieten nur eine extrem limitierte Funktionalität. Manche Boards verarbeiten nur einen einzigen Verschlüsselungsalgorithmus, z.B. Triple-DES, andere Verfahren wie DES, MD5, SHA1, und vor allem die IPSec-Komprimierung erfolgt weiterhin in Software. Viel besser stehen solche Boards da, welche die komplette Palette von IPSec-Transformationen in Hardware realisiert haben und damit das Muttersystem ganz entscheidend entlasten.

Manche Chips, wie zum Beispiel der HiFn 7851, unterstützen sogar Public-Key-Verfahren und bringen somit auch einen enormen Leistungsschub in das extrem rechenintensive IKE-Protokoll. Auf eine solche Funktionalität ist ganz besonders dann Wert zu legen, wenn in einem kurzen Zeitraum eine große Anzahl von Sicherheitsassoziationen aufgebaut wird und eine Authentifizierung auf Basis von digitalen Signaturen und Zertifikaten erfolgt. Dies ist üblicherweise bei Remote-Access-VPN der Fall, wenn sich zu bestimmten Stoßzeiten sehr viele Benutzer anmelden.

Bei Hardware-Beschleunigern muss man zwei Ansätze unterscheiden: einmal spezielle Erweiterungsboards, die mit speziellen Krypto-Chips ausgestattet sind, in denen die Algorithmen tatsächlich per Hardware abgebildet sind, sowie andere Boards, die mit so genannten „General-Purpose-Prozessoren“ ausgestattet sind, die lediglich die System-CPU entlasten und keine spezielle Logik zum Verarbeiten von kryptologischen Verfahren implementiert haben.

Ein anderes Qualitätskriterium für IPSec, speziell bei Sprach- oder interaktiven Videoanwendungen, ist die Verzögerungszeit (Delay) der Pakete. Hier gibt es bestimmte Grenzen, die nicht überschritten werden dürfen. Beispielsweise sollte bei Sprachanwendungen die gesamte Paketlaufzeit unter 200 ms liegen.

Hier zuerst die schlechte Nachricht: Eine Echtzeitanwendung für bestimmte Maschinensteuerungen, die eine Verzögerung von einer Millisekunde (1/1000 Sekunde) gegenüber unverschlüsseltem IP nicht mehr toleriert, ist für manche IPSec-Systeme ein Problem.

Die gute Nachricht ist die, dass man in solchen Umgebungen ohnehin selten IPSec einsetzt und alle sonstigen Daten-, Sprach- und Videoapplikationen mit Verzögerungen im Bereich einiger Millisekunden überhaupt keine Probleme haben. Normale Router, die nicht speziell für IPSec entwickelt wurden, weisen in der Regel unter günstigen Bedingungen Paketlaufzeiten im 10-Millisekunden-Bereich auf, die jedoch unter Last durchaus in den Bereich einiger hundert Millisekunden kommen.

Spezielle VPN-Systeme wie VPN-Gateways oder VPN-Router sind in der Regel wesentlich performanter, manche Systeme liegen sogar im Mikrosekundenbereich.

Diese Größenordnung relativiert sich endgültig, wenn man sich einerseits ins Gedächtnis zurückruft, welche Laufzeiten in günstigen Fällen im Internet auftreten, und andererseits berücksichtigt, dass dieser Wert in einem normalen VPN-System deutlich niedriger, in der Regel im Bereich einer Millisekunde wird.

Zusammenfassend ist zu sagen, dass gute Implementierungen von IPSec sowohl im traditionellen Daten- als auch im Sprach- und Videoumfeld eingesetzt werden können. Der Durchsatz ist sehr gut, und die Verzögerungen sind sehr klein, in jedem Fall aber um Größenordnungen kleiner als die durch die Router produzierten Delays. Man darf sich nicht durch einige wenige schwarze Schafe und IPSec-Implementierungen auf Standardroutern, die niemals für Verschlüsselungssoftware ausgelegt wurden, täuschen lassen. Ein gutes, auf Verschlüsselung und Tunneling optimiertes VPN-System, und davon gibt es mittlerweile einige, lässt in Bezug auf Performance heute keine Wünsche mehr offen – und wem das immer noch nicht ausreicht, der kann in vielen Systemen Hardware-Beschleuniger nachrüsten.

5.11 Zukünftige Entwicklungen

Die Zukunft von IPSec kann man als vielversprechend bezeichnen. Dies hat zwei Gründe. Zum einen gibt es momentan kein ernst zu nehmendes, konkurrierendes Protokoll, und andererseits ist IPSec im IP-Protokoll der Zukunft, IP Version 6, bereits integriert. Genau genommen wurde es ja von diesem neuen Protokoll auf das alte IP portiert. Die Industrie tut ihr Eigenes dazu, indem IPSec in die Betriebssysteme und das Netzwerkequipment fast aller Hersteller Einzug gehalten hat. Auch neue Algorithmen wie AES oder RIPEMD-160 sind in IPSec berücksichtigt.

SSL ist keine Konkurrenz zu IPSec, sondern eine Ergänzung auf Applikationsebene mit vergleichbaren Sicherheitsfunktionen.

Allerdings ist IP-Security längst nicht überall einsetzbar. Insbesondere im Bereich von IP-Multicasting muss man vorerst darauf verzichten. Dies hängt damit zusammen, dass IPSec von seiner Struktur her ein reines Punkt-zu-Punkt-Protokoll ist. Beim Multicast kommuniziert ein IP-System jedoch mit einer Gruppe von IP-Systemen, die unter einer so genannten Multicast-Adresse erreichbar sind. Hier kann man weder IPSec noch IKE einsetzen.

Eine der wichtigsten Entwicklungen im IPSec-Umfeld sind daher die Aktivitäten im Bereich der Multicasting-Arbeitsgruppe der Internet Engineering Task Force. Hier ist allerdings noch einiges an Entwicklungsarbeit notwendig, bis man zu brauchbaren Lösungsvorschlägen kommen wird.

Weiterhin in Arbeit sind Standards für IPSec Remote Access, die automatische Konfiguration von IPSec, IPSec-Roaming und eine neue Version von IKE.

6 Das IKE-Protokoll

Im vorangegangenen Kapitel zum Thema IP-Security wurde davon ausgegangen, dass die für die verschiedenen kryptografischen Algorithmen benötigten Schlüssel bereits dort sind, wo sie benötigt werden. Beide Gegenstellen einer IPSec-Sicherheitsassoziation teilen sich einen gleichen, symmetrischen Schlüssel, mit dem eine Seite ver- und die andere Seite wieder entschlüsseln kann. Im Kapitel über Sicherheitstechnologie wurde außerdem erwähnt, dass die Lebensdauer der Schlüssel nicht zu hoch sein soll, sie also regelmäßig erneuert werden sollen.

6.1 Das Henne-Ei-Problem

Jetzt fragt man sich natürlich, wie das im Netzwerkumfeld zu bewerkstelligen ist, wenn beide Partner einer Sicherheitsassoziation (SA) über ein physikalisches Medium, nennen wir es im Weiteren einfach Leitung, miteinander verbunden sind. Über diese Leitung kann man die Schlüssel nicht austauschen, denn sie ist ja nicht gesichert, da noch keine Schlüssel auf beiden Seiten vorliegen. Diese müssen erst ausgetauscht werden. Das typische Henne-Ei-Problem: Was war zuerst da?

Ein Ausweg, der lange Zeit der einzige war, bestand darin, die Schlüssel über ein anderes, sicheres Medium zu den Gegenstellen zu transportieren. Die manuelle Schlüsselverteilung, also der Transport des oder der Schlüssel auf Papier oder Datenträger und die manuelle Hinterlegung auf den Sicherheitssystemen, ist ein Beispiel hierfür. Diese Verfahren haben jedoch alle einen Nachteil: Sie erhöhen meist die Komplexität und damit auch meist die Fehleranfälligkeit und in der Folge die Unsicherheit des gesamten Systems. Lange Zeit wurde nach einem anderen Weg gesucht, von dem man sich aber nie so recht vorstellen konnte, wie er funktionieren würde.

Dieser andere, neue Weg zur Schlüsselverteilung ist erst seit Mitte der 70er-Jahre bekannt, seit Whitfield Diffie und Martin Hellman mit ihrer die Kryptografie revolutionierenden Arbeit eine Lösung des Schlüsseltausch-Problems präsentierten, die auch nach ihnen benannt wurde. Das Diffie-Hellman-Verfahren ist ein auf Methoden der Public-Key-Kryptografie beruhender Algorithmus zum sicheren Erzeugen von Schlüsseln durch die Übertragung bestimmter Informationen über ein unsicheres Medium. Auf Basis dieses Verfahrens wurde eine Reihe von Schlüsseltauschprotokollen entwickelt, die zum Teil auch die Grundlage des Internet-Key-Exchange-Protokolls (IKE) bilden, das zum Erzeugen von IPSec-Sicherheitsassoziationen eingesetzt wird.

Das IKE-Protokoll ist eine Instanziierung von ISAKMP, dem Internet Security Association and Key Management Protocol, und benutzt Austauschschritte, die im Oakley-Protokoll spezifiziert werden. IKE ist kein auf IPSec festgelegtes Protokoll, sondern wird durch Spezifikationen in der IPSec Domain of Interpretation (IPSec-DOI) im RFC2407 für

den Einsatz mit IP-Security mit entsprechenden Datenstrukturen, Konstanten und Funktionen versehen. Andere DOIs können IKE auch für andere Protokolle nutzbar machen, die einen sicheren Schlüsseltausch benötigen.

IKE benutzt, ebenso wie IPSec, das Konzept der Sicherheitsassoziationen (SA), die sich allerdings von den IPSec-SAs unterscheiden, unter anderem da sie bidirektional sind und zwischen Initiator und Responder unterscheiden, je nachdem, von welcher Seite IKE gestartet wurde. IKE kann in den verschiedensten Modi und Kombinationen mit etlichen Authentifizierungsverfahren arbeiten, es kann mehrere SAs für IPSec in einem Austausch erzeugen und verschiedene Sicherheitsstufen implementieren.

Diese Flexibilität von IKE macht das Protokoll allerdings auch sehr komplex und langwierig zu implementieren. IKE, oder besser gesagt dessen praktische Implementierung, ist auch sehr oft der Auslöser, wenn IPSec-Systeme unterschiedlicher Hersteller keine Sicherheitsassoziation miteinander erzeugen können. Aus diesem Grund ist dieses Kapitel sehr ausführlich angelegt, um Ihnen die Möglichkeit zu geben, bei Problemen die Meldungen der jeweiligen Logfiles richtig zu interpretieren und die notwendigen Maßnahmen einzuleiten. Außerdem ist das IKE-Protokoll äußerst sicherheitskritisch, denn von ihm hängen alle später eingesetzten Dienste und die von diesen verwendeten Schlüssel ab.

6.2 ISAKMP

Das Internet Security Association and Key Management Protocol ist ein Regelwerk, welches das Verhalten der beteiligten Gegenstellen in den verschiedenen Phasen der Verbindungsverhandlung festlegt. Es legt die Paketformate der auszutauschenden Pakete fest, definiert bestimmte Funktionen und legt die Übergänge zwischen den verschiedenen Zuständen einer ISAKMP-Aushandlung fest. Aber es legt nicht fest, wie dies zu geschehen hat, und ist daher auch nicht unmittelbar in Form eines Protokolls zu implementieren. Dies bleibt den Dokumenten zu IKE, Oakley und der IPSec-DOI vorbehalten.

Im Internet Security Association and Key Management Protocol werden verschiedene Austauschvorgänge beschrieben, innerhalb derer der Initiator einer Verbindung mit dem Responder bestimmte Nachrichten austauscht. Anzahl, Art und zeitliche Abfolge der Nachrichten sowie die benötigten Informationen in diesen Paketen hängen von der Art des Austauschvorgangs ab.

ISAKMP ist nicht an bestimmte kryptografische Protokolle oder Algorithmen gebunden, ebenso wenig an IPSec oder bestimmte Schlüsseltauschprotokolle. Es bietet vielmehr eine flexible Möglichkeit, Sicherheitsassoziationen aufzubauen, und erlaubt es, die verschiedensten Techniken dafür einzusetzen. Damit bekommt man auch eine elegante Möglichkeit geboten, im Zuge der technologischen Entwicklungen neue Verschlüsselungsverfahren oder Schlüsseltauschprotokolle zu implementieren, ohne das zugrunde liegende Protokoll verändern zu müssen.

Das Internet Security Association and Key Management Protocol bietet innerhalb seiner Funktion zur Erzeugung von Sicherheitsassoziationen für ISAKMP und IPSec auch Schutz vor einer ganzen Reihe von Angriffen, insbesondere vor Denial-of-Service-Angriffen, dem Stehlen von Verbindungen und Man-in-the-Middle-Angriffen.

ISAKMP spezifiziert Nachrichten, die zwischen Initiator und Responder ausgetauscht werden, und legt ihre Formate fest. Diese Nachrichten werden in heute üblichen Implementierungen in UDP-Pakete mit der Quell- und Zielportnummer 500 eingepackt. Es sind zwar optional auch weitere Kombinationen von Protokoll- und Portnummern möglich, jedoch müssen alle Implementierungen den UDP-Port 500 unterstützen.

ISAKMP regelt die Erzeugung von Austauschvorgängen und den dafür benötigten Nachrichten zum Erzeugen, Konfigurieren und Löschen von Sicherheitsassoziationen. Weiterhin wird der Austausch von Werten beschrieben, die von den Schlüsseltauschprotokollen benötigt werden. Diese Protokolle selbst sind aber nicht Gegenstand von ISAKMP, sondern sie benutzen nur dessen Nachrichten. Es werden fünf verschiedene Arten von Austauschvorgängen definiert, die verschiedene Ziele verfolgen und unterschiedlich starke Schutzmechanismen implementiert haben. Darin ist innerhalb eines gewissen Rahmens präzise festgelegt, in welcher Abfolge die Nachrichten zu übertragen sind, wie sie verarbeitet werden müssen, welche Art von Nutzdaten enthalten sein sollen und welche optional sind. Die Sicherheitsstrategie von IKE und IPSec hat ebenfalls noch einen Einfluss auf das Verhalten eines Austauschvorgangs, speziell darauf, wann er aufgrund bestimmter kritisch erscheinender Zustände abubrechen ist.

6.2.1 Die Sicherheit von ISAKMP

ISAKMP ist aufgrund der Tatsache, dass es eine Sicherheitsassoziation für Sicherheitsprotokolle wie IP-Security aushandelt, natürlich als äußerst sicherheitskritisch anzusehen. Denn falls sich in dieser Phase bereits Ansatzpunkte für Angriffe bieten, sind die nachfolgenden Sicherheitsassoziationen auch betroffen. Denn selbst der sicherste Verschlüsselungsalgorithmus ist wirkungslos, wenn sein Schlüssel bekannt ist.

Aus diesem Grund wurde eine ganze Reihe von Schutzmechanismen für das Internet Security Association and Key Management Protocol festgelegt. Diese sind aber nicht mit dem zu verwechseln, was ISAKMP für die folgenden Sicherheitsassoziationen der Sicherheitsprotokolle aushandelt.

Authentifizierung

Ein ganz wesentlicher Punkt in diesem Zusammenhang ist die gegenseitige Authentifizierung der beiden Gegenstellen. Hier muss man sich aber noch einmal vor Augen halten, dass es sich hierbei keinesfalls um eine Benutzer-Authentifizierung handelt, sondern um eine gegenseitige Identifikation auf der Netzwerkebene. Diese kann zwar mit einer Benutzer-Authentifizierung kombiniert werden, falls es sich bei einer Gegenstelle um einen Host handelt, dessen spezifische Informationen zur Identifikation des Systems vom Benutzer selbst stammen, beispielsweise ein Passwort (Pre-Shared Secret) oder ein digitales Zertifikat der betreffenden Person. Aber im Kontext von ISAKMP bzw. IKE, das, wie im Folgenden beschrieben, dessen Austauschvorgänge benutzt, werden wir von einer Authentifizierung der Endsysteme ausgehen und uns nicht damit beschäftigen, wie die notwendigen, geheimen Informationen zum Nachweis der Identität zur Verfügung gestellt wurden.

Als Mechanismen zur Authentifizierung dienen im Wesentlichen Methoden, die auf Basis von Einwegfunktionen Zufallszahlen in Verbindung mit geheimen Informationen zur Identität der betreffenden Gegenstelle in einen Wert transformieren, der über ein unsicheres Medium zur Gegenstelle übertragen werden kann. Die Gegenstelle berechnet ebenfalls diesen Wert und vergleicht beide. Im Fall einer Übereinstimmung ist die Gegenseite authentisch. Als Funktion wird in der Regel die Hash-Funktion eingesetzt, die in den ersten Nachrichten innerhalb eines Austauschvorgangs vereinbart wurde.

Da diesen Verfahren je nach Qualität ihrer Implementierung nur eine begrenzte Sicherheit bescheinigt wird, werden von der Sicherheitsgruppe der Internet Engineering Task Force (IETF) solche Verfahren bevorzugt, die auf digitalen Signaturen unter Verwendung von Public-Key-Algorithmen und digitalen Zertifikaten basieren. Allerdings wird dadurch eine erhebliche zusätzliche Komplexität eingeführt, die durch die erforderliche Verwaltung der öffentlichen Schlüssel oder Zertifikate bedingt ist. In vielen Organisationen ist die dazu notwendige Infrastruktur in Form von Schlüsselverwaltungssystemen (PKI) oder Verzeichnisdiensten noch nicht oder nicht in ausreichendem Maße implementiert.

Als Folge davon sieht die Realität der Authentifizierung im IP-Security- und ISAKMP-Umfeld auch heute noch größtenteils so aus, dass sie nach wie vor auf so genannten Pre-Shared Keys, also de facto einem Passwort in Verbindung mit einem Hash-Verfahren, beruht. Nichtsdestotrotz fordert der Standard zwingend, dass auch eine starke Authentifizierung implementiert werden muss. Hierzu wird die Benutzung eines Algorithmus zur Erzeugung von digitalen Signaturen vorgeschrieben. Dieser ist zwar nicht spezifiziert, jedoch findet man in fast allen Implementierungen das RSA-Verfahren, dessen Patentschutz mittlerweile abgelaufen ist, oder den Digitalen Signaturstandard (DSS) vor.

Aber man kann, insbesondere bei IPSec-Gateways, auch die Authentifizierung per Pre-Shared Key relativ sicher machen. Denn hier kann man entsprechend lange und nicht per Wörterbuchangriff zu erratende hexadezimale Zahlen eingeben. In IPSec-Host-Implementierungen, in denen der Pre-Shared Key ein Kennwort ist, das der Benutzer im Gedächtnis behalten muss, ist dies nicht möglich, hier bieten sich leider alle bekannten Möglichkeiten von Angriffen auf Passwörter.

An dieser Stelle sei auch ein wichtiger Sicherheitstipp angebracht: Viele IPSec-Host-Implementierungen bieten optional eine starke Authentifizierung auf Basis von Token-Karten mit einem Einmal-Code an. Diese hat aber keinen Einfluss auf die Authentifizierung in ISAKMP, IPSec-ESP oder IPSec-AH! Denn diese Form des Identitätsnachweises eines Benutzers erfolgt in vernünftigen Implementierungen unter dem Schutz einer bereits etablierten Sicherheitsassoziation, die vorher auf Basis von Pre-Shared Keys erzeugt wurde. Nachfolgende SAs benutzen ohnehin Werte, die auf dem Schlüsseltausch basieren. Also authentifizieren, wie gesagt, solche Systeme ausschließlich den Benutzer. Die einzige Schnittstelle zu IKE besteht darin, dass bei einer fehlgeschlagenen Authentifizierung die Sicherheitsassoziation gelöscht und IKE terminiert wird.

Kryptografie und Schlüsseltausch

Die Kryptografie ist die Stütze des Internet Security Association and Key Management Protocols, denn sehr viele seiner Funktionen basieren auf den Entwicklungen und Technologien dieser Wissenschaft. Insbesondere die Erzeugung von symmetrischen Schlüsseln zur Verwendung mit Verschlüsselungsverfahren oder Algorithmen zur Integritätsprüfung und Authentifizierung basiert auf der Public-Key-Kryptografie.

Mit der Public-Key-Kryptografie bieten sich zwei verschiedene Möglichkeiten des Schlüsseltauschs an: Man kann den Schlüssel selbst mit einem Algorithmus wie dem RSA-Verfahren verschlüsseln und zur Gegenstelle übertragen, oder man kann mit einem speziellen Verfahren, wie dem von Diffie-Hellman, den Schlüssel auf beiden Seiten erzeugen. Man kann also zwischen der Schlüsselübertragung und der Schlüsselerzeugung wählen. Im ISAKMP-Standard wird, zugegebenermaßen etwas schwammig formuliert, lediglich ein authentifizierter Schlüsseltausch vorgeschrieben. IKE schlägt die Methode der Schlüsselerzeugung per Oakley-Protokoll mit dem Diffie-Hellman-Verfahren vor, und praktisch alle Implementierungen richten sich auch danach.

Die Authentifizierung des Schlüsseltauschs in ISAKMP ist unabdingbar, denn Public-Key-Algorithmen wie RSA oder Diffie-Hellman selbst bieten keinerlei Schutz gegen so genannte Man-in-the-Middle-Angriffe, bei denen sich ein Angreifer in die Kommunikation einschaltet und den beiden Gegenstellen jeweils die andere vorspiegelt. Die Authentifizierung kann direkt während des Schlüsseltauschs oder unmittelbar danach erfolgen, sollte aber zum Schutz vor Denial-of-Service-Angriffen rechenintensiven Funktionen stets vorangestellt sein.

Die Behandlung von Schlüsseln innerhalb von Netzwerk-Sicherheitsprotokollen wie IKE und IP-Security unterscheidet sich in einem entscheidenden Punkt von anderen Anwendungen. Während verschlüsselte Daten verschiedener Applikationen teilweise Jahre oder Jahrzehnte geheim bleiben müssen und nach oder innerhalb dieser Zeitspanne zur weiteren Verwendung entschlüsselt werden müssen, liegen im Netzwerkbereich ganz andere Voraussetzungen vor. Die Schlüssel werden dort nur so lange benötigt, bis das letzte übertragene Paket innerhalb der Lebensdauer einer Sicherheitsassoziation vom Empfänger entschlüsselt wurde. Dann kann oder – um es ganz deutlich zu sagen – muss der Schlüssel zuverlässig vernichtet werden, denn er wird definitiv nicht mehr benötigt.

Als noch kritischer ist die Behandlung der privaten Komponenten eines Schlüsseltauschverfahrens wie Diffie-Hellman einzustufen. Denn hier wird der private, auf zufälliger Basis erzeugte Schlüssel nur so lange benötigt, bis beide Seiten daraus ihren symmetrischen Hauptschlüssel berechnet haben. Die erforderliche Lebensdauer liegt somit maximal im einstelligen Sekundenbereich. Eine längere Speicherung birgt ein erhebliches Sicherheitsrisiko, da nur ein einziger bekannt gewordener privater Schlüssel auf einer der beiden Seiten einer Sicherheitsassoziation die komplette Kommunikation kompromittiert, d. h., dass Vertraulichkeit, Integrität und Authentizität der übertragenen Pakete nicht mehr gewährleistet sind. Aus diesem Grund muss die private Diffie-Hellman-Komponente sofort nach der Erzeugung des symmetrischen Schlüssels sicher vernichtet werden. In der Regel erfolgt dies durch mehrfaches Überschreiben und die anschließende Freigabe des benutzten Speicherbereichs.

Schutz vor Denial-of-Service-Angriffen

Die schlechte Nachricht zuerst: Ein absoluter Schutz gegen Denial-of-Service-Angriffe ist in keinem Übertragungssystem möglich. Das einfache Fluten von öffentlichen Schnittstellen mit Datenpaketen ist immer eine, wenn auch brachiale und mittlerweile meist mit einer empfindlichen Strafe für den Angreifer endende, Möglichkeit.

Die gute Nachricht ist die, dass es gegen die subtileren Angriffe eine ganze Reihe brauchbarer Schutzmechanismen gibt. In ISAKMP basiert der Schutz auf so genannten Anti-

Clogging-Tokens (ACT), schlicht Cookies genannt. Mit Hilfe dieser Technik kann man wirkungsvoll verhindern, dass ein Angreifer mit gefälschten IP-Absenderadressen an einer ISAKMP-Aushandlung teilnimmt und die Gegenstellen zu ebenso umfangreichen wie sinnlosen Berechnungen und Operationen veranlasst.

Schutz vor dem Stehlen von Verbindungen

Ein Angreifer, der eine Verbindung stehlen will, indem er Vermittlungssysteme entsprechend manipuliert, um anstelle einer der beiden Gegenstellen weiter an ISAKMP-Aushandlungen teilzunehmen, wird dadurch abgeblockt, dass die Authentifizierung mit der Aushandlung der Sicherheitsassoziation und des Schlüsselaustauschs verbunden ist.

Schutz vor Man-in-the-Middle-Angriffen

Ein Man-in-the-Middle-Angriff auf eine ISAKMP-Aushandlung würde versuchen, bestimmte Nachrichten zu ändern oder zu löschen. In Abbildung 6.1 sehen Sie einen vereinfacht dargestellten Angriff auf die Aushandlung einer Sicherheitsassoziation in ISAKMP. Der Initiator möchte in diesem Beispiel mit dem Responder wahlweise das ESP-Protokoll ohne, mit DES- oder mit Triple-DES-Verschlüsselung vereinbaren. Er würde, wie in Abbildung 6.1 gezeigt, eine Nachricht mit dem entsprechenden Vorschlag abschicken, und der Responder, dessen strenge Sicherheitsstrategie in unserem Fall nur eine starke Verschlüsselung zulässt, würde aus diesem Angebot den Vorschlag ESP-3-DES auswählen.

Der Angreifer, unser Man-in-the-Middle, schaltet sich nun in die Verbindung ein und versucht, dem Initiator den originalen Responder vorzutäuschen und umgekehrt. Er manipuliert die Antwort des Responders dahingehend, dass er aus dem Vorschlag des Initiators keine Verschlüsselung (ESP-NUL) auswählt, obwohl der Responder in Wirklichkeit 3-DES selektiert hat. Dann würde der Initiator im Weiteren mit dem Man-in-the-Middle unverschlüsselt kommunizieren, der originale Responder würde aber wie von ihm gefordert mit Triple-DES-Verschlüsselung arbeiten.

Beim Man-in-the-Middle selbst und auf der Strecke zwischen ihm und dem Initiator wären die Daten unverschlüsselt und somit nicht mehr vertraulich. Die Gegenstellen selbst glauben aber jeweils, mit dem originalen Partner zu kommunizieren und sich auf eine beiderseitig akzeptable Protection Suite geeinigt zu haben.

Gegen diese Art von Angriff richtet sich gleich eine ganze Menge von Maßnahmen in ISAKMP. Zum einen sind das die bereits beschriebenen Maßnahmen gegen das Stehlen von Verbindungen sowie die Bindung der Austauschvorgänge an eine Authentifizierung. Die Erzeugung der Cookies basiert unter anderem auf der aktuellen Zeit, so dass hierdurch auch gleichzeitig ein Schutz gegen wiederholtes Senden gewährleistet wird. Andere Angriffe, z.B. auf spezielle Protokolle wie TCP, werden durch den Einsatz von UDP vermieden. Der sichere Transport von UDP-Paketen wird durch geeignete Maßnahmen im ISAKMP erreicht, so dass der Verzicht auf das verbindungsorientierte TCP keine Probleme mit sich bringt.

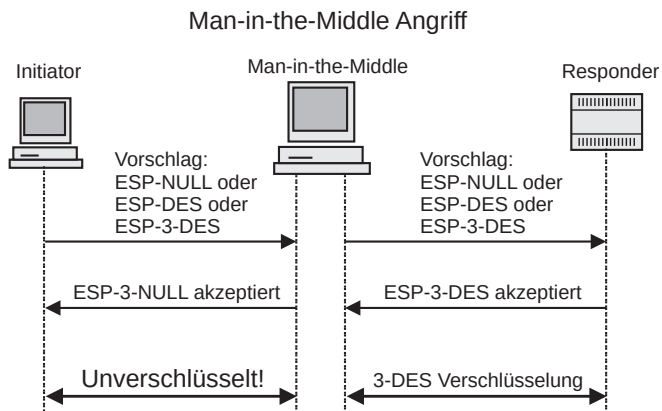
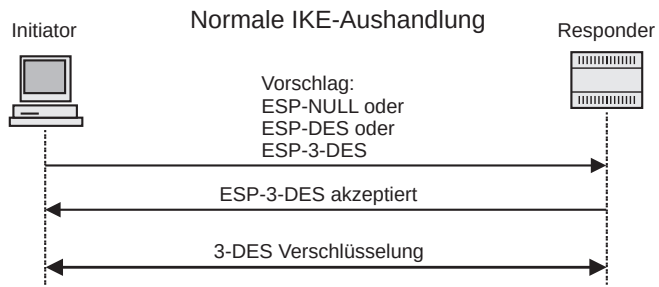


Abbildung 6.1: Ein (theoretischer) Man-in-the-Middle-Angriff auf IKE

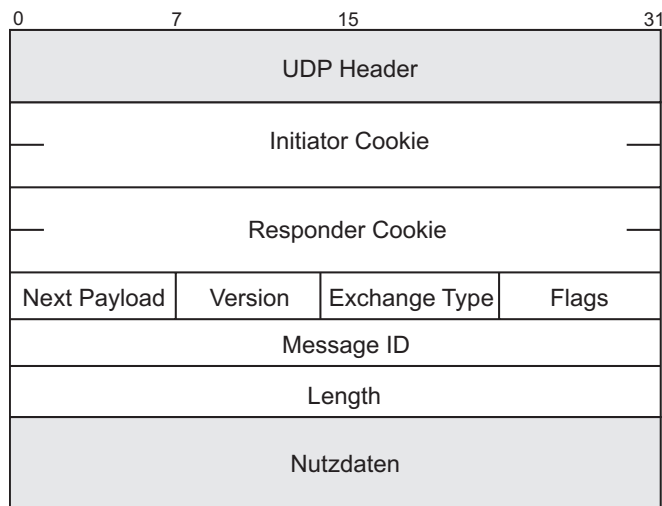


Abbildung 6.2: Der ISAKMP folgt direkt auf den UDP-Header.

6.2.2 Der ISAKMP-Header

In Abbildung 6.2 sehen Sie das Format des ISAKMP-Headers. Die Felder haben folgende Bedeutung:

Cookies

Die ersten 16 Byte dieses Headers werden von so genannten Cookies, je eines vom Initiator und eines vom Responder, belegt. Diese Cookies dienen vor allem zum Schutz vor Replay-Angriffen und Angriffen mit gefälschten Absenderadressen. Gebildet werden diese Werte aufgrund von Informationen, die nur dem Erzeuger bekannt sind, sowie einem Zeitstempel, der als Parameter für eine geeignete Funktion dient, um den Wert zu berechnen.

Es gibt keine feste Vorschrift, jedoch wird empfohlen, eine schnelle Hash-Funktion wie MD5 zu benutzen und einen Hash-Wert über die IP-Adressen, die Portnummern, eine lokal erzeugte, geheime Zufallszahl, die aktuelle Zeit und das Datum zu bilden. Somit ist nämlich garantiert, dass Informationen in die Cookies mit eingehen, die auf spezifischen Eigenschaften der Gegenstelle und der Verbindung beruhen, aber durch den Einfluss der aktuellen Zeit innerhalb ihrer Abfolge garantiert unterschiedlich sind. Die beiden Cookies zusammen bilden innerhalb der Verbindung zwischen den beiden IP-Endsystemen einen eindeutigen, 128 Bit großen Wert und dienen zusätzlich zu ihrer Sicherheitsfunktion auch als ISAKMP-Security-Parameter-Index (SPI).

Die Art der Informationen, die auf den ISAKMP-Header folgen, wird durch bestimmte vordefinierte Typennummern festgelegt. Folgende, im Weiteren näher beschriebene Nutzdaten sind spezifiziert:

- Keine Nutzdaten enthalten (None)

- Sicherheitsassoziation

- Proposal

- Transform

- Key Exchange

- Identification

- Certificate

- Certificate Request

- Hash

- Signature

- Nonce

- Notification

- Delete

- Vendor ID

- NAT Discovery (RFC3947)

- NAT Original Address (RFC3947)

Die Art der Informationen bestimmt auch deren Struktur und eventuell deren Länge, sofern darin keine variabel langen Werte enthalten sind.

Version (Major Version, Minor Version)

Die Version des Protokolls wird in einem Haupt- und Nebefeld festgelegt. Die Implementierungen sollten niemals Pakete verarbeiten, bei denen ein Feld oder beide Felder höhere Versionsnummern enthalten als sie selbst.

Exchange Type

Der Wert dieses Feldes legt fest, um welche Art von Austausch es sich bei diesem Paket handelt. Die Arten legen unter anderem implizit fest, wie viele Austauschoperationen durchgeführt werden, welche Nutzdaten im Paket folgen, welche Zustandsänderungen nach diesen Operationen eintreten und was sie in Summe bewirken sollen. Folgende Exchange-Typen sind in ISAKMP definiert:

- None (Keine Operation)

- Base Exchange

- Identity Protection Exchange (daraus wird der IKE Main Mode abgeleitet)

- Authentication Only Exchange

- Aggressive Exchange

- Informational Exchange

Flags

Optionen, welche die weitere Verarbeitung des ISAKMP-Pakets steuern. Folgende drei Optionen können im Augenblick benutzt werden:

- E-Bit (Encryption-Bit)

- C-Bit (Commit-Bit)

- A-Bit (Authentication-Only-Bit)

Das Encryption-Bit wird gesetzt, wenn angezeigt werden soll, dass die Daten, die auf den Header folgen, verschlüsselt sind. Die Verschlüsselung wurde mit dem Algorithmus durchgeführt, der in der ISAKMP-Sicherheitsassoziation ausgehandelt wurde.

Das Commit Flag dient zur Synchronisierung der Schlüsselerzeugung. Man kann es benutzen, um zu verhindern, dass verschlüsselte Daten eintreffen, bevor die Sicherheitsassoziation vollständig aufgebaut wurde und beide Seiten bereit sind.

Mit gesetztem A-Bit wird angezeigt, dass ein Informationale Exchange mit Notify-Daten vorliegt, das nicht verschlüsselt ist, aber mit dem vereinbarten Authentifizierungsverfahren auf Integrität geprüft werden muss.

Message ID

Eine Identifikationsnummer für die vorliegende Nachricht. Sie wird auf Zufallsbasis erzeugt.

Length

Dieses Feld gibt die Länge der vollständigen Nachricht, also ISAKMP-Header plus Nutzdaten, in Byte an.

6.2.3 Der ISAKMP-Nutzdaten-Header

Auf den ISAKMP-Header folgen die Nutzdaten, die aus mehreren Instanzen bestehen können, die aufeinander folgen können. Die Nutzdaten bestehen aus Strukturen, die alle mit einem generischen Header beginnen.

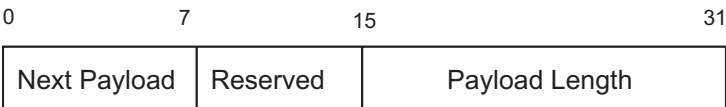


Abbildung 6.3: Der generische ISAKMP-Nutzdaten-Header

In diesem Header sind die für alle Arten von Nutzdaten gültigen Informationen enthalten:

Next Payload Dieses Feld zeigt an, ob nach der Nutzdatenstruktur noch eine weitere folgt und welcher Art sie ist. Ist dieses Feld 0, dann folgt nichts mehr nach. Der Typ der ersten Struktur wird im ISAKMP-Header festgelegt.

Reserved Dieses Feld ist für spätere Erweiterungen reserviert.

Payload Length Die vollständige Länge der vorliegenden Nutzlast, inklusive Header.

6.3 ISAKMP-Nutzdaten

Dem generischen Header folgen die Datenstrukturen, in denen die Nutzlast, also der Inhalt der Nachricht, enthalten ist. Für jeden der verschiedenen Nutzlasttypen gibt es ein spezielles Format zur Repräsentation der zu übertragenden Informationen.

6.3.1 Security Association Payload

Diese Datenstruktur wird benutzt, um Sicherheitsattribute auszuhandeln und festzulegen, in welcher Umgebung (DOI) die Aushandlung stattfindet.

Abbildung 6.4 zeigt das Datenformat. Die Felder haben folgende Bedeutung:

Domain of Interpretation (DOI)

Dieser Wert gibt an, in welcher Umgebung die Aushandlung erfolgt. 0 bedeutet, dass eine ISAKMP-SA ausgehandelt wird, der Wert 1 steht für IPsec.

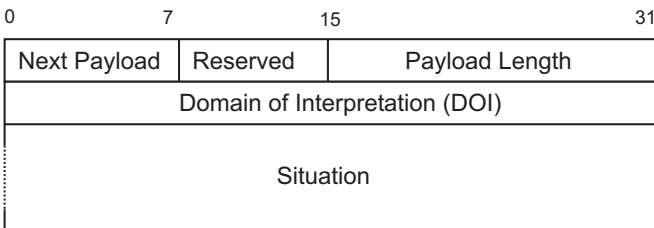


Abbildung 6.4: Die SA Payload

Situation

Dieses Feld hängt von der jeweiligen DOI ab und beschreibt die Situation, in der die Aus-
handlung stattfindet.

6.3.2 Proposal Payload

In dieser Struktur sind die Informationen enthalten, mit denen eine Sicherheitsassozia-
tion zwischen Initiator und Responder ausgehandelt wird. Diese so genannten Proposals
enthalten die Sicherheitsfunktionen, die benutzt werden sollen, um die notwendigen
Dienste zur Verfügung zu stellen.

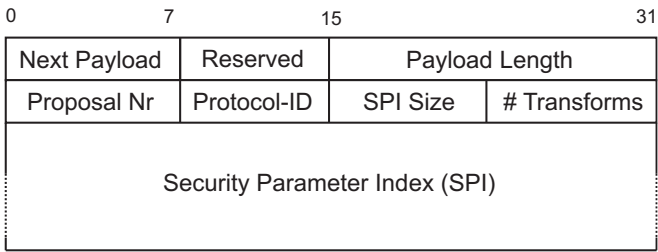


Abbildung 6.5: Die Proposal Payload

In der Regel werden in einem Austausch mehrere Vorschläge gemacht, die logisch ver-
knüpft werden können. Die Felder haben folgende Bedeutung:

- Proposal Nr** Die Nummer des aktuellen Vorschlags.
- Protocol-ID** Die Protokoll-ID der gewünschten Sicherheitsfunktionen. Diese Funktio-
nen sind in der Regel ESP, AH usw.
- SPI Size** Dieses Feld gibt die Länge des folgenden Security Parameter Index (SPI) in Byte an.
- No of Transforms** Anzahl der Vorschläge (Proposals), die gemacht werden.
- SPI** Hier wird der SPI der sendenden Gegenstelle eingetragen.

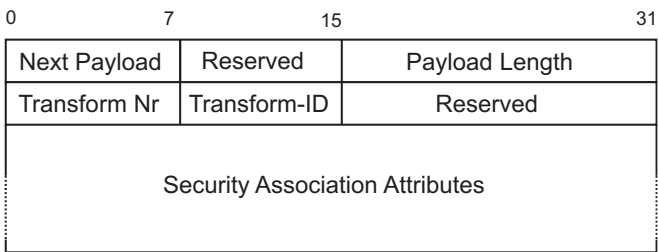


Abbildung 6.6: Das Format der Transform Payload

6.3.3 Transform Payload

In der Transform Payload werden die Informationen zu den gewünschten Sicherheitsdiensten selbst hinterlegt. Es können mehrere Transformationen in einer Nachricht vorkommen, ihre Anzahl wird in der Proposal Payload angegeben.

Transform Nr Die eindeutige Nummer der Transform Payload.

Transform-ID Die in der DOI für die entsprechenden Funktionen spezifizierten Konstanten werden in diesem Feld eingetragen.

SA Attributes Hier sind die Attribute der Sicherheitsassoziation enthalten.

6.3.4 Key Exchange Payload

Diese Nachricht enthält Informationen, die zur Erzeugung der Sitzungsschlüssel benötigt werden. Das Key-Exchange-Data-Feld enthält zum Beispiel die öffentlichen Werte eines Diffie-Hellman-Austauschs im Fall des IKE-Protokolls.

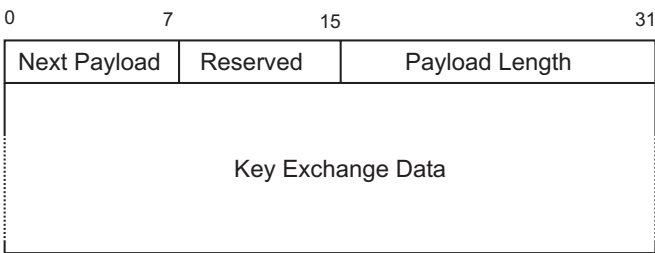


Abbildung 6.7: Die Key Exchange Payload

Das Format, in dem die Daten in diesem Feld eingetragen sind, hängt dabei von der DOI und dem ausgewählten Schlüsseltauschverfahren ab.

6.3.5 Identification Payload

In dieser Datenstruktur sind Informationen zur Identität der Gegenstelle enthalten. Diese werden benutzt, um den Partner zu identifizieren und mit diesen Informationen auch eine Authentifizierung durchführen zu können.

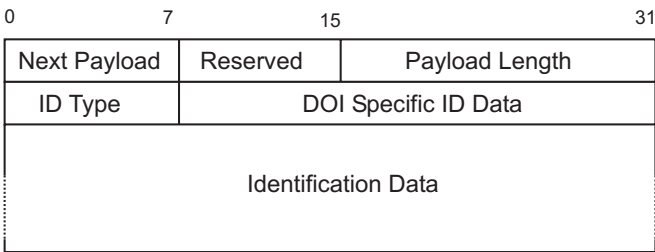


Abbildung 6.8: Die ID Payload

ID Type Dieser Wert legt fest, welcher Art die Informationen zur Identifikation sind.

DOI Specific ID Data Hier stehen DOI-spezifische Identifizierungsdaten.

Identification Data In diesem Feld sind die Identitätsinformationen enthalten.

6.3.6 Certificate Payload

Diese Nutzlast dient dazu, digitale Zertifikate zu übertragen, falls diese zur Authentifizierung benötigt werden. Eine Certificate Payload muss zu jedem Zeitpunkt einer Aushandlung akzeptiert werden. Falls keine Directory Services zur Verfügung stehen, mit denen Zertifikate verteilt werden, sollten Zertifikate in den Exchanges übertragen werden.

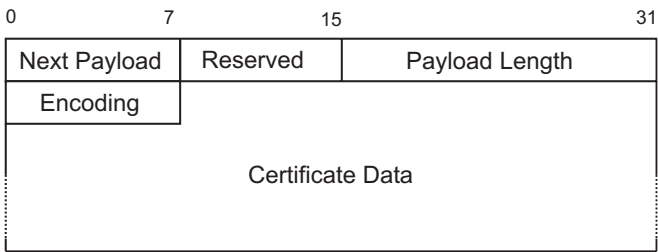


Abbildung 6.9: Die Certificate Payload

Certificate Encoding

Dieses Feld enthält Informationen über die Kodierung des Zertifikats oder über die Art der Daten im nachfolgenden Feld. Folgende Kodierungen werden unterstützt:

- PKCS#7-kodiertes X.509-Zertifikat
- PGP-Zertifikat
- DSS-signierter Schlüssel
- X.509-Zertifikat zur Signatur
- X.509-Zertifikat zum Schlüsseltausch
- Kerberos Token
- Certificate Revocation List (CRL)
- Authority Revocation List (ARL)
- SPKI-Zertifikat
- X.509-Attribut-Zertifikat

Certificate Data

Hier befindet sich das eigentliche Zertifikat oder eine Information in einem der oben angeführten Datenformate.

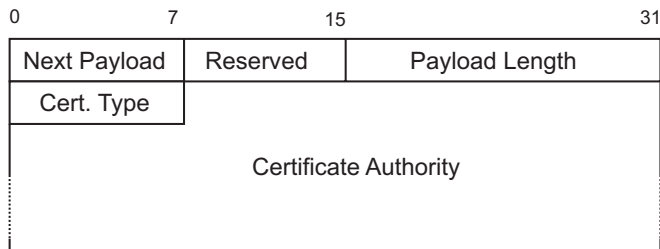


Abbildung 6.10: Die Certificate Request Payload

6.3.7 Certificate Request Payload

Dies ist eine Datenstruktur, die dazu benutzt wird, in einer ISAKMP-Nachricht von der Gegenstelle ein Zertifikat anzufordern. Sie ist nicht an bestimmte Nachrichten oder Zustände von IKE gebunden und sollte dann verwendet werden, wenn den beteiligten Gegenstellen kein Verzeichnisdienst für die Aufgaben der Zertifikatverteilung zur Verfügung steht. Der Empfänger einer Certificate Request Payload muss dem Sender sein Zertifikat zusenden.

Certificate Type Der Typ des angeforderten Zertifikats wird in diesem Feld eingetragen. Es gelten die gleichen Typvereinbarungen wie in der Certificate Payload.

Certificate Authority In diesem Bereich steht ein Wert, der die Certificate Authority identifiziert, die der Sender akzeptiert. In Fällen, in denen dies nicht notwendig ist, kann dieses Feld auch leer bleiben. Das Format des Eintrags ist meist der so genannte Distinguished Name der ausstellenden CA.

6.3.8 Hash, Signature und Nonce Payload

Diese von ihrer Struktur her identischen drei Datentypen übertragen entweder einen Hash-Wert, eine digitale Signatur oder einen Nonce.

Hash Dieses Feld enthält einen Hash-Wert, der aus der Hash-Funktion gebildet wurde, die durch ISAKMP-Nachrichten zur Bildung der ISAKMP-Sicherheitsassoziation zwischen den beiden Gegenstellen vereinbart wurde. Die Länge der Daten im Hash-Feld hängt vom verwendeten Algorithmus ab.

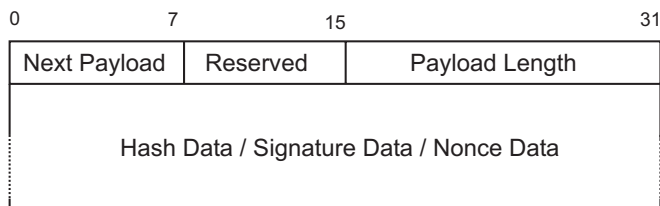


Abbildung 6.11: Das Format der Hash, Signature und Nonce Payload

Signature

Im Falle einer Signature Payload ist der Wert im Datenfeld eine digitale Signatur, die von der Funktion erzeugt wurde, die bei der Aushandlung der ISAKMP-Sicherheitsassoziation vereinbart wurde. Die Länge ist auch vom Signaturalgorithmus abhängig.

Nonce

Nonce ist ein Name für einen Wert, der eine sehr starke Zufallskomponente enthält. Er wird verwendet, um den Nachrichtenaustausch vor Angriffen durch wiederholtes Senden (Replay-Angriff) zu schützen. Er wird meist in Schlüsseltauschpaketen mit übertragen.

6.3.9 Notification Payload

Diese Datenstruktur wird verwendet, um darin Informationen zwischen zwei IKE-Gegenstellen auszutauschen. Dies können Mitteilungen über bestimmte Fehlerzustände sein, zum Beispiel darüber, warum in einer Aushandlung ein Vorschlag zurückgewiesen wurde. Die Felder haben folgende Bedeutungen:

Domain of Interpretation (DOI)

Dieses Feld zeigt, in welcher Domain of Interpretation (ISAKMP oder IPSec) die Mitteilung erfolgt.

Protocol-ID

Hier steht die Protokollnummer des Protokolls, auf das sich die Mitteilung bezieht. Im Fall von IP-Security können das ISKMP, IPSec-ESP und IPSec-AH sein.

SPI Size

Dieser Wert gibt an, wie groß das Feld Security Parameter Index im übernächsten Bereich der Datenstruktur ist. Falls es sich um eine ISAKMP-Notification handelt, ist dieser Wert irrelevant, da der SPI in diesem Fall aus der Kombination von Initiator- und Responder-Cookie besteht.

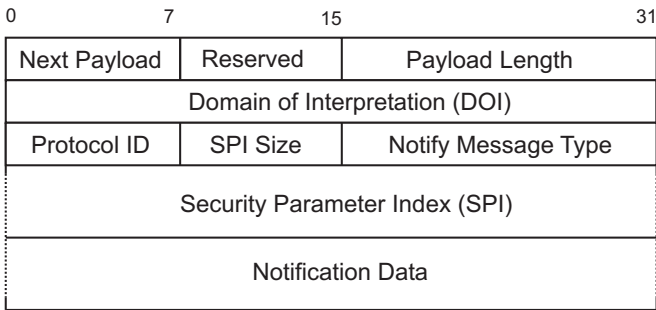


Abbildung 6.12: Die Notification Payload

Notify Message Type

Der Typ der Nachricht, der in einem 16 Bit langen Wert kodiert ist, legt fest, um welche Art von Notification Payload es sich handelt. Im Augenblick sind im Standard nur solche Typen festgelegt, die Auskunft über die Fehlerzustände geben, aufgrund derer der Aufbau einer Sicherheitsassoziation gescheitert ist.

Security Parameter Index

In diesem Feld steht der SPI der empfangenden Gegenstelle.

Notification Data

Hier stehen nähere Angaben zu dem Fehler, der durch den Notify Message Type spezifiziert wurde.

6.3.10 Delete Payload

Wenn eine Gegenstelle eine Sicherheitsassoziation gelöscht hat, schickt sie ihrem Partner diese Nachricht, um dies anzuzeigen. Sie erwartet keine Bestätigung mehr, dass der Empfänger die Nachricht erhalten hat. Es wird empfohlen, dass eine Gegenstelle, die eine Delete Payload erhält, die entsprechende Sicherheitsassoziation ebenfalls löscht, da sie nicht mehr in Benutzung ist und auch nicht reaktiviert werden kann.

Domain of Interpretation (DOI)

Dieses Feld teilt mit, in welcher Domain of Interpretation (ISAKMP oder IPSec) die Mitteilung erfolgt.

Protocol ID

Hier steht die Protokollnummer des Protokolls, das durch die gelöschte SA geschützt wurde. Im Fall von IP-Security können dies ISKMP, IPSec-ESP und IPSec-AH sein.

0	7	15	31
Next Payload	Reserved	Payload Length	
Domain of Interpretation (DOI)			
Protocol ID	SPI Size	Notify Message Type	
Security Parameter Index (SPI)			

Abbildung 6.13: Die Delete Payload

SPI Size

Dieser Wert gibt die Länge der SPIs an. Da nur ein einziges Protokoll pro Datenstruktur möglich ist, sind alle SPIs vom gleichen Typ und gleich lang. Falls eine ISAKMP-Sicherheitsassoziation gelöscht wird, ist dieser Wert obsolet, da deren SPI aus den beiden Cookies besteht.

Number of SPIs

Falls mehrere zu einem Protokoll gehörende SPIs gelöscht werden, kann dies in einer einzigen Mitteilung erfolgen. In diesem Fall werden die SPIs der gelöschten Sicherheitsassoziation nacheinander in einer Delete Payload übertragen. Was jedoch nicht funktioniert, ist, mehrere verschiedene Protokolle in einer Nachricht unterzubringen. Es muss für jedes Protokoll eine eigene Datenstruktur verwendet werden.

Security Parameter Index(es)

Hier stehen der oder die gelöschten SPIs, falls es sich nicht um eine ISAKMP-SA handelt.

6.3.11 Vendor ID Payload

In das Datenfeld dieser Nachricht kann der Hersteller des IPSec-Systems eine selbst definierte Konstante eintragen. Damit kann er seine eigenen Implementierungen und auch verschiedene Versionen davon identifizieren.

Die Vendor ID wird nicht selten benutzt, um in der IKE-Phase 2, nach dem Aufbau der ISAKMP-SA, herstellerspezifische Nachrichten und Informationen austauschen zu können. Dazu müssen sich beide Gegenstellen die notwendigen IDs zugeschickt haben. Die Vendor ID muss während des Aufbaus der ISAKMP-Sicherheitsassoziation ausgetauscht werden; später, in Phase 2, ist dies nicht mehr möglich.

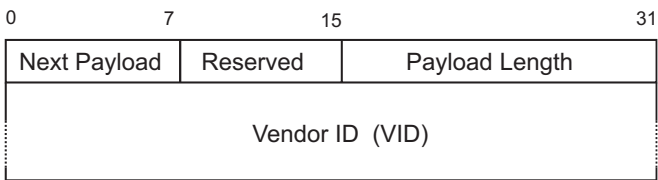


Abbildung 6.14: Die Vendor ID Payload

Eine mögliche und von vielen Herstellern für ihre IPSec Remote Access Clients eingesetzte Anwendung der Vendor ID, in Verbindung mit nachfolgenden privaten Payload-Formaten, ist die Konfiguration von entfernten IPSec-Clientsystemen mit Parametern wie dynamisch zugewiesenen IP-Adressen, Netzwerkmasken, WINS-Adressen, Policies und allen möglichen anderen Parametern.

Im Datenfeld selbst wird in der Regel kein Text eingetragen, sondern ein Hash-Wert, der im Allgemeinen über den Namen der Firma, den Produktnamen, eine Versionsnummer oder bestimmte Kapazitäten der jeweiligen Implementierung berechnet wurde.

6.4 Die ISAKMP-Austauschvorgänge

Die Austauschvorgänge im Internet Security Association and Key Management Protocol werden in zwei Phasen unterteilt. In der Phase 1 wird eine ISAKMP-Sicherheitsassoziation aufgebaut, unter deren Schutz – dies ist dann die Phase 2 – die Austauschvorgänge für die Sicherheitsprotokolle wie IPSec-ESP oder IPSec-AH erfolgen können. Dies ist natürlich ein

etwas zeitaufwändiger Prozess, wenn zum Beispiel für eine IPSec-ESP-Sicherheitsassoziation vorher noch eine ISAKMP-SA aufgebaut werden muss. Falls aber innerhalb einer Verbindung noch weitere Protokoll-SAs erzeugt werden sollen, kann dies unter dem Schutz der ISAKMP-SA erfolgen, was dann insgesamt nicht mehr so aufwändig ist.

Die Austauschvorgänge in den entsprechenden Phasen werden während der Beschreibung des IKE-Protokolls ausführlich erläutert, deshalb folgt hier nur ein kurzer Überblick über die Phasen als solche:

6.4.1 Phase 1

Die Phase 1 ist die Phase, in der noch keine Sicherheitsdienste (Verschlüsselung, Integritätsprüfung, Authentifizierung) zwischen Initiator und Responder zur Verfügung stehen. Also müssen hier besondere Sicherheitsvorkehrungen greifen. In der Phase 1 wird eine ISAKMP-Sicherheitsassoziation etabliert, welche die notwendigen Sicherheitsdienste aushandelt und zur Anwendung bringt, um sowohl sich selbst als auch die Operationen in der folgenden Phase 2 zu schützen.

6.4.2 Phase 2

Die Operationen in der Phase 2 sind bereits durch die Sicherheitsdienste geschützt, die in der Phase 1 ausgehandelt wurden. Aus diesem Grund kommt man in der Phase 2 auch mit weniger Nachrichten als in der Phase 1 aus, weshalb diese Phase im IKE-Protokoll, das die ISAKMP-Austauschvorgänge benutzt, auch als IKE Quick Mode bezeichnet wird. Hier werden die eigentlichen Security-Protokolle ausgehandelt, die zum Schutz der Nutzdaten dienen, die zwischen den beiden Gegenstellen übertragen werden sollen.

6.4.3 Die Austauschvorgänge

Im ISAKMP sind verschiedene Austauschvorgänge festgelegt, die auch vom IKE-Protokoll benutzt werden. Diese Vorgänge legen genau fest, in welcher Reihenfolge welche Art von Nachrichten ausgetauscht werden, wie diese Nachrichten aufgebaut sind und wie die Nachrichten von den IKE-Instanzen zu verarbeiten sind. Eine ausführliche Beschreibung der Austauschvorgänge und Nachrichten erfolgt in den Kapiteln zum IKE-Protokoll. Hier wird nur ein Überblick vermittelt, welche Arten es gibt.

Folgende Austauschvorgänge sind im Internet Security Association and Key Management Protocol spezifiziert:

Base Exchange

Dieser Vorgang besteht aus insgesamt vier Nachrichten, zwei in jede Richtung. In den ersten beiden Nachrichten wird die Sicherheitsassoziation ausgehandelt, die beiden folgenden dienen zum Schlüsseltausch und zur Authentifizierung. Die Informationen zur Authentifizierung, also auch zur Identität der Gegenstellen, und zum Schlüsseltausch werden in einer Nachricht zusammengefasst und somit im Klartext übertragen, da zu diesem Zeitpunkt offensichtlich aufgrund der noch fehlenden Schlüssel keine Verschlüsselung möglich ist.

Identity Protection Exchange

Hier wird das Manko, dass in der Base Exchange die Informationen zur Identität der Kommunikationspartner im Klartext übertragen werden, behoben. Allerdings wird dies durch einen Overhead in Form zweier zusätzlicher Nachrichten erkauft, so dass hier insgesamt sechs statt vier Nachrichten ausgetauscht werden müssen. Hier werden in den letzten beiden Nachrichten die Informationen zur Identität übertragen, verschlüsselt mit dem Schlüssel, der aus den Daten der beiden vorhergehenden Exchanges vom Schlüsseltauschprotokoll berechnet wurde. Daher rührt auch der Name für diesen Austauschvorgang.

Dieser Austausch ist die Grundlage für den IKE Main Mode, den man in IPSec-Security-Gateways einzusetzen pflegt.

Aggressive Exchange

Dieser Austausch besteht aus insgesamt nur drei Nachrichten und zeichnet sich dadurch aus, dass in diese Nachrichten so viel an Informationen gepackt wird, wie zum Zeitpunkt des Sendens möglich ist. Ein Schutz der Identität findet nicht statt, und es besteht eine etwas größere Gefahr von Denial-of-Service-Angriffen, da die notwendigen Umlaufzeiten für Schutzmaßnahmen wie Anti-Clogging-Tokens nicht gegeben sind.

Dieser Exchange ist die Basis für den IKE Aggressive Mode, der vor allem in Remote-Access-Applikationen in IPSec-PC-Implementierungen eingesetzt wird.

Authentication Only Exchange

Dieser Austausch dient ausschließlich dazu, Nachrichten auszutauschen, in denen Informationen zur Authentifizierung enthalten sind. Im IKE-Protokoll finden sie keine Verwendung.

Informational Exchange

Ein derartiger Austausch besteht nur aus einer einzigen Nachricht. Eine Bestätigung ist, trotz Verwendung von UDP als Transportprotokoll, nicht vorgesehen. Dieser Austausch dient zum Management von Sicherheitsassoziationen und zur Information über Fehlerzustände.

6.4.4 Das Oakley Key Determination Protocol

Das Oakley Key Determination Protocol ist ein Schlüsseltauschprotokoll, das auf der Grundlage von ISAKMP und des Diffie-Hellman-Verfahrens einen authentifizierten Schlüsseltausch ermöglicht. Es dient neben dem Internet Security Association and Key Management Protocol als Grundlage des IKE-Protokolls.

Oakley benutzt die Austauschvorgänge des ISAKMP-Protokolls. Das Format der Nachrichten und ihre Verarbeitung wird in den folgenden Kapiteln zu IKE näher erläutert. An dieser Stelle seien jedoch die so genannten Oakley-Gruppen erwähnt. Diese Gruppen spezifizieren die Umgebung, in der ein Schlüsseltausch stattfindet, indem sie Grundparameter des Diffie-Hellman-Algorithmus festlegen. Es werden der Generator und die Primzahl angegeben, die als Modulus benutzt wird. Weiterhin wird festgelegt, ob die Gruppe auf elliptischen Kurven oder dem Modulus einer Exponentiation beruht. Das

Gruppenkonzept bringt unter anderem den Vorteil, dass sich Schlüsseltauschprogramme nur noch auf die Gruppe einigen müssen. Eine Übertragung der ganzen anderen notwendigen Informationen kann entfallen, da sie den Gegenstellen ja im Voraus bekannt sind.

Im IKE-Protokoll muss die Oakley-Gruppe 1 unterstützt werden, andere Gruppen sind optional. In die Gruppen wurde sehr viel an Arbeit investiert, so wurden zum Beispiel alle Primzahlen getestet, was bei einer 1536 Bit großen Primzahl doch schon einiges an Aufwand erfordert. Es sind zurzeit fünf wohl bekannte Oakley-Gruppen definiert.

Die Oakley Gruppe 1

Diese Gruppe basiert auf modularer Exponentiation und verwendet eine 768 Bit große Primzahl als Modulus und den Generator 2. Um sich ein Bild von dieser 232-stelligen Primzahl machen zu können, sei sie hier einmal in dezimaler Schreibweise dargestellt:

1.552.518.092.300.708.935.130.918.131.258.481.755.631.334.049.434.514.313.202.351.194.90
2.966.239.949.102.107.258.669.453.876.591.642.442.910.007.680.288.864.229.150.803.718.91
8.046.342.632.727.613.031.282.983.744.380.820.890.196.288.509.170.691.316.593.175.367.46
9.551.763.119.843.371.637.221.007.210.577.919

Der Diffie-Hellman-Algorithmus bildet seinen öffentlichen Wert, indem er die Zahl 2, den so genannten Generator, mit einer 180 Dezimalstellen großen Zufallszahl, dem privaten Wert, exponenziert und durch die obige Primzahl modulo dividiert. Das Ergebnis ist der öffentliche Wert des Diffie-Hellman-Verfahrens, der über ein unsicheres Medium zur Gegenseite geschickt werden kann. Man kann hier auch erkennen, dass solche Verfahren einiges an Rechenzeit verschlingen, und sollte sich dies beim Design seines VPN vor Augen halten.

Diese Gruppe muss in IKE implementiert werden, alle anderen Gruppen sind optional.

Die weiteren Oakley-Gruppen

Die restlichen Oakley-Gruppen definieren modulare Exponentiationen (ModP) oder basieren auf Gruppen elliptischer Kurven (ECC) über Galois-Feldern. Folgende Gruppen sind spezifiziert:

Gruppe	Modulus	Generator	Referenz

Tabelle 6.1: Die Oakley-Gruppen auf Basis modularer Exponentiation (ModP)

Diese Gruppen sind in IKE optional, die meisten Implementierungen unterstützen zusätzlich zur obligatorischen Gruppe 1 noch die Gruppen 2 und 5 und zunehmend auch die modernen und effizienten ECC-Verfahren. Von den ECC-Verfahren sind zwei (Gruppe 3 und 4) im RFC2409 standardisiert, die restlichen beiden sind in einem informellen RFC beschrieben. An dieser Stelle sei erwähnt, das sich die Gruppen 3 und 4 aufgrund ihrer Natur (keine Primzahl als Basis) mittlerweile als unsicher erwiesen haben und möglichst nicht mehr eingesetzt werden sollten! Diese Probleme gelten für die ECC-Gruppen aus dem Internet-Draft „draft-ietf-ipsec-ike-ecc-groups-06.txt“ nicht, hier werden insbesondere die Gruppe 7 und 9 (163 und 283 Bit) in aktuellen IKE-Implementierungen oft verwendet. Die Gruppen 7 bis 13 entstammen einer Empfehlung des NIST im Anhang zum FIPS-186-2-Dokument, das den digitalen Signaturstandard beschreibt. Für weitere Informationen zum Thema ECC sei auf das Kapitel zur Sicherheitstechnologie oder die einschlägige Fachliteratur verwiesen.

Gruppe	Kurventyp	Schlüssellänge	Referenz
	^N), N nicht prim	155 Bit	RFC2409
4	GF(2 ^N), N nicht prim	185 Bit	RFC2409
7	GF(2 ^N), N prim, Binär, Koblitz	163 Bit	FIPS-186-2
8	GF(2 ^N), N prim, Binär, Koblitz	233 Bit	FIPS-186-2
9	GF(2 ^N), N prim, Binär, Koblitz	283 Bit	FIPS-186-2
10	GF(2 ^N), N prim, Binär, Random	409 Bit	FIPS-186-2
11	GF(2 ^N), N prim, Binär, Koblitz	409 Bit	FIPS-186-2
12	GF(2 ^N), N prim, Binär, Random	571 Bit	FIPS-186-2
13	GF(2 ^N), N prim, Binär, Koblitz	571 Bit	FIPS-186-2

Tabelle 6.2: Die verschiedenen ECC-Gruppen für das IKE-Protokoll

Die kurzen Schlüssellängen der ECC-Verfahren haben keine reduzierte Sicherheit zur Folge. Die Algorithmen sind sehr viel stärker als die, die auf der Basis modularer Exponentiation beruhen. Durch die kurzen Schlüssel können die Berechnungen sehr effizient und damit schnell erfolgen. In Tabelle 6.3 finden Sie eine Übersicht über vergleichbare Schlüssellängen unterschiedlicher Verfahren.

	ECC GF(2 ^N), N prim	Diffie-Hellman, RSA

Tabelle 6.3: Vergleichbare Schlüssellängen unterschiedlicher Algorithmen

6.5 Der Aufbau von IKE

Das Internet-Key-Exchange-Protokoll benutzt ISAKMP, Oakley und die IPSec Domain of Interpretation, um Sicherheitsassoziationen für Sicherheitsprotokolle zu erzeugen und den erforderlichen Schlüsseltausch durchzuführen. Das Konzept des universellen Schlüsseltauschprotokolls SKEME (Secure Key Exchange Mechanism) diente als Grundlage für IKE. Obwohl alle diese Protokolle einen mehr oder minder starken Einfluss auf das Internet-Key-Exchange-Protokoll hatten, ist es ein von diesen völlig unabhängiges, eigenständiges Protokoll.

Zusätzlich zu den in ISAKMP festgelegten Austauscharten werden in IKE zwei neue Exchanges eingeführt: der Quick Mode und der New Group Mode. Der Quick Mode wird in Phase 2 eingesetzt und dient dazu, auf schnelle Art und Weise mit nur drei Nachrichten eine IPSec-Sicherheitsassoziation aufzubauen. Der New Group Mode wird dazu benutzt, zwischen Initiator und Responder eine neue Oakley-Gruppe mit allen notwendigen Parametern zu vereinbaren. Er wird genau genommen zwischen Phase 1 und Phase 2 ausgehandelt, denn sein Ergebnis muss der Phase 2 bereits zur Verfügung stehen. Andererseits darf der New Group Mode nur unter dem Schutz einer ISAKMP-Sicherheitsassoziation ausgehandelt werden, was in Phase 1 geschieht.

IKE definiert alle notwendigen Austauschvorgänge so weit, dass es unmittelbar zu implementieren ist – und zwar sowohl mit der IPSec DOI für IP-Security-Protokolle als auch unter Verwendung anderer DOIs auch mit anderen Protokollen. Hier interessiert aber nur der Einsatz in Verbindung mit IPSec. Das IKE-Protokoll legt auch ein Rollenverhalten für die beteiligten Gegenstellen fest und unterscheidet zwischen dem Initiator, der, wie der Name bereits vermuten lässt, die Kommunikationsvorgänge beginnt, und dem Responder, der darauf antwortet.

IKE benutzt Nachrichten und Austauschvorgänge von ISAKMP und benutzt dessen Daten- und Paketformate. Ebenso wird dessen Modell der beiden Phasen umgesetzt, innerhalb derer die notwendigen Sicherheitsassoziationen erzeugt werden. In Phase 1 sind drei Austauschvorgänge (Exchange) möglich: der IKE Main Mode, der auf dem ISAKMP Identity Protection Exchange aufbaut, der IKE Aggressive Mode, abgeleitet vom gleichnamigen ISAKMP-Austauschvorgang, sowie der Informational Exchange. In der Phase 2 gibt es den so genannten IKE Quick Mode, der keine ISAKMP-Entsprechung hat, jedoch auf dessen Nachrichten und Informationsformaten basiert.

Das Oakley Key Determination Protocol liefert die notwendigen Grundlagen für einen sicheren und authentifizierten Schlüsseltausch. Dort sind auch die Oakley-Gruppen mit allen notwendigen Funktionen definiert, um die notwendige Umgebung zu spezifizieren, in der die Austausch- und Berechnungsvorgänge stattfinden.

6.5.1 Perfect Forwarding Secrecy

Auf praktisch allen heute verfügbaren IPSec-Security-Gateways und IPSec-Host-Implementierungen kann man optional die so genannte Perfect Forwarding Secrecy (PFS) aktivieren. Diese Option legt fest, wie Schlüssel in einer Sicherheitsassoziation generiert werden. Hierzu gibt es grundsätzlich zwei verschiedene Möglichkeiten: Entweder man generiert alle Schlüssel auf Basis der gleichen Grunddaten, oder man erzeugt Schlüssel

immer aus völlig neuen, von den bisher benutzten völlig unabhängigen Daten. Letzteres bezeichnet man als Perfect Forwarding Secrecy.

Dadurch wird erreicht, dass die nachfolgenden Schlüssel nicht von einer Kompromittierung eines Schlüssels betroffen sind, egal wodurch diese zustande kam. Denn ohne PFS wäre es denkbar – sofern ein Schlüssel bekannt wäre –, dass Schlüssel, die aus dem gleichen Hauptschlüssel erzeugt werden, ebenfalls bekannt werden könnten. Mit PFS hingegen ist garantiert, dass alle Schlüssel, die unter dieser Option erzeugt wurden, völlig unabhängig voneinander sind.

6.5.2 Die Attribute einer IPSec Security Association

Die Attribute einer IPSec-Sicherheitsassoziation werden in der IKE-Phase 2 zwischen dem Initiator und Responder ausgehandelt. Sie dienen dazu, zusätzlich zu den ausgehandelten Sicherheitsdiensten der Protection Suite weitere Optionen zwischen den Gegenstellen zu vereinbaren. Diese Attribute werden in speziellen ISAKMP-Nutzlastformaten übertragen. Hierbei unterscheidet man grundsätzlich zwischen Attributen variabler und fester Länge, was auch das Format der Pakete bestimmt. Das erste Bit in der Datenstruktur, die zur Übertragung eingesetzt wird, legt fest, ob es sich um ein Attribut fester (1) oder variabler (0) Länge handelt. Das zweite 15 Bit lange Feld legt den Typ des Attributs fest, und das dritte Feld enthält entweder den Wert oder die Länge des Attributs, je nachdem, ob es eine feste Länge hat oder nicht. Falls es ein Attribut variabler Länge ist, ist noch ein viertes Feld vorhanden, in dem der Attributwert enthalten ist.

Es gibt eine Reihe verschiedener Attribute, von denen folgende obligatorisch für die Aushandlungen in der Phase 2 (IPSec-Aushandlung) einer IKE-Implementierung sind:

- SA Life Type

- SA Life Duration

- Authentication Algorithm

- Optionale IPSec-Attribute

SA Life Type

Dieses Attribut legt die Einheit in Sekunden oder Kilobyte fest, in der die Lebensdauer der vorliegenden Sicherheitsassoziation angegeben ist. In der Regel sollte man eine Lebensdauer auf Basis übertragener Pakete festlegen, denn nur damit kann man seine Sicherheitsstrategie darauf ausrichten, wie viele Daten maximal entschlüsselt werden können, falls der dazugehörige Schlüssel kompromittiert ist.

SA Life Duration

Hiermit wird die Länge der Lebensdauer der vorliegenden SA abhängig vom SA Life Type in Sekunden oder Kilobyte festgelegt.

Hier ein Tipp, falls es bei IPSec-Implementierungen verschiedener Geräte zu Problemen beim Aufbau einer SA kommt. Oft sind unterschiedliche Werte in den beteiligten Endgeräten die Ursache für derartige Probleme. Der Standard, hier die IPSec DOI, schreibt kein verbindliches Verhalten vor, falls aufgrund der Sicherheitsstrategie des Responders die

vom Initiator vorgeschlagene Lebenszeit nicht akzeptabel ist. Hier gibt es drei Möglichkeiten: Entweder kann die Aushandlung sofort scheitern, der Responder akzeptiert das Proposal, benutzt aber einfach seine eigenen Werte ohne Rückmeldung an den Initiator, oder er akzeptiert den Vorschlag, benutzt seine eigene Lebensdauer und informiert den Responder mit einem informellen Nachrichtenpaket über seinen eigenen Wert.

Authentication Algorithm

Dieser Wert muss vorhanden sein, um den korrekten Authentifizierungsalgorithmus für IPSec-ESP oder IPSec-AH festzulegen. Es gibt nur eine Ausnahme, in der dieser Wert nicht präsent sein muss, und zwar dann, wenn IPSec-ESP ohne Authentifizierung benutzt wird. In diesem Fall darf dieser Wert nicht vorhanden sein.

Optionale IPSec-Attribute

Weitere optionale IPSec-Attribute spezifizieren Dinge wie IPSec-Modi (Tunnel- oder Transport-Modus), weitere Oakley-Gruppen, Werte für spezielle Verschlüsselungsalgorithmen mit variabler Schlüssellänge und Rundenzahl und private Komprimierungsverfahren. Folgende Attribute sind in der IPSec Domain of Interpretation festgelegt:

- Group Description
- Encapsulating Mode
- Key Length
- Key Rounds
- Compress Dictionary Size
- Compress Private Algorithm

6.5.3 Die Attribute einer IKE Security Association

Im Internet Key Exchange Protocol gibt es noch eine Reihe weiterer Attribute, die in der Phase 1 zur Aushandlung der ISAKMP-SA benutzt werden und daher nicht im IPSec-DOI, sondern direkt im IKE-Protokoll festgelegt sind:

- Encryption Algorithm
- Hash Algorithm
- Authentication Method
- Group Attributes
- Life Type
- Life Duration
- PRF
- Key Length
- Field Size
- Group Order

Encryption Algorithm

Dieses Attribut legt den Verschlüsselungsalgorithmus fest, der zur Verschlüsselung der IKE-Nachrichten benutzt wird. Folgende Algorithmen sind im Augenblick von der IANA (Internet Assigned Numbers Authority) festgelegt, weitere können folgen oder privat zwischen zwei Gegenstellen festgelegt werden:

- DES-CBC (obligatorisch)
- Triple-DES-CBC
- IDEA-CBC
- Blowfish-CBC
- RC5-R16-B64-CBC
- CAST-CBC
- AES-CBC

Hash Algorithm

Das Hash-Verfahren, das innerhalb von IKE zur Integritätsprüfung und Authentifizierung benutzt werden kann, kann aus den drei folgenden Algorithmen ausgewählt werden:

- MD5 (obligatorisch)
- SHA (obligatorisch)
- Tiger
- RIPEMD-160
- AES-XCBC-PRF-128

Die Algorithmen müssen sowohl den HMAC als auch ihren natürlichen Modus unterstützen. Falls kein PRF-Attribut (siehe unten) zwischen den beiden Gegenstellen vereinbart wird, dann werden diese Algorithmen als Pseudo Random Function (PRF) in IKE benutzt, unter anderem auch, um den Hauptschlüssel zu berechnen.

Authentication Method

Hier wird festgelegt, welche Authentifizierungsmethode in der Phase 1 benutzt wird. Folgende Varianten sind aushandelbar:

- Pre-Shared Key (obligatorisch)
- DSS Signatures
- RSA Signatures
- Encrypt with RSA
- Revised Encryption with RSA

Die Aushandlung einer Authentifizierungsmethode hat einen entscheidenden Einfluss auf das Verhalten der Aushandlungen und ihre Nachrichtenformate.

Group Attributes

Diese Attribute ermöglichen es, eine andere als die Oakley-Gruppe 1 zwischen Initiator und Responder zu vereinbaren, die später weiteren Austauschvorgängen in der Phase 2 zur Verfügung stehen kann. Dabei kann auf wohl bekannte Oakley-Gruppen zurückgegriffen werden, indem man einfach deren Gruppenattribut austauscht oder man komplett neue Gruppen spezifiziert und die notwendigen Parameter wie Primzahlen, Generatoren, Kurvenparameter usw. austauscht. Die auf modularer Exponentiation basierende Gruppe 1, mit einer 768 Bit großen Primzahl, ist in IKE obligatorisch. Folgende Parameter können zusätzlich zur Gruppennummer (Group Description) ausgetauscht werden, um eine andere Gruppe zu erzeugen:

Group Description

- Default 768-Bit-MODP Group (obligatorisch)

- Alternate 1024-Bit-MODP Group

- EC2N Group on GP[2155]

- EC2N Group on GP[2185]

Group Type

- MOP (Modulare Exponentiation)

- ECP (Elliptische Kurven über Galois-Feld $GF[P]$)

- EC2N (Elliptische Kurven über Galois-Feld $GF[2N]$)

- Group Prime or Irreducible Polynomial

- Group Generator One

- Group Generator Two

- Group Curve A

- Group Curve B

Life Type

Damit wird die Einheit der Lebensdauer der ISAKMP-SA festgelegt. Diese kann in Sekunden oder Kilobyte angegeben werden.

Life Duration

Hierin wird die Länge der Lebensdauer der vorliegenden SA in der im Attribut Life Type ausgehandelten Einheit angegeben.

PRF

Dieses Attribut wird in fast keiner Implementierung benutzt. Es gibt auch keine Spezifikationen hierzu. Als PRF (Pseudo Random Function) werden in IKE daher meist die HMAC-Versionen des ausgehandelten Hash-Algorithmus verwendet. Jedoch können Implementierungen auch private Funktionen angeben und in einer IKE-Aushandlung vereinbaren.

Key Length

Bei Verschlüsselungsalgorithmen mit variabler Schlüssellänge muss hier die verwendete Größe des Schlüssels angegeben werden. Falls Algorithmen mit fester Schlüssellänge benutzt werden, darf dieses Attribut nicht verwendet werden.

Field Size

Die Feldgröße in Bits einer Diffie-Hellman-Gruppe wird in diesem Attribut übertragen.

Group Order

Hierin werden Angaben zur Gruppenreihenfolge zu einer Diffie-Hellman-Gruppe übertragen, die mit elliptischen Kurven arbeitet.

6.5.4 IKE-Sicherheitsverfahren

Im Internet-Key-Exchange-Protokoll werden sowohl bidirektionale ISAKMP-Sicherheitsassoziationen als auch unidirektionale SAs für die Sicherheitsprotokolle von IPSec erzeugt. Zu jeder SA existiert eine so genannte Protection Suite (PS), die von den beiden Seiten ausgehandelt wurde und festlegt, welche Sicherheitsverfahren benutzt werden, um die Daten zu schützen, die innerhalb der SA übertragen werden. Standardkonforme Implementierungen von IKE und den IP-Security-Protokollen können immer – sofern beide Gegenstellen eine zueinander passende Sicherheitsstrategie verfolgen – eine Sicherheitsassoziation erzeugen. Das wird garantiert, indem alle Implementierungen einen Mindestsatz von Sicherheitsdiensten zur Verfügung stellen müssen, auf die sich alle Beteiligten einigen können. Alle IKE-Implementierungen müssen folgende Dienste für ihre ISAKMP-Sicherheitsassoziation zur Verfügung stellen und aushandeln können:

- DES als Verschlüsselungsalgorithmus

- MD5 und SHA als Hash-Verfahren

- Authentifizierung mittels Pre-Shared Key

- Modulare Exponentiation in der Oakley-Gruppe 1

Weitere Dienste wie Triple-DES oder andere Hash-Algorithmen sind optional und sollten, müssen aber nicht zur Verfügung stehen. In der Praxis werden auch meist Triple-DES, AES und gelegentlich die Oakley-Gruppen 2 und 5 implementiert, um etwas mehr an Sicherheit zu bieten.

6.5.5 Die Schlüsselerzeugung in IKE

Die Erzeugung der Schlüssel in IKE ist ein Punkt, dem man eine gewisse Beachtung schenken sollte, nicht zuletzt deshalb, weil damit auch ein Teil der oft hitzig geführten Diskussionen um Schlüssellängen etwas an Bedeutung verliert.

Wenn Sie bereits in dem Stadium sind, für Ihr geplantes IPSec-VPN bestimmte Gateways und Client-Implementierungen auszuwählen, werden Sie sich nicht selten fragen, wie denn Systeme verschiedener Hersteller mit 112-Bit-Triple-DES, 168-Bit-Triple-DES und 192-Bit-Triple-DES oder einfach nur Triple-DES miteinander kommunizieren können.

Obendrein treten dann noch Anbieter mit so genannter 128-Bit-Software auf den Plan und behaupten, wie alle anderen auch, mit den übrigen kompatibel zu sein.

Nach dem, was im Kapitel zur Sicherheitstechnologie beschrieben wurde, kann das eigentlich gar nicht möglich sein. Ist es auch nicht. Leider sind viele dieser Angaben, milde ausgedrückt, unpräzise, aus dem Zusammenhang gerissen und in einigen Fällen schlicht falsch.

Um die Zusammenhänge etwas zu verdeutlichen, betrachten wir an dieser Stelle, ohne hier den genauen Ablauf der Nachrichten zum Schlüsseltausch kennen zu müssen, wie die IKE-Schlüssel für die IKE- und IPSec-ESP-Sicherheitsassoziationen erzeugt werden. Nehmen wir einmal der Einfachheit halber an, beide SAs sollen Triple-DES als Verschlüsselungs- und MD5 als Hash-Algorithmus benutzen. Man benötigt also drei Schlüssel mit je 56 Bit Länge (aus einem 64-Bit-String). Nehmen wir weiterhin an, es erfolge eine Authentifizierung mit einem Pre-Shared Key im IKE Main Mode. Dann würde die Erzeugung der Schlüssel zur Verschlüsselung der Nutzdaten auf folgende Art und Weise erfolgen:

Es werden verschiedene Schlüssel berechnet: ein Grundschlüssel (SKEYID), aus dem alle weiteren Schlüssel abgeleitet werden, ein Schlüssel (SKEYID_e) zur Verschlüsselung der ISAKMP-SA, ein Schlüssel (SKEYID_a) zur Authentifizierung der Nachrichten und ein Schlüssel (SKEYID_d), aus dem in nachfolgenden IPSec-Sicherheitsassoziationen in Phase 2 weitere Schlüssel generiert werden können. Diese Schlüssel werden im Fall der Authentifizierung mit einem Pre-Shared Key auf folgende Weise erzeugt:

$$\text{SKEYID} = \text{HMAC-MD5}(\text{Pre-Shared Key} + N_i)$$
$$\text{SKEYID}_d = \text{HMAC-MD5}(\text{SKEYID} + \text{KE} + C_i + C_r + 0x00)$$
$$\text{SKEYID}_a = \text{HMAC-MD5}(\text{SKEYID} + \text{SKEYID}_d + \text{KE} + C_i + C_r + 0x01)$$
$$\text{SKEYID}_e = \text{HMAC-MD5}(\text{SKEYID} + \text{SKEYID}_d + \text{KE} + C_i + C_r + 0x02)$$

Das Pluszeichen steht für eine Verkettung der Werte. Die Bezeichner C_i und C_r sind die Werte der Initiator- und Responder-Cookies. An diesem Beispiel, an dem uns hier nicht so sehr die Art der Berechnung interessiert, sieht man sehr gut, dass alle vier Schlüssel nur so lang sein können wie die Ausgabe der Hashf-Funktion, in unserem Fall 128 Bit.

Falls aus diesem Wert nun die erforderlichen Werte für den Triple-DES-Algorithmus in der ISAKMP-SA gewonnen werden sollen, muss dieser Wert auf mindestens 192 Bit vergrößert werden. Dies erfolgt durch eine relativ einfache und schnelle Verkettungsfunktion:

$$\text{KEYMAT} = K_1 + K_2 \text{ mit:}$$
$$K_1 = \text{HMAC-MD5}(\text{SKEYID}_e + 0x00)$$
$$K_2 = \text{HMAC-MD5}(\text{SKEYID}_e + K_1)$$

Aus diesem 256 Bit langen String können die drei Teilschlüssel für Triple-DES extrahiert werden. Allerdings – und an dieser Stelle sollte man auf diesen Punkt hinweisen – ist diese Methode nicht geeignet, tatsächlich eine kryptografische Stärke des Schlüssels von tatsächlich 168 Bit zu gewährleisten. Bei den verwendeten Schlüssellängen ist dies zwar kein Sicherheitsproblem, aber eine Brute-Force-Attacke müsste in diesem Fall nicht alle

2^{168} Möglichkeiten von Triple-DES ausprobieren, sondern nur die 2^{128} Möglichkeiten von SKEYID_e, und könnte daraus mit obiger, schnell auszuführender Rechenanweisung die drei Teilschlüssel generieren. Aber wenn man die Kosten hierfür, also die Berechnungszeit für KEYMAT, mit einkalkuliert, dann liegt die Sicherheit hier immer noch deutlich über 128 Bit, was in jedem Fall heutzutage ausreichend ist.

Aus dem Schlüssel SKEYID_d werden alle Schlüssel für unsere IPSec-ESP-Sicherheitsassoziation erzeugt, also die drei 56-Bit-Schlüssel für Triple-DES und der 128-Bit-Schlüssel für HMAC-MD5-96. Falls als Option Perfect Forwarding Secrecy (PFS) ausgehandelt wurde, erfolgt in der IKE-Phase 2, im Quick-Mode-Austausch, nochmals ein neuer Diffie-Hellman-Schlüsseltausch; es stehen also wieder neue KE-Werte zur Verfügung. Wenn keine PFS eingesetzt wird, benutzt man in dieser Phase einfach die Werte aus der Phase 1.

Nachdem diese Austauschvorgänge abgeschlossen sind, wird im IKE Quick Mode das Schlüsselmaterial erzeugt, das von den IPSec-ESP-Sicherheitsprotokollen benötigt wird. Das Schlüsselmaterial ist eine Bitfolge ausreichender Länge, aus der die Schlüssel für Triple-DES und HMAC-MD5 extrahiert werden. Dies hat selbstverständlich auf allen Seiten in der gleichen Art zu erfolgen, denn sonst wäre offensichtlich keine Entschlüsselung oder Authentifizierung auf der jeweils anderen Gegenstelle mehr möglich, da es sich um symmetrische Algorithmen handelt, die identische Werte benötigen!

Aus diesem Grund wurde in dem Standard zur Sicherheitsarchitektur für IPSec (RFC2401) und einem weiteren Dokument zur Benutzung von Verschlüsselungsalgorithmen in IPSec (RFC2451) genau festgelegt, wie aus einem hinreichend langen Schlüsselmaterial die benötigten Schlüssel extrahiert werden müssen. Der RFC2401 legt dabei Folgendes fest: Für symmetrische Verschlüsselungsverfahren werden die Schlüssel aus den n höchstwertigen Bits extrahiert, und die restlichen Bits werden für die Hash-Algorithmen im HMAC-Modus benutzt, wobei die Zahl n die Summe aller benötigten Bits des Verschlüsselungsverfahrens ist. Der RFC2451 legt fest, wie die verschiedenen Algorithmen mit Schlüsseln versorgt werden. In unserem Fall wird dort spezifiziert, dass Triple-DES mit drei unabhängigen Teilschlüsseln von je 64 Bit Länge, also insgesamt 192 Bits, versorgt werden muss. In jedem Schlüssel wird später jedes achte Bit entfernt, so dass drei Schlüssel mit 56 Bit zur Verfügung stehen. Also benötigen wir für unsere Sicherheitsassoziation Schlüsselmaterial mit insgesamt mindestens 320 Bit Länge. Dies wird in IKE auf folgende Weise erzeugt:

KEYMAT = $K1 + K2 + K3$, wobei $K1$ bis $K3$ auf folgende Weise erzeugt werden:

$K1 = \text{HMAC-MD5}(\text{SKEYID\_d} + \text{KE} + \text{Protokoll} + \text{SPI} + \text{Ni} + \text{Nr})$

$K2 = \text{HMAC-MD5}(\text{SKEYID\_d} + K1 + \text{KE} + \text{Protokoll} + \text{SPI} + \text{Ni} + \text{Nr})$

$K3 = \text{HMAC-MD5}(\text{SKEYID\_d} + K2 + \text{KE} + \text{Protokoll} + \text{SPI} + \text{Ni} + \text{Nr})$

Hier wird ein 384 Bit langer Wert erzeugt, dessen erste 320 Bit zur Verwendung als Schlüssel in der IPSec-ESP-Sicherheitsassoziation herangezogen werden. Bei dieser Verkettungsfunktion ist, im Gegensatz zu der für die ISAKMP-SA benutzten, die Möglichkeit gegeben (wie in unserem Beispiel zu sehen), durch Benutzung der Perfect Forwarding Secrecy die Daten der neuen Schlüsselaushandlung (KE) in Phase 2 mit in die Berechnung aufzunehmen. Auf diese Weise wird eine hohe kryptografische Stärke des Schlüsselmaterials erreicht.

Damit wurde aufgezeigt, dass Triple-DES in IPSec-ESP mit drei unabhängigen Teilschlüsseln zu je 56 Bit arbeitet (aus den 64 Bits der drei ursprünglichen Teilschlüssel wurde jedes achte Bit entfernt), was eine Gesamtlänge von 168 Bit ergibt. Es gibt im standardkonformen IPSec keine Schlüssellängen von 128 oder 192 Bit – und schon gar keine Implementierung, die mit Triple-DES und zwei unabhängigen Teilschlüsseln mit insgesamt 112 Bit arbeitet! Solche Implementierungen, wenn es sie denn gäbe, wären zu allen anderen Systemen inkompatibel.

An dieser Stelle sei noch erwähnt, dass die Sicherheit dieser ganzen Verfahren davon abhängt, dass beide Seiten einer Sicherheitsassoziation den Schlüsseltausch (und insbesondere die privaten Zufallszahlen des Diffie-Hellman-Verfahrens) vernünftig und sicher erzeugen und diese Werte sobald möglich vernichten. Sobald nur eine Seite ein böses Spiel treibt und in der Zufallszahl nur eine Entropie von zum Beispiel 20 Bit hätte oder die Zahl gar speichert und weitergibt, wäre damit die ganze Sicherheit der Verbindung verloren, egal wie gut die Gegenstelle ihre eigene Zufallszahl erzeugt! Denn das Wesen des Diffie-Hellman-Austauschs liegt gerade darin, dass zur Erzeugung des Schlüssels nur der eigene private Wert benötigt wird.

IKE legt in der Aushandlung der Sicherheitsassoziation weiterhin fest, auf welche Art die Authentifizierung erfolgen soll. IKE erlaubt die folgenden vier verschiedenen Methoden:

- Pre-Shared Key
- Digital Signature
- Public Key Encryption (RSA)
- Revised Mode of Public Key Encryption (RSA)

Hier erfolgt die Authentifizierung durch Bildung eines Hash-Wertes über eine ganze Reihe von Werten durch die vereinbarte Hash-Funktion. Die Werte bestehen aus dem Hauptschlüssel, der unter anderem aus dem Pre-Shared Key berechnet wurde, den gesamten Nutzdaten der Sicherheitsassoziation (Proposals), den öffentlichen Diffie-Hellman-Werten, den beiden Cookies und den Identitäten von Initiator und Responder. Die Hash-Funktion (PRF, Pseudo Random Function) wird entweder explizit für die SA ausgehandelt, oder es wird die HMAC-Variante der allgemein ausgehandelten Hash-Funktion benutzt.

Die Authentifizierung erfolgt durch die digitale Signatur eines Hash-Wertes, der analog zu dem obigen erzeugt wurde. Die Signatur erfolgt mit dem privaten Schlüssel und einem Signaturalgorithmus, meist RSA. Optional können noch von der Gegenstelle die Zertifikate des Gegenübers angefordert werden, um dessen öffentlichen Schlüssel zu bekommen, falls dieser nicht lokal gespeichert oder über einen Verzeichnisdienst online zur Verfügung steht. Die Zertifikate werden dann zusammen mit der Signatur übertragen.

In diesem Verfahren werden so genannte *Nonces*, spezielle Zufallszahlen, sowie die Identitäten mit dem öffentlichen Schlüssel der Gegenstelle verschlüsselt. Zu diesem Zweck muss der Initiator den öffentlichen Schlüssel des Responders bereits vorher kennen. Als Verfahren wird meist RSA eingesetzt, und der Chiffretext muss im PKCS#1-Format kodiert sein. Die PKCS-Formate sind Industriestandards der Firma RSA Data Security und auf deren Webseite beschrieben. Leider wird diese Art der Authentifizierung nicht oft benutzt, obwohl sie vom Sicherheitsstandpunkt erhebliche Vorteile, auch gegenüber digitalen Signaturen, aufweist. Der Grund liegt in der Performance, die hier benötigt wird, denn zusätzlich zum Diffie-Hellman-Schlüsseltausch sind vier zusätzliche Public-Key-Ver- bzw. Entschlüsselungen pro Gegenstelle notwendig.

Somit ist diese Art der Authentifizierung in Systemen wie Remote-Access-VPN-Konzentratoren, die teilweise eine große Anzahl von Verbindungen in kurzer Zeit aufnehmen müssen, nicht praktikabel, weil die Public-Key-Verfahren praktisch die komplette Rechenzeit okkupieren und den Anmeldevorgang sehr stark verzögern und den laufenden Datenverkehr stören würden. Hier müsste man sehr viel in umfangreiche Ressourcen, wie Hardware-Beschleuniger für Public-Key-Verfahren, investieren, was ein solches System in der Summe sehr teuer machen würde.

Also hat man das Verfahren revidiert und die Zahl der Public-Key-Operationen reduziert. Hier wird zwar der Nonce-Wert immer noch mit dem öffentlichen Schlüssel verschlüsselt, jedoch wird die Identität mit dem ausgehandelten symmetrischen, etwa 1000fach schneller arbeitenden Verfahren verschlüsselt. Der Schlüssel wird aus dem Nonce abgeleitet und auch gleichzeitig benutzt, um die Nachrichten zum Schlüsseltausch zu verschlüsseln.

Von der Art der Authentifizierung im IKE-Protokoll hängt auch die Art und Weise ab, mit der die Hauptschlüssel berechnet werden, aus denen sich die Schlüssel zur Verschlüsselung, Integritätsprüfung und Authentifizierung ableiten. Heutige Implementierungen basieren hauptsächlich auf den ersten beiden Verfahren mit Pre-Shared Key oder digitalen Signaturen. Die beiden anderen Verfahren, insbesondere das letzte, haben leider bisher nur wenig Verbreitung gefunden. Deshalb leider, weil man in einem IKE Aggressive Mode Exchange mit den Public-Key-Verfahren die Identitäten der Kommunikationspartner schützen könnte, da sie in jedem Fall verschlüsselt übertragen werden.

Im IKE Main Mode wird eine ISAKMP-Sicherheitsassoziation erzeugt. Hierfür stehen zwei Austauschtypen zur Verfügung: der so genannte IKE Main Mode, eine Instanziierung des ISAKMP Identity Protection Exchange, und der IKE Aggressive Mode. Sie unterscheiden sich durch die Anzahl von benutzten Nachrichten und die damit verbundene Sicherheitsstufe. Unter bestimmten Bedingungen kann eine Sicherheitsstrategie sogar den Einsatz des Aggressive Mode verbieten.

Im Folgenden sind die Austauschvorgänge im Main Mode beschrieben. Da sie sich, je nach dem ausgehandelten Verfahren zur Authentifizierung, sehr stark in den Formaten und Bedeutungen der Nachrichten unterscheiden, werden sie für jede Version der Übersichtlichkeit halber einzeln beschrieben.

Bei den im Folgenden beschriebenen Main-Mode-Austauschvorgängen wird von folgenden gleichen Bedingungen ausgegangen: Es sollen DES-CBC, MD5 und die Oakley-Gruppe 1 ausgehandelt und benutzt werden. Die vier Beispiele unterscheiden sich nur in der Art der Authentifizierung.

Die Authentifizierung dieses Austauschs basiert auf einem Pre-Shared Key, also einer Information, die beiden Systemen zur Verfügung steht. Die praktische Implementierung sieht in der Regel so aus, dass der Pre-Shared Key manuell auf den Systemen eingetragen wird oder bei IPSec-Host-Implementierungen aus dem Benutzerpasswort gewonnen wird. Der IKE Main Mode besteht aus sechs Nachrichten, die nach ihrer Reihenfolge und nach ihren Aufgaben in drei Gruppen zu unterteilen sind: Aushandlung der Sicherheitsassoziation, Schlüsseltausch und Authentifizierung.

Wie in Abbildung 6.15 zu sehen ist, beginnt der Austausch mit zwei Nachrichten, in denen sich Initiator und Responder auf eine Protection Suite für die ISAKMP-Sicherheitsassoziation einigen – oder auch nicht, falls die Vorschläge für den Responder nicht akzeptabel sind. Im Gegensatz zu anderen Protokollen, wie zum Beispiel PPP, findet hier keine mehrstufige Verhandlung statt, sondern der Initiator unterbreitet eine Reihe von Vorschlägen in einer einzigen Nachricht, und der Responder wählt einen Vorschlag davon aus oder lehnt alles ab. In letzterem Fall kommt die Aushandlung zum Ende, indem sie schlicht terminiert wird. Die Vorschläge (Proposals) des Initiators können sehr komplex aufgebaut und logisch (und, oder) miteinander verknüpft werden. Bei bestimmten Umgebungen, zum Beispiel im Fall von Remote-Access-VPN, in denen eine zentrale Sicherheitsstrategie eingesetzt wird, sieht es in der Praxis meist so aus, dass der Initiator alles in einen Vorschlag packt, was er an Funktionen unterstützt, und das zentrale VPN-Gateway aus diesem Packen das Proposal auswählt, das seiner Sicherheitsstrategie entspricht.

Sobald die ersten beiden Nachrichten ausgetauscht, verarbeitet und ausgewertet sind, steht fest, ob sich Initiator und Responder auf eine ISAKMP-Sicherheitsassoziation einigen konnten oder nicht. Im letzteren Fall wird der Verbindungsaufbau abgebrochen; ansonsten wird durch den vom Responder ausgewählten Vorschlag der weitere Verlauf des Austauschvorgangs festgelegt. In unserem Fall soll eine Authentifizierung auf Basis eines Pre-Shared Keys erfolgen.

Die folgenden beiden Nachrichten dienen zur Übertragung von Informationen, mit deren Hilfe der Initiator und der Responder lokal den symmetrischen Hauptschlüssel berechnen können. Zu diesem Zweck erzeugen sich beide je zwei strikt zufällige Zahlen, von denen eine als Exponent (x im Initiator und y im Responder) für das Diffie-Hellman-Verfahren und die andere als Nonce-Wert dient. Da sich beide Seiten auf die Oakley-Gruppe 1 geeinigt haben, werden nun die öffentlichen Diffie-Hellman-Werte aus der Basis 2, der 768 Bit großen Primzahl n und dem geheimen Exponenten berechnet. Der

Initiator erzeugt seinen Wert mit $2x \pmod n$, und der Responder berechnet $2y \pmod n$. Die Zahlen 2 und n brauchen nicht ausgetauscht zu werden, da sie implizit durch Auswahl der Oakley-Gruppe 1 in den beiden ersten Nachrichten festgelegt wurden.

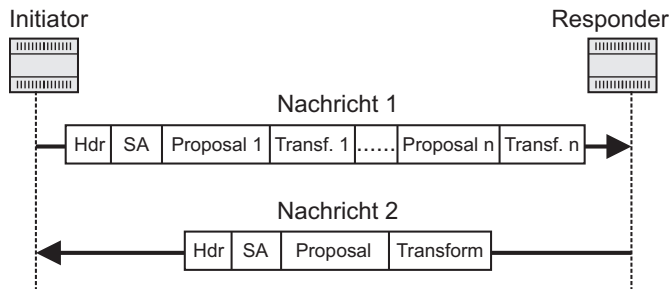


Abbildung 6.15: Die ersten beiden Nachrichten im IKE Main Mode

Aus diesen ganzen Werten formen der Initiator und der Responder die benötigten Datenstrukturen für ihre Nachrichten und tauschen diese anschließend aus, wie in Abbildung 6.16 zu sehen ist. Es erfolgt aber auf keiner der beiden Seiten sofort eine Berechnung des Hauptschlüssels. Dies wird so lange verzögert, bis die Cookies ausgewertet sind und damit der jeweilige Absender verifiziert ist.

$SKEYID = HMAC-MD5(Pres-Shared\ Key + N_i)$

$SKEYID\_d = HMAC-MD5(SKEYID + KE + C_i + C_r + 0x00)$

$SKEYID\_a = HMAC-MD5(SKEYID + SKEYID\_d + KE + C_i + C_r + 0x01)$

$SKEYID\_e = HMAC-MD5(SKEYID + SKEYID\_d + KE + C_i + C_r + 0x02)$

$SKEYID\_e$ wird zur Verschlüsselung der ISAKMP-Nutzdaten benutzt, und $SKEYID\_a$ dient zur Authentifizierung der Pakete mit Hilfe der ausgehandelten Hash-Funktion. In unserem Beispiel sind dies DES zur Datenverschlüsselung und HMAC-MD5 als Hash-Algorithmus. Der Schlüssel $SKEYID\_d$ dient als Grundschlüssel der IPSec-SAs, die in der IKE-Phase 2 erzeugt werden.

An diesem Beispiel wird offensichtlich, dass – falls eine Authentifizierung mit einem Pre-Shared Key durchgeführt wird – der Responder wissen muss, wessen Pre-Shared Key er für die Berechnung von $SKEYID$ benutzen soll. Er besitzt die Keys aller möglichen Gegenstellen und muss einen davon auswählen. Die Identität des Initiators wird ihm aber erst in der vorletzten Nachricht mitgeteilt, die jedoch bereits verschlüsselt sein muss, und zwar mit einem Schlüssel, der teilweise auf dem Pre-Shared Key basiert. Als einzige Möglichkeit, vorher den richtigen Key auszuwählen, bleibt leider nur die Kenntnis der IP-Adresse des Initiators. Leider deshalb, weil damit keine dynamische Zuweisung der IP-Adresse an den Initiator möglich ist, denn der Responder muss eine Tabelle mit IP-Adressen und zugehörigen Pre-Shared Keys pflegen. Aber bei der Einwahl bei Internet Service Providern werden grundsätzlich die IP-Adressen dynamisch zugewiesen. Mögliche Auswege für Remote-Access-VPN bietet die nachfolgend beschriebene Authentifizierung mit digitalen Signaturen oder der IKE Aggressive Mode.

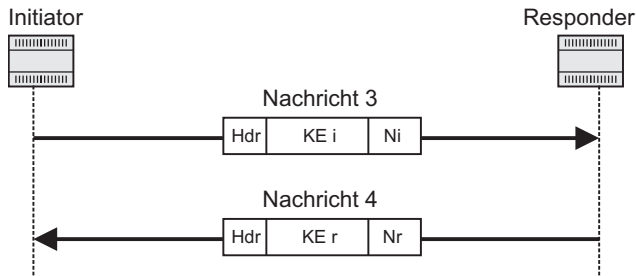


Abbildung 6.16: Der Schlüsseltausch im IKE Mainmode

Die letzten beiden Nachrichten im IKE Main Mode dienen zur gegenseitigen Authentifizierung von Initiator und Responder. Eine beiderseitige Authentifizierung ist im IKE-Protokoll obligatorisch. Zu diesem Zweck bilden beide Seiten einen geeigneten Hash-Wert, anhand dessen die Gegenstelle die Identität des Senders überprüfen kann. Als Eingangswerte für die Hash-Funktion dienen die beiden öffentlichen Diffie-Hellman-Werte (KEi, KEr), der Initiator- und der Responder-Cookie (Ci, Cr), die Daten der Vorschläge des Initiators für die ISAKMP-Sicherheitsassoziation (SA) und die Identität des Senders (IDi, IDr):

$$\text{HASH\_I} = \text{HMAC-MD5}(\text{SKEYID}, \text{KEi} + \text{KEr} + \text{Ci} + \text{Cr} + \text{SAi} + \text{IDi})$$

$$\text{HASH\_R} = \text{HMAC-MD5}(\text{SKEYID}; \text{KEr} + \text{KEi} + \text{Cr} + \text{Ci} + \text{SAi} + \text{IDr})$$

Der Initiator erzeugt eine Nachricht, in der er seine Identität (IDi) und seinen Hash-Wert (HASH_I) zum Responder schickt, um sich zu authentifizieren. Alles außer dem ISAKMP-Header ist in dieser Nachricht verschlüsselt, in unserem Beispiel mit DES, um die Identität des Initiators zu verbergen. Daraus resultiert auch die Namensgebung (Identity Protection Exchange) dieses Austauschvorgangs. Der Responder authentifiziert sich auf die gleiche Art durch die verschlüsselte Übermittlung seines Hashwerts (HASH_R) und seiner Identitätsinformationen (IDr) an den Initiator.

Hier wieder ein Praxistipp: Falls es Probleme innerhalb eines Austauschvorgangs gibt, bekommt man oft die Fehlermeldung, dass die Authentifizierung fehlgeschlagen sei. Als Problemquelle vermutet man hier natürlich zuerst den Pre-Shared Key, was in der Regel auch zutrifft. Aber diese Meldung kann auch ein Indiz für einen Angriff sein. Denn die Authentifizierung in den letzten beiden Nachrichten schlägt auch dann fehl, wenn die anderen Werte wie Cookies, SA-Vorschlag usw. verändert wurden, und nicht nur, wenn der Pre-Shared Key, auf dem SKEYID basiert, nicht identisch ist.

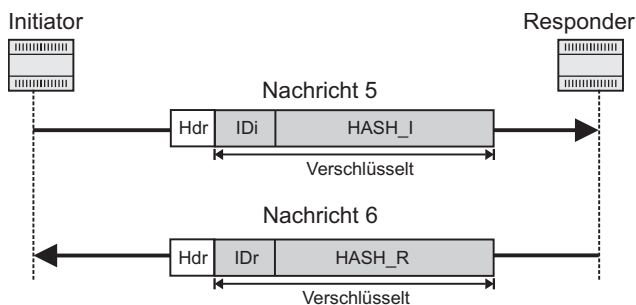


Abbildung 6.17: Im IKE Main Mode erfolgt die Authentifizierung verschlüsselt in den letzten beiden Nachrichten.

Nachdem diese sechs Nachrichten übermittelt und fehlerfrei verarbeitet wurden, besteht eine ISAKMP-Sicherheitsassoziation, unter deren Schutz in der IKE-Phase 2 weitere SAs für IPSec-Protokolle wie IPSec-ESP oder IPSec-AH erzeugt werden können. Dieser Schutz besteht aus Diensten zur Sicherstellung der Nachrichtenintegrität, der Authentizität und der Vertraulichkeit der übertragenen Daten durch Einsatz der ausgehandelten kryptografischen Verfahren.

Eine Authentifizierung mittels einer digitalen Signatur unterscheidet sich innerhalb der ersten beiden Nachrichten im IKE Main Mode nur durch den Inhalt der Vorschläge vom vorhergehenden Beispiel. Hier wird statt der Pre-Shared-Key-Authentifizierung eine mit digitaler Signatur erfolgende ausgehandelt.

Die beiden Nachrichten zum Schlüsseltausch können als optionalen Wert eine Anforderung eines digitalen Zertifikats der Gegenstelle (CERT-REQ) enthalten. Falls dies nicht benötigt wird, sind die Nachrichten zwar identisch zu denen der Pre-Shared-Key-Version, aber die Behandlung der übertragenen Werte unterscheidet sich von der anderer Authentifizierungsmethoden. Der Wert des Hauptschlüssels, aus dem sich alle anderen Schlüssel ableiten, wird nach folgender Rechenvorschrift ermittelt:

$$\text{SKEYID} = \text{HMAC-MD5}(\text{Ni} + \text{Nr} + \text{KE})$$

KE ist das Ergebnis des Diffie-Hellman-Verfahrens, also $2xy \bmod n$. Aus diesem Grundwert werden alle weiteren Schlüssel erzeugt.

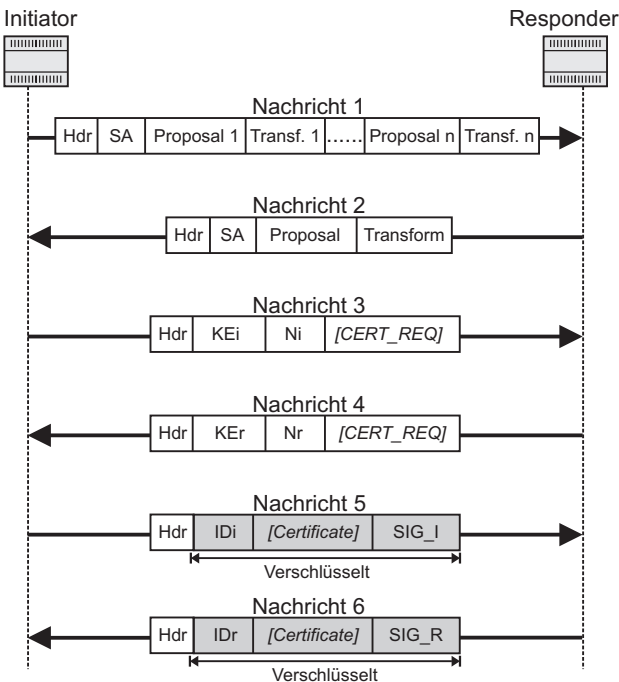


Abbildung 6.18: Ein IKE-Main-Mode-Austausch mit Authentifizierung durch digitale Signaturen.

Die letzten beiden Nachrichten, die zur wechselseitigen Authentifizierung von Initiator und Responder dienen, übertragen keinen Hash-Wert, sondern jeweils eine Signatur der Hash-Werte HASH_I und HASH_R. Im Fall von RSA als Signaturalgorithmus würden die Signaturen aus den Hash-Werten wie folgt erzeugt:

$$\text{SIG\_I} = \text{RSA}(\text{Ki\_private}, \text{HASH\_I})$$
$$\text{SIG\_R} = \text{RSA}(\text{Kr\_private}, \text{HASH\_R})$$

Die privaten Schlüssel Ki_private und Kr_private korrespondieren mit deren öffentlichen Schlüsseln, die zum Verifizieren der Signatur benötigt werden. Diese Schlüssel sind der Gegenstelle entweder bereits bekannt, oder sie werden in einem Zertifikat übertragen, das innerhalb der dritten und vierten Nachricht angefordert werden kann. Im ersten Fall müssen die öffentlichen Schlüssel auf eine andere Art und Weise, die über den Definitionsbereich des IKE-Protokolls hinausgeht, auf den beteiligten Gegenstellen bekannt gemacht werden. Dies kann entweder durch manuelle Schlüssel hinterlegung oder etwas moderner und zuverlässiger durch den Einsatz eines Verzeichnisdienstes erfolgen. Dies gilt auch für Zertifikate: Hier verwendet man eine so genannte PKI (Public Key Infrastructure), welche die nötige Infrastruktur zur Verwaltung und Verteilung von digitalen Zertifikaten bereitstellt.

Diese Art von Verschlüsselung benutzt ein asymmetrisches (Public-Key-)Verfahren, das zur Ver- und Entschlüsselung unterschiedliche, jedoch einander zugeordnete Schlüssel und Algorithmen benutzt. Die beiden Schlüssel werden unterschiedlich behandelt; es gibt einen öffentlichen Schlüssel und einen privaten Schlüssel. Der öffentliche Schlüssel (Public Key) darf jedermann bekannt sein, der private (Private Key) hingegen muss vertraulich bleiben und darf niemandem außer seinem Besitzer bekannt sein. Private Schlüssel werden in der Regel in verschlüsselter Form in speziellen Formaten auf Datenträgern oder SmartCards gespeichert.

Auch im IKE Main Mode verwendet man ein solches Verfahren zur Authentifizierung des Benutzers. Das Prinzip ist recht einfach: Die Seite, die sich identifizieren möchte, überträgt ihren mit dem öffentlichen Schlüssel der Gegenstelle verschlüsselten Nonce- und ID-Wert. Der Empfänger entschlüsselt die Daten und verifiziert sie. Dies kann aber niemand sonst nachvollziehen, da nur der Empfänger im Besitz des notwendigen privaten Schlüssels ist.

Die Art der Authentifizierung wird in den ersten beiden Nachrichten vereinbart. Die Nachrichten zum Schlüsseltausch unterscheiden sich wesentlich von denen zur Authentifizierung mit Signaturen oder einem Pre-Shared Key. Eine Vorbedingung dieses Verfahrens ist, dass der Initiator den öffentlichen Schlüssel des Responders, den er zum Verschlüsseln benötigt, bereits vor dem Austauschvorgang kennt. Ein Austausch von Zertifikaten zu diesem Zweck ist nicht möglich, da diese vorher übertragen werden müssten und damit nicht verschlüsselt werden können. Damit wäre jedoch die Identität der Teilnehmer an dem Austauschvorgang nicht mehr gewährleistet, was aber für diesen Austausch typisch obligatorisch ist. Üblicherweise werden die öffentlichen Schlüssel durch digitale Zertifikate verteilt, da hier eine zuverlässige Bindung der Identitäten der Teilnehmer an diesem Verfahren an ihre öffentlichen Schlüssel besteht.

In den Nachrichten zum Schlüsseltausch werden also, wie in Abbildung 6.19 zu sehen, neben den öffentlichen Diffie-Hellman-Werten auch der verschlüsselte Nonce-Wert und die verschlüsselte Identität des Initiators und in der vierten Nachricht umgekehrt auch die des Responders übertragen. Falls der Initiator mehrere öffentliche Schlüssel des Responders besitzt, muss der diesem weiterhin mitteilen, welchen er benutzt hat. Dies tut er, indem er einen Hash-Wert über das Zertifikat bildet, dessen Schlüssel benutzt wurde, und diesen Wert mit zum Responder schickt. Dieser kann anhand des Hash-Wertes ermitteln, welchen seiner privaten Schlüssel er zum Dechiffrieren benutzen muss. Bei der Verschlüsselung der Nonce- und Identitätswerte werden nur die Nutzdaten verschlüsselt. Die Payload-Header müssen unverschlüsselt bleiben, um die Art und vor allem die Länge der Nutzlast bekannt zu machen.

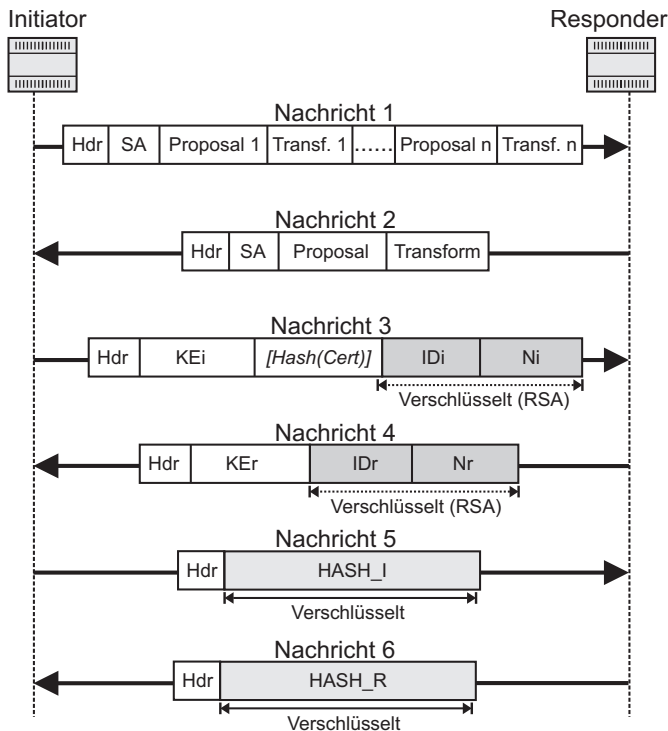


Abbildung 6.19: Die Authentifizierung mit RSA-Public-Key-Verschlüsselung im IKE Main Mode

Die verschlüsselten ID- und Nonce-Nutzdaten, hier durch ein vorangestelltes * gekennzeichnet, werden durch eine RSA-Verschlüsselung wie folgt erzeugt:

$*ID_i = \text{RSA}(K_{r\_public}, ID_i)$
 $*N_i = \text{RSA}(K_{r\_public}, N_i)$
 $*ID_r = \text{RSA}(K_{i\_public}, ID_r)$
 $*N_r = \text{RSA}(K_{i\_public}, N_r)$

Die Entschlüsselung der Informationen erfolgt mit den korrespondierenden privaten Schlüsseln von Initiator und Responder.

Die letzten beiden Nachrichten innerhalb dieses Austauschvorgangs tauschen wieder die Hash-Werte von Initiator und Responder aus, um eine abschließende Authentifizierung beider Identitäten und des vollständigen Austauschs vorzunehmen.

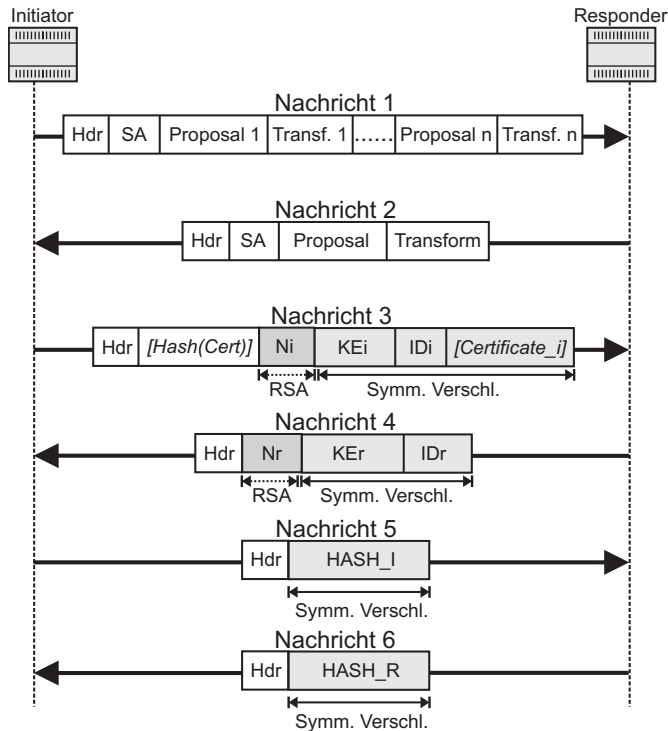


Abbildung 6.20: Die revidierte Public-Key-Verschlüsselung im IKE Main Mode

Dieses Verfahren ist sehr zuverlässig, allerdings zu einem sehr hohen Preis. Denn beide Gegenstellen müssen jeweils vier RSA-Operationen durchführen, jeweils zwei zum Ver- und zwei weitere zum Entschlüsseln. Dies macht diese Art der Authentifizierung sehr rechenintensiv und in der Folge davon sehr langsam. Eine Ver- oder Entschlüsselung mit RSA ist etwa 1000fach langsamer als ein symmetrisches Verfahren wie DES. Dies wirft dann Probleme auf, wenn innerhalb einer kurzen Zeitspanne sehr viele ISAKMP-SAs aufgebaut werden müssen, wie es zum Beispiel in Remote-Access-VPN der Fall ist. Also hat man dieses Verfahren etwas verändert und eine revidierte, schnellere Version der Public-Key-Verschlüsselung spezifiziert.

Diese Version unterscheidet sich von der vorhergehenden dadurch, dass man einen der Verschlüsselungsschritte nicht mit RSA, sondern mit einem symmetrischen Verfahren durchführt. Die Menge der in den Schlüsseltausch-Nachrichten übertragenen Daten ist dadurch zwar etwas größer, aber die Ausführungszeit der notwendigen Berechnungen ist dafür wesentlich kürzer geworden.

Hier wird, wie in Abbildung 6.20 illustriert, nur der Nonce-Wert mit RSA verschlüsselt. Der Rest der Nutzdaten wird mit dem in den ersten beiden Nachrichten ausgehandelten symmetrischen Verfahren verschlüsselt, in unserem Fall also mit DES-CBC. Außer der Identität wird auch noch der öffentliche Diffie-Hellman-Wert verschlüsselt, was noch ein höheres Maß an Sicherheit in diesem Verfahren gewährleistet. Auch hier kann ein Hash-Wert übertragen werden, um ein bestimmtes Zertifikat zu indizieren, aus dem der benutzte öffentliche Schlüssel entnommen wurde.

Die Schlüssel, die benötigt werden, um den öffentlichen Diffie-Hellman-Wert und den Nonce-Wert von Initiator und Responder zu verschlüsseln, wurden aus folgenden Werten berechnet:

$$\begin{aligned} K_{ei} &= \text{HMAC-MD5}(N_i + C_i) \\ K_{er} &= \text{HMAC-MD5}(N_r + C_r) \end{aligned}$$

Es werden für beide Richtungen unterschiedliche Schlüssel verwendet. Falls ein Verschlüsselungsverfahren längere Schlüssel benötigt als die 128 Bit, die HMAC-MD5 erzeugt, dann erfolgt so lange eine Verkettung des Hash-Wertes nach folgender Regel, bis die notwendige Gesamtlänge erreicht wurde:

$$K = K_1 + K_2 + \dots + K_n$$

Die benötigten Teilschlüssel K_1 bis K_n werden wie folgt berechnet:

$$\begin{aligned} K_1 &= \text{HMAC-MD5}(K_{ei} + N_i + 0x00) \\ K_2 &= \text{HMAC-MD5}(K_{ei} + K_1) \\ &\dots\dots\dots \\ K_n &= \text{HMAC-MD5}(K_{ei} + K_{n-1}) \end{aligned}$$

Die zu übertragenden, mit DES verschlüsselten ID- und KE-Nutzdaten, hier durch ein vorangestelltes* gekennzeichnet, werden auf folgende Weise erzeugt:

$$\begin{aligned} *ID_i &= \text{DES}(K_{ei}, ID_i) \\ *ID_r &= \text{DES}(K_{er}, ID_r) \\ *KE_i &= \text{DES}(K_{ei}, KE_i) \\ *KE_r &= \text{DES}(K_{er}, KE_r) \end{aligned}$$

Da DES üblicherweise im CBC-Modus betrieben wird, werden hierfür Initialisierungsvektoren (IV) benötigt. Der erste IV besitzt den Wert 0x0000000000000000, die nachfolgenden erhalten den Wert des Chiffretextes der jeweils vorhergehenden Verschlüsselung.

Die beiden letzten Nachrichten in diesem Austausch führen wieder die abschließende Authentifizierung durch die Hash-Werte von Initiator und Responder durch.

Der IKE Aggressive Mode ist eine Instanziierung des ISAKMP Aggressive Exchange, der den Aufbau einer ISAKMP-Sicherheitsassoziation mit nur drei Nachrichten ermöglicht. Aufgrund der Struktur dieses Austauschvorgangs muss die IP-Adresse des Initiators nicht vorher bekannt sein und kann somit auch, zum Beispiel bei der Einwahl bei einem Internet Service Provider, dynamisch zugewiesen werden.

Das Prinzip des IKE Aggressive Mode ist es, so viel an Informationen in die Nachrichten hineinzupacken, wie es zum gegebenen Zeitpunkt möglich ist. Allerdings hat dies auch seinen Preis, der hier in Form einer etwas reduzierten Sicherheit gegen bestimmte Arten von Denial-of-Service-Angriffen in Erscheinung tritt. Der Grund besteht darin, dass bereits vor dem Umlauf der Cookies sehr rechenintensive Prozesse stattfinden.

Weiterhin ist die Identität des Initiators bei einer Authentifizierung mittels Pre-Shared Key nicht geschützt, da dieser seine Identitätsinformationen bereits in die erste, noch unverschlüsselte Nachricht einfügt. Dies birgt andererseits den Vorteil, dass die IP-Adresse des Initiators nicht statisch sein muss.

Die Aushandlungsmöglichkeiten sind ebenfalls etwas eingeschränkt, da in der ersten Nachricht bereits Werte übertragen werden, die bestimmte Voraussetzungen bedingen, die im Main Mode noch aushandelbar wären. Zum Beispiel wird der öffentliche Diffie-Hellman-Wert des Initiators bereits in der ersten Nachricht übermittelt. Also muss sich der Initiator schon zu diesem Zeitpunkt für eine Oakley-Gruppe entschieden haben und kann nicht mehr verschiedene Vorschläge mit verschiedenen Gruppen machen.

In praktischen Implementierungen, also zum Beispiel in einem Remote-Access-VPN, in dem der IKE Aggressive Mode üblicherweise verwendet wird, ist diese Einschränkung jedoch nicht so einschneidend. Denn in solch einem abgeschlossenen Netz mit einer übergreifenden Sicherheitsstrategie und der Kenntnis der technischen Möglichkeiten der eingesetzten Systeme kennt der Initiator bereits die Sicherheitsstrategie und kann dem Responder ein relativ kurzes Proposal unterbreiten. Er muss nicht alle möglichen Kombinationen und Optionen dort hineinpacken, sondern kann ganz wenige, unter Umständen vielleicht sogar nur einen einzigen Vorschlag machen, von dem er weiß, dass ihn der Responder annimmt.

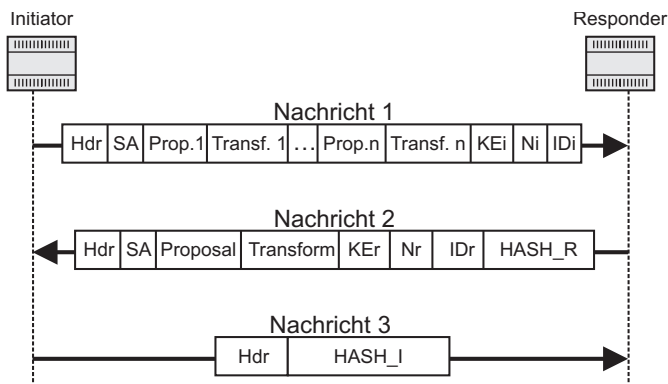


Abbildung 6.21: Der IKE Aggressive Mode mit Pre-Shared Keys

Ein anderes Szenario ist ein Extranet-VPN. Da hier oft Geräte verschiedener Hersteller in verschiedenen Firmen mit unterschiedlichen Sicherheitsstrategien eine ISAKMP-Sicherheitsassoziation aufbauen sollen, sind die Auswahlmöglichkeiten in der Regel reichhaltiger. Allerdings arbeiten solche Verbindungen meist zwischen IPSec-Security-Gateways, die eine feste IP-Adresse besitzen und somit im flexibleren IKE Main Mode arbeiten können.

In unserem Beispiel in Abbildung 6.21 sehen Sie den Aufbau einer ISAKMP-Sicherheitsassoziation im IKE Aggressive Mode. Die Authentifizierung erfolgt durch Pre-Shared Secrets.

In der ersten Nachricht schickt der Initiator der ISAKMP-Sicherheitsassoziation seine Vorschläge (SAi), seinen öffentlichen Diffie-Hellman-Wert (KEi), seinen Nonce-Wert (Ni) und seine Identitätsinformationen (IDi) zum Responder. Diese Nachricht ist unverschlüsselt und enthält im ISAKMP-Header nur das Initiator-Cookie. Somit muss der Responder mit seinen Berechnungen für den Schlüsseltausch beginnen, bevor er sein eigenes Cookie zurückbekommen hat. Er weiß somit noch nicht sicher, ob er auch tatsächlich mit dem Initiator kommuniziert.

Der Responder sucht nun aufgrund der ID-Informationen des Initiators den zugehörigen Pre-Shared Key aus seiner Datenbank und berechnet über das Diffie-Hellman-Verfahren seinen Hauptschlüssel (SKEYID) und alle weiteren, notwendigen Schlüssel. Deren Berechnung erfolgt in der gleichen Weise wie im IKE Main Mode.

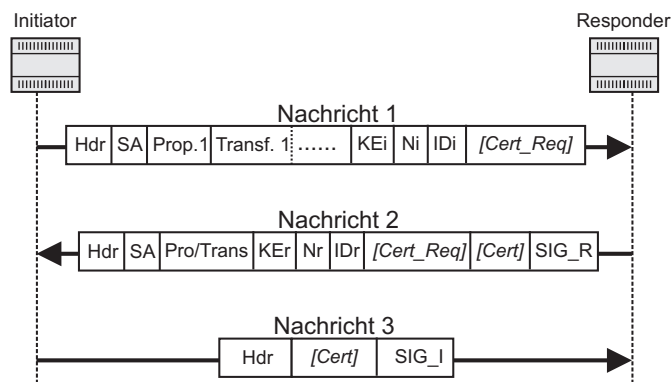


Abbildung 6.22: Authentifizierung mit digitaler Signatur im IKE Aggressive Mode

Der Responder berechnet außerdem seinen öffentlichen Diffie-Hellman-Wert und hat damit genügend Informationen, um seinen Hash-Wert HASH_R zu berechnen. Diesen Hash-Wert überträgt er zusammen mit seinem öffentlichen Diffie-Hellman-Wert (KEr), dem von ihm ausgewählten Vorschlag (SAr), seinem Nonce-Wert (Nr) und seiner Identitätsinformation (IDr) zum Initiator. Dieses Paket ist ebenfalls noch unverschlüsselt, enthält jetzt aber beide Cookies.

Der Initiator kann nun sein Cookie verifizieren und anschließend seine Berechnungen vornehmen, um die erforderlichen Schlüssel zu erzeugen. Nachdem dies geschehen ist, erzeugt der Initiator seinen Hash-Wert HASH_I und überträgt ihn in der dritten und letzten Nachricht des IKE-Aggressive-Mode-Austauschs zum Responder. Diese Nachricht ist verschlüsselt, und das Cookie des Responders ist jetzt auch einmal „umgelaufen“. Er kann es auswerten und anschließend den erhaltenen Hash-Wert überprüfen, um den Initiator und den Austauschvorgang zu authentifizieren.

Auch hier werden nur drei Nachrichten benötigt, in denen alle benötigten Datenstrukturen enthalten sind. Der Unterschied zur Authentifizierung mit einem Pre-Shared Key besteht auch im Aggressive Mode darin, dass statt der Hash-Werte HASH_I und HASH_R die digitalen Signaturen dieser Hash-Werte übertragen werden. Optional können noch digitale Zertifikate übertragen werden. Auch hier gelten alle Einschränkungen wie im vorangegangenen Beispiel, insbesondere die, dass die Identitäten nicht verborgen sind und eine gewisse Gefahr durch Denial-of-Service-Angriffe besteht.

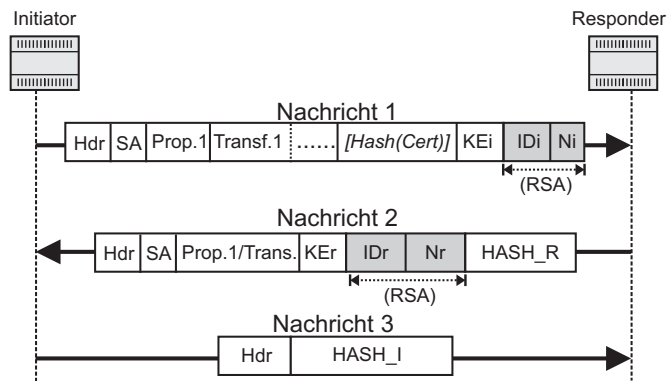


Abbildung 6.23: Die RSA-Public-Key-Verschlüsselung im IKE Aggressive Mode

Die generelle Funktionsweise der Authentifizierung entspricht der im IKE Main Mode. Auch hier werden so viele Daten wie möglich in die Nachrichten gepackt. Hier werden auch nur drei UDP-Pakete zwischen Initiator und Responder ausgetauscht, um eine ISAKMP-Sicherheitsassoziation aufzubauen.

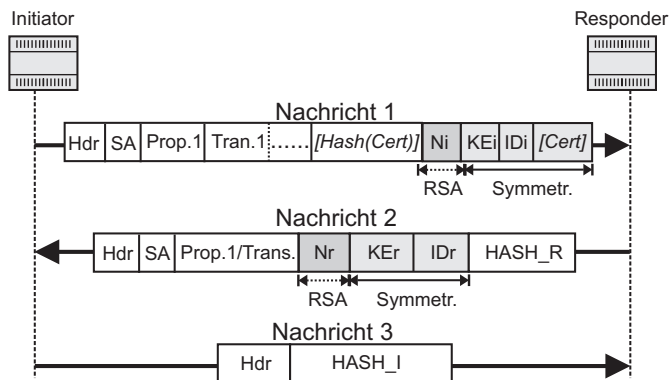


Abbildung 6.24: Die revidierte RSA-Public-Key-Verschlüsselung im IKE Aggressive Mode

Diese Art der Authentifizierung hat gegenüber den beiden vorher beschriebenen den großen Vorteil, auch im IKE Aggressive Mode die Identität der beteiligten Gegenstellen zu verbergen. Denn die Identitätsinformationen werden mit dem öffentlichen Schlüssel der jeweiligen Gegenstelle verschlüsselt und erst dann übertragen. Die Angriffspunkte für Denial-of-Service-Angriffe bleiben jedoch bestehen, weil vollständige Cookie-Umläufe bis zur Berechnung der Diffie-Hellman-Funktion nicht möglich sind.

Auch mit der geänderten Methode der Public-Key-Verschlüsselung wird die Identität von Initiator und Responder geschützt, indem sie verschlüsselt übertragen werden. Die Berechnung der Schlüssel und die Verschlüsselung selbst erfolgen auf die gleiche Art wie im IKE Main Mode, wodurch eine insgesamt höhere Verarbeitungsgeschwindigkeit ermöglicht wird.

Der IKE Quick Mode basiert auf keinem ISAKMP-Austauschtyp, sondern existiert nur im IKE-Protokoll und ist immer an einen vorangegangenen Phase-1-Austausch gebunden. Mit dem Quick Mode, der nur in der IKE-Phase 2, also unter dem Schutz einer in Phase 1 erzeugten ISAKMP-SA, möglich ist, werden die Sicherheitsassoziationen von IPSec oder anderen Sicherheitsprotokollen ausgehandelt. Innerhalb einer ISAKMP-SA können mehrere IPSec-SAs erzeugt werden. Es können auch innerhalb eines Quick-Mode-Austauschvorgangs gleichzeitig mehrere SAs erzeugt werden.

Da durch die ISAKMP-SA bereits Dienste zur Integritätsprüfung, Authentifizierung und Sicherstellung der Vertraulichkeit zur Verfügung stehen, kann der Austausch durch nur insgesamt drei Nachrichten erfolgen. Es werden im Weiteren auch die Identitäten der Gegenstellen als gegeben angesehen, die sich in Phase 1 gegenseitig authentifiziert haben. Der Quick Mode erzeugt eine Sicherheitsassoziation für nachfolgende Sicherheitsprotokolle und benutzt daher ähnliche Mechanismen, wie es die ISAKMP-Austauschvorgänge in der Phase 1 tun. Auch hier werden Nonce-Werte erzeugt und ausgetauscht, um sich einerseits vor Angriffen durch wiederholtes Senden von Nachrichten zu schützen und andererseits neue Zufallswerte zur Generierung von Schlüsselmateriale zu erhalten. Zu diesem Zweck kann optional, falls die Sicherheitsstrategie dies vorschreibt, ein Austausch von neu erzeugten öffentlichen Diffie-Hellman-Werten stattfinden.

Alle Pakete in Phase 2 werden verschlüsselt übertragen. Im ersten Paket des Initiators in unserem Beispiel, in dem eine IPSec-ESP-Sicherheitsassoziation mit Triple-DES und HMAC-MD5-96 erzeugt werden soll, müssen der Vorschlag (SA), der Nonce-Wert und der HASH_1 enthalten sein. Optional können noch Client-IDs (IDc) und im Fall von Perfect Forwarding Secrecy (PFS) der öffentliche Diffie-Hellman-Wert (KEi) übertragen werden.

Der Hash-Wert HASH_1 wird auf folgende Weise ermittelt:

$$\text{HASH\_1} = \text{HMAC-MD5}(\text{SKEYID\_a} + \text{MID} + \text{SA} + \text{Ni} [+ \text{KEi} [+ \text{Idci} + \text{IDcr}]])$$

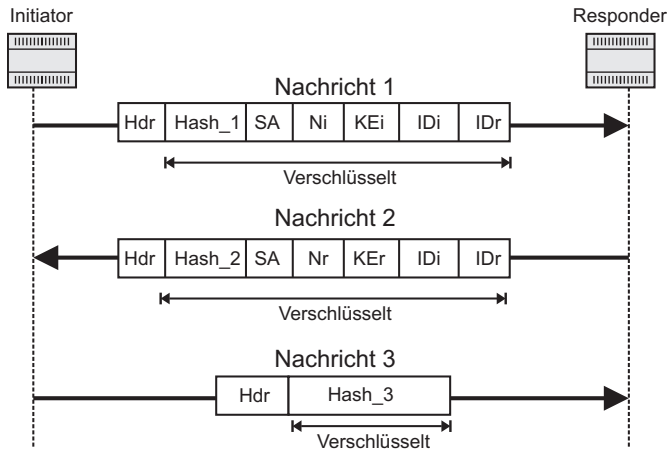


Abbildung 6.25: Der IKE-Quick-Mode-Austausch erfolgt unter dem Schutz der IKE-SA.

Die HMAC-Version des in Phase 1 bereits ausgehandelten MD5-Hash-Verfahrens wird als Pseudo Random Function (PRF) zum Erzeugen der Hash-Werte benutzt. MID ist hierbei die Message ID aus dem ISAKMP-Header; die optionalen Parameter sind in eckige Klammern gesetzt. Der Hash-Wert muss in jedem Paket unmittelbar auf den ISAKMP-Header folgen, und die SA-Nutzdaten in den ersten beiden Nachrichten müssen nach dem Hash-Wert eingefügt werden. Die Reihenfolge der restlichen Nutzdaten ist optional, mit Ausnahme der Client-IDs, die am Ende einer Nachricht stehen. Diese Client-IDs werden dann benutzt, wenn für einen anderen Initiator und Responder, als den in Phase 1 beteiligten, ein Quick-Mode-Austausch erfolgt.

Der Responder verarbeitet die eingegangene Nachricht und konstruiert, falls kein Fehler aufgetreten ist und er einen Vorschlag des Initiators akzeptabel findet, die zweite Nachricht dieses Austauschvorgangs. Der Hash-Wert HASH_2 dieser Nachricht wird, je nach Nutzdaten, wie folgt erzeugt:

$$\text{HASH\_2} = \text{HMAC-MD5}(\text{SKEYID\_a} + \text{MID} + \text{Ni} + \text{SA} + \text{Nr} [+ \text{KEr} [+ \text{IDcr}]])$$

Auch hier kann im Falle von PFS optional der öffentliche Diffie-Hellman-Wert mit übertragen werden.

Der Initiator verarbeitet die Nachricht und erzeugt die dritte und letzte Nachricht dieses Austauschvorgangs, in der nur der HASH_3 enthalten ist. Dieser Hash-Wert wird aus den folgenden Werten erzeugt:

$$\text{HASH\_3} = \text{HMAC-MD5}(\text{SKEYID\_a} + 0x00 + \text{MID} + \text{Ni} + \text{Nr})$$

Nachdem der Responder die Nachricht verarbeitet und den Hash-Wert überprüft hat, ist der Quick-Mode-Austausch abgeschlossen.

Der Quick Mode berechnet auch die für die erzeugte IPSec-Sicherheitsassoziation benötigten Schlüssel. In unserem Beispiel wurde das IPSec-ESP-Protokoll ausgehandelt, mit HMAC-MD5-96 und Triple-DES als Funktionen zur Authentifizierung und Verschlüsselung. Insgesamt werden hierfür 320 Bit an Schlüsselmaterial benötigt. Diese 320 Bits bestehen aus 128 Bit für HMAC-MD5-96 und dreimal 64 Bit für Triple-DES. Beachten Sie,

dass der Triple-DES-Algorithmus seine drei 56-Bit-Schlüssel aus den drei 64-Bit-Strings dadurch erzeugt, dass er jedes achte Bit daraus entfernt. Der in diesem Beispiel benötigte Grundschlüssel KEYMAT kann, je nachdem, ob PFS gefordert ist, auf zwei verschiedene Arten erzeugt werden. Im Fall von PFS wurden im Quick Mode öffentliche Diffie-Hellman-Werte zwischen Initiator und Responder ausgetauscht, die mit in die Berechnung einfließen. Aus diesen beiden Werten und ihrem privaten, geheimen Wert berechnen beide Seiten den identischen Wert KE als Ergebnis des Diffie-Hellman-Verfahrens. Der Grundschlüssel mit PFS wird wie folgt berechnet:

$$\text{KEYMAT} = \text{HMAC-MD5}(\text{SKEYID\_d} + \text{KE} + \text{Protokoll} + \text{SPI} + \text{Ni} + \text{Nr})$$

Die Werte für das Protokoll, in unserem Fall IPSec-ESP, und den Security Parameter Index (SPI) stammen aus dem Paket, in dem der Vorschlag für die Sicherheitsassoziation ausgewählt wurde.

Das Ergebnis der HMAC-MD5-Funktion ist aber nur 128 Bit lang, also muss das benötigte 320 Bit lange Schlüsselmateriale (K) durch Verkettung dieses Wertes erzeugt werden:

$$K = K1 + K2 + K3 \text{ mit}$$
$$K1 = \text{HMAC-MD5}(\text{SKEYID\_d} + \text{KE} + \text{Protokoll} + \text{SPI} + \text{Ni} + \text{Nr})$$
$$K2 = \text{HMAC-MD5}(\text{SKEYID\_d} + K1 + \text{KE} + \text{Protokoll} + \text{SPI} + \text{Ni} + \text{Nr})$$
$$K3 = \text{HMAC-MD5}(\text{SKEYID\_d} + K2 + \text{KE} + \text{Protokoll} + \text{SPI} + \text{Ni} + \text{Nr})$$

Falls keine Perfect Forwarding Secrecy benötigt wird, erzeugt IKE den Grundschlüssel aus dem dafür vorgesehenen Grundschlüssel (SKEYID_d) der ISAKMP-Sicherheitsassoziation und den Nonce-Werten des vorliegenden Quick-Mode-Austauschs:

$$\text{KEYMAT} = \text{HMAC-MD5}(\text{SKEYID\_d} + \text{Protokoll} + \text{SPI} + \text{Ni} + \text{Nr})$$

Das Ergebnis der HMAC-MD5-Funktion ist auch in diesem Fall nur 128 Bit lang, also erfolgt wieder eine Verkettung:

$$K = K1 + K2 + K3 \text{ mit}$$
$$K1 = \text{HMAC-MD5}(\text{SKEYID\_d} + \text{Protokoll} + \text{SPI} + \text{Ni} + \text{Nr})$$
$$K2 = \text{HMAC-MD5}(\text{SKEYID\_d} + K1 + \text{Protokoll} + \text{SPI} + \text{Ni} + \text{Nr})$$
$$K3 = \text{HMAC-MD5}(\text{SKEYID\_d} + K2 + \text{Protokoll} + \text{SPI} + \text{Ni} + \text{Nr})$$

Aus diesen 384 Bit werden für das IPSec-ESP-Protokoll die ersten 192 Bit für Triple-DES und die folgenden 128 Bit für den HMAC-MD5-Algorithmus verwendet; der Rest bleibt ungenutzt.

Wie in den vorangegangenen Abschnitten deutlich wurde, wird in IKE eine ganze Reihe teilweise recht aufwändiger Rechenoperationen durchgeführt. Ihre Anzahl und Art richten sich ausschließlich nach der verwendeten Sicherheitsstrategie. So ist zum Beispiel der Aufbau einer IPSec-Sicherheitsassoziation unter Verwendung von Public-Key-Verfahren zur Authentifizierung und Perfect Forwarding Secrecy sehr aufwändig und langwierig zu berechnen, während der gleiche Vorgang mit einer Pre-Shared-Key-Authentifizierung ohne PFS um ein Vielfaches schneller ist.

Man muss sich immer vor Augen halten, in welchen Dimensionen hier gerechnet wird. Asymmetrische (Public-Key-)Verfahren wie Diffie-Hellman oder RSA erzeugen und rechnen mit Primzahlen in Bereichen von 768 bis 1.536 Bit, führen Berechnungen mit Exponenten im Bereich von 180 Dezimalstellen durch oder dividieren durch 1.536 Bit große Primzahlen. Solche Operationen, die zudem nicht mit voller Unterstützung von numerischen Coprozessoren auf Basis von Gleitkomma-Arithmetik durchgeführt werden können, verschlingen eine immense CPU-Zeit und auch einiges an Speicher für ihre Zwischenergebnisse.

Auf Security-Gateways, die verschiedene Netze miteinander verbinden, fällt IKE nicht so sehr ins Gewicht, da es nur relativ selten aktiv ist und wenige SAs erzeugt und verwaltet. Hier kann man auch bedenkenlos die Intervalle für eine neue Schlüsselgenerierung auf kurze Werte oder übertragene Datenvolumina, z.B. 100 MByte oder 1 GByte, setzen und generell mit Perfect Forwarding Secrecy arbeiten.

Im Fall von Remote-Access-VPN sieht die Sache allerdings schon ganz anders aus, denn hier müssen oft in relativ kurzen Zeitspannen sehr viele Sicherheitsassoziationen erzeugt werden – pro Benutzer werden ja mindestens eine ISAKMP-SA und zwei unidirektionale IPSec-SAs erzeugt. Bei solchen Abschätzungen muss man ebenfalls berücksichtigen, dass solche Anmeldevorgänge auch bereits bestehende Verbindungen beeinflussen, indem sie die dort benötigten Ressourcen abziehen. Der magische Wert für Remote-Access-VPN-Konzentratoren, über den sich fast alle Hersteller leider beharrlich totschweigen, ist denn auch nicht so sehr der Datendurchsatz als solcher, sondern die Angabe, wie viele Anmeldungen pro Minute erfolgen können. Interessant ist in diesem Zusammenhang auch die etwas ketzerische Frage, ob während dieser Zeitspanne die 1000 vielleicht bereits angemeldeten Benutzer überhaupt noch Daten übertragen können oder ob der VPN-Konzentrator nur noch mit IKE-Funktionen beschäftigt ist.

Der Datendurchsatz ist natürlich auch ein wichtiger Faktor, aber er ist durch den Hersteller am leichtesten in den Griff zu bekommen. Meist setzt man Beschleuniger-Hardware ein, welche die symmetrischen und die Hash-Verfahren auf speziellen Chips abarbeitet. Für Public-Key-Verfahren gibt es leider nur ganz wenige Chipsätze und diese besitzen oft auch nicht den Beschleunigungsfaktor von Chips zur Verarbeitung von symmetrischen Algorithmen. Also werden in sehr vielen Systemen, auch solchen mit Hardware-Verschlüsselung, gerade die rechenintensiven Public-Key-Verfahren immer noch per Software implementiert.

An dieser Stelle ein Tipp, falls Sie vor der Auswahl oder Beurteilung von Hardware- gegenüber Software-Verschlüsselung stehen: Sehr oft wird damit geworben, dass Hardware-Verschlüsselung schneller als deren Software-Variante sei. Dies stimmt nicht immer, es ist zwar sehr oft der Fall, insbesondere bei Algorithmen wie DES, die bereits als Hardware-Verfahren entwickelt wurden, aber eben nicht immer. Algorithmen, die speziell zur Implementierung in Software entworfen und optimiert worden sind, können unter Umständen sogar an Performance einbüßen, da die Verwaltung der Beschleunigertypen durch das System sowie die Prozess- und die Datenverteilung eine gewisse Zusatzlast erzeugen.

Man muss weiterhin sehr genau untersuchen, welche Protokolle die Hardware überhaupt unterstützt. Je größer die Protokollvielfalt, desto besser. Was aber auf alle Fälle unterstützt werden sollte, ist eine Hardware-Datenkompression, da die Kompression auf den Schichten unterhalb von IPSec sinnlos ist, da verschlüsselte Daten nicht mehr komprimiert werden können.

Es sollte im Idealfall auch möglich sein, die Performance eines VPN-Konzentrators durch den Einsatz von mehreren, parallel betriebenen Hardware-Beschleunigern skalieren zu können.

Die ideale Hardware würde RSA, Diffie-Hellman, DES, Triple-DES, AES, MD5, SHA1 und IPSec-Kompression mit entsprechenden Chipsätzen unterstützen, so dass das Muttersystem die rechenintensiven Prozesse auf die Karten auslagern kann. Insbesondere für IKE sollten Diffie-Hellman und RSA durch spezielle Chipsätze bearbeitet werden können, ansonsten bringt eine Beschleunigerkarte keinen nennenswerten Performance-Gewinn.

Ein kleines Beispiel soll verdeutlichen, welche Leistungssteigerung gute Beschleunigerchips bringen können. Die kalifornische Firma Hi/fn stellt verschiedene Chips her, die speziell zum Beschleunigen von kryptografischen Algorithmen entwickelt wurden. Zum Vergleich nehmen wir zwei Chips dieser Firma. Einer davon, der Hi/fn 6500, besitzt ein spezielles Maskendesign zur Beschleunigung von modularer Arithmetik. Der andere Chip, der Hi/fn 7901, ist ein spezieller Beschleuniger für symmetrische Verfahren, der „nur“ eine normale Integer-CPU für Public-Key-Verfahren eingebaut hat.

	Hi/fn 7901, Integer-CPU	Hi/fn 6500, Public-Key-ASIC

Tabelle 6.4: Vergleich zwischen Integer-CPU und ASIC-Design zur Beschleunigung von Public-Key-Arithmetik

Zum Vergleich habe ich zwei verschiedene IKE-Aushandlungen herangezogen, einmal eine Authentifizierung mit Pre-Shared Key, die zwei Public-Key-Berechnungen (Diffie-Hellman, 1024 Bit) benötigt, und zum anderen eine Authentifizierung mit digitaler Signatur. Der zweite Fall ist wesentlich aufwändiger. Er benötigt pro Gegenstelle zwei 1024-Bit-Diffie-Hellman-Berechnungen (Oakley-Gruppe.2), eine 1024-Bit-RSA-Verschlüsselung und zwei 1024-Bit-RSA-Entschlüsselungen.

Man kann sehr gut erkennen, wie groß der Unterschied zwischen einer „normalen“ Integer-CPU und einem speziellen Maskendesign für bestimmte Public-Key-Algorithmen ist. Falls Sie vor der Auswahl von Hardware-Beschleunigern stehen, sollten Sie sich sehr genau darüber informieren, wie die Karte und die verwendeten Chips aufgebaut sind, ansonsten können Sie leicht unangenehme Überraschungen erleben. Als sehr sinnvoll erweisen sich Designs mit getrennten Chips für symmetrische und asymmetrische Verfahren oder neue „Super-Chips“, die beide Methoden in Form eines entsprechenden Maskendesigns unterstützen. Ein solcher Chip ist beispielsweise der 7811 von HiFn, der neben AES-Hardware-Unterstützung auch Diffie-Hellman (bis 1536 Bit) und die Erzeugung echter Zufallszahlen in Hardware realisiert hat.

6.10 NAT mit IKE und IPSec

Bevor wir uns mit den zwei Standards zur Erkennung von Network Address Translation (NAT) zwischen Initiator und Responder befassen, möchte ich noch einen kurzen Überblick über NAT-Problematik in IKE und vor allem IPSec geben.

6.10.1 NAT und IPSec

NAT-Systeme, insbesondere auch 1-zu-N-NAT mit Port Address Translation (PAT), werden sehr oft in kleinen Standorten oder Heimbüros eingesetzt. Dies hat vor allem finanzielle Gründe, denn ein Internetanschluss mit nur einer einzigen, am besten noch dynamischen IP-Adresse ist die mit Abstand günstigste Variante, um ins Netz zu kommen. Das Problem dabei ist aber, dass in den kleinen Standorten meist mehr als ein Endgerät existiert, man also intern mehrere Geräte hat, die sich eine einzige offizielle IP-Adresse teilen müssen. Die Lösung sind so genannte NAT/PAT-Systeme, welche die Source-IP-Adressen durch die offizielle IP-Adresse ersetzen (NAT) und die Source-Portnummer von UDP oder TCP durch eine andere, wahlfreie Nummer ersetzen (PAT). Der NAT-Router führt intern eine Tabelle, um über die geänderten Source-Portnummern bei zurückkommenden Paketen wieder die korrekte IP-Adresse ermitteln zu können. Zu dieser Tabelle wird auch für jeden Eintrag ein Timer verwaltet, der dafür sorgt, dass nach einer bestimmten Zeit der Inaktivität der betreffende Eintrag gelöscht wird. Diese Lösung hat aber auch einige Nachteile:

Pro Service (z.B. Mail-, FTP- oder Webserver) kann jeweils nur ein einziges System von der Seite des Internets erreicht werden. Hierbei muss ein spezielles Port-Forwarding im NAT-Router eingerichtet werden.

Der Inaktivitäts-Timer kann auch Verbindungen unterbrechen, die nur sehr unregelmäßig Daten austauschen, was zu einem Abbruch bestimmter Applikationen führen kann.

Das Ganze funktioniert nur bei TCP und UDP – und nicht bei IPSec-ESP, was natürlich den Einsatz von VPN-Clients oder Gateways hinter einem NAT/PAT-Router erst einmal ausschließt.

Es gibt zwei Lösungsansätze, welche die beiden letzten Probleme umschiffen können, von denen allerdings nur einer uneingeschränkt verwendbar ist. Die schlechtere Lösung ist die, dass der NAT-Router „IPSec-aware“ ist, also IPSec erkennen kann, in dem Fall keine Ports ändert (es sind ja auch keine da), sondern eine Tabelle führt, in der statt der Portnummern der SPI (Security Parameter Index) mit den lokalen IP-Adressen verknüpft ist. Diese Lösung setzt voraus, dass alle möglichen NAT/PAT-Router dieses Feature besitzen. Dies ist keineswegs der Fall, und bei denen, die es tun, sind verschiedene, inkompatible Lösungen im Einsatz.

Die bessere Lösung ist die, dass IKE und IPSec sich selbst der Sache annehmen und somit unabhängig von allen Infrastruktur-Komponenten bleiben.

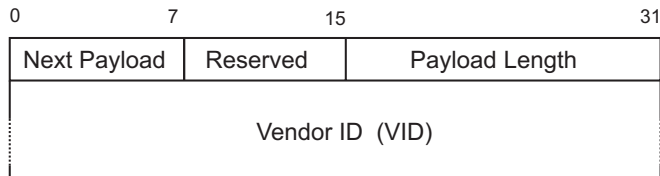


Abbildung 6.26: Die Vendor ID in IKE gibt an, welche Funktionen eine Gegenstelle unterstützt.

6.10.2 Automatische Erkennung von NAT-Routern

Die Lösung dieses Problems wurde von der IP Security-Arbeitsgruppe der IETF zweigeteilt angelegt. Das IKE-Protokoll erhielt eine Erweiterung (RFC 3947, Negotiation of NAT-Traversal in the IKE), um während des Aufbaus einer IKE Security Association zu erkennen, ob ein NAT-System zwischen Initiator und Responder arbeitet, und im positiven Fall eine Einkapselung der IPSec-ESP-Pakete in UDP auszuhandeln. Die zweite Stufe ist die Einkapselung der IPSec-Pakete in UDP-Datagramme (RFC3948, UDP Encapsulation of IPSec ESP Packets), was im Prinzip nichts anderes ist, als zwischen IP-Header und ESP-Header einen UDP-Header einzufügen. Dieses Verfahren bezeichnet man teilweise auch als IPSec NAT Traversal.

Wichtig in diesem Zusammenhang ist auch das Senden von so genannten NAT-Keep-Alive-Paketen, die dafür sorgen, dass die Tabellen im NAT-Router nicht aufgrund von eventuellen Timeouts gelöscht werden. Beim gleichzeitigen Einsatz von TCP-Applikationen und IPSec-Transport-Mode müssen der Gegenstelle auch die originalen IP-Adressen bekannt sein, damit die fortlaufenden TCP-Prüfsummen korrekt berechnet werden können. Als absolute Grundvoraussetzung muss die IKE-Implementierung natürlich auch in der Lage sein, IKE-Pakete zu verarbeiten, in denen die Source-Portnummer nicht mehr, wie von im Standard empfohlen, 500 ist.

Der Responder muss weiterhin einen entsprechenden Port geöffnet haben, der auf der Portnummer hört, die für die UDP-Encapsulation reserviert wurde. Die IANA (Internet Assigned Numbers Authority) hat hierfür die Portnummern tcp/4500 und udp/4500 reserviert.

Dieser gesamte Prozess gliedert sich in verschiedene Funktionsblöcke:

- Die Ermittlung, ob sowohl Initiator als auch Responder das hier beschriebene Verfahren überhaupt beherrschen.
- Die Erkennung von NAT-Systemen entlang des Übertragungsweges.
- Die Aktivierung der NAT-Keep-Alive-Funktion auf der richtigen Seite.
- Die Aktivierung von NAT-Traversal und gegebenenfalls Übertragung der originalen IP-Adressen im Quick Mode, falls es sich um eine IPSec-Transport-Mode-Verbindung handelt.

Beide Gegenstellen teilen sich in dem jeweils ersten gesendeten Paket über eine spezielle Vendor-ID Payload gegenseitig mit, ob sie den RFC3947 beherrschen. Die Vendor-ID entstammt dem MD5-Hash-Wert des Strings „RFC 3947“.

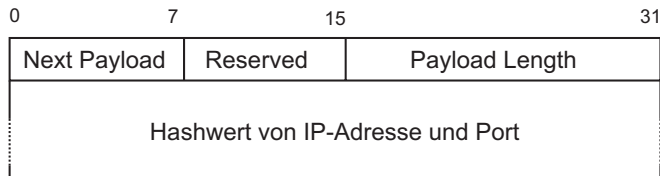


Abbildung 6.27: Die NAT Discovery (NAT-D) Payload.

Die NAT-Erkennung ist so ausgelegt, dass nicht nur erkannt wird, ob sich ein aktiver NAT-Router im Übertragungsweg befindet, sondern auch auf welcher Seite er arbeitet. Das ist wichtig, denn die NAT-Keep-Alive-Pakete müssen unbedingt auf der Seite erzeugt werden, die hinter dem NAT-System liegt. Um den gesamten Ablauf zu verstehen, beschäftigen wir uns zuerst einmal mit dem IKE Main Mode.

NAT-Erkennung im IKE Main Mode

Der Initiator sendet seine ISAKMP-Exchange (UDP/500/500) zum Responder. In diesem Paket ist zusätzlich die Vendor ID auf den oben beschriebenen Wert gesetzt, damit die Gegenseite über die RFC3947-Kapazität der Gegenstelle informiert wird. Falls ein aktiver NAT-Router zwischen den beiden liegt, kann das Paket verändert werden, in diesem Fall würde die Source-Portnummer auf einen anderen Wert, nehmen wir einmal 12345, gesetzt.

Der Responder antwortet ganz normal, fügt jedoch ebenfalls seine entsprechende Vendor ID Payload ein und schickt das Paket an die Absender-IP-Adresse mit 12345 als Destination-Portnummer und 500 als Sourceport.

Nun wissen beide Seiten, dass sie den RFC3947 unterstützen, und können prüfen, ob und vor allem wo ein NAT-Router die Pakete manipuliert. Zu diesem Zweck werden in den beiden folgenden Exchanges (Key Exchange) zwei oder mehrere zusätzliche Payloads übertragen, die so genannten NAT Discovery Payloads (NAT-D), deren Format in Abbildung 6.27 illustriert ist. Der Hash-Wert darin wird wie folgt berechnet:

Die erste NAT-D-Payload enthält immer die IP-Adresse der Gegenseite (Remote NAT-D Payload) und die folgende Payload die eigene (Local NAT-D Payload). Falls ein System mehrere Interfaces besitzt und es nicht vorher bestimmbar ist, über welches die Pakete letztendlich verschickt werden, können der zweiten NAT-D Payload weitere Local NAT-D Payloads für die weiteren Interfaces folgen.

Der Initiator schickt nun seine entsprechend erweiterte Key-Exchange zum Responder. Der Responder überprüft, ob die erste NAT-D Payload mit einer seiner Local NAT-D Payloads übereinstimmt. Wenn ja, dann hat kein NAT stattgefunden. Stimmen sie nicht überein, dann muss der Responder hinter einem NAT-Router liegen. In diesem Fall muss der Responder anfangen, NAT-Keep-Alive-Pakete zu senden. Diese Keep-Alive-Pakete werden von der Gegenseite nicht weiter ausgewertet oder verarbeitet, sie dienen nur dazu, die Umsetzungsstabellen im NAT-Router regelmäßig zu aktualisieren.

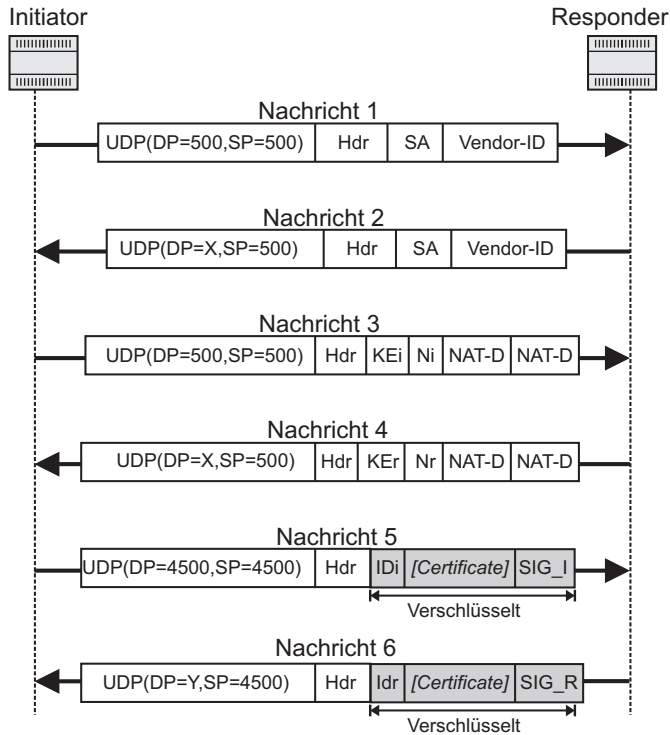


Abbildung 6.28: NAT Discovery im IKE Main Mode

Der Responder antwortet nun mit seiner Key Exchange, erweitert um seine NAT-D Payloads. Der Initiator führt nach Erhalt des Pakets ebenfalls die gleichen Prüfungen durch und reagiert entsprechend.

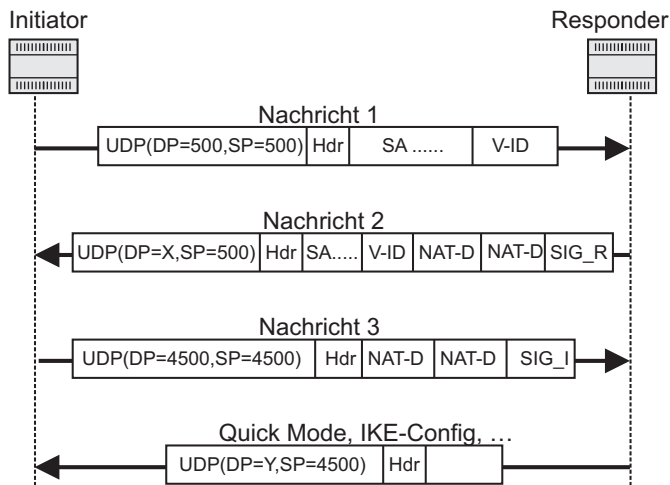


Abbildung 6.29: NAT Discovery im IKE Aggressive Mode

Die beiden letzten Exchanges (Authentifizierung) werden normal übertragen, allerdings bereits mit den geänderten Portnummern, insbesondere werden statt der normalen Portnummern (udp/500) die Nummern für UDP-Encapsulation (udp/4500) verwendet. Es ist zu beachten, dass der NAT/PAT-Router für die Portnummer 4500 ein neues Mapping vornimmt (der Wert Y in den Abbildungen), das einen anderen Portwert benutzt als das Mapping von Port 500 (der Wert X in den Abbildungen).

NAT-Erkennung im IKE Aggressive Mode

Im IKE Aggressive Mode wird die Funktion der Erkennung in bereits in der zweiten und dritten Exchange durchgeführt. Der Initiator fügt die zusätzliche Vendor-ID Payload in sein erstes Paket ein.

Der Responder weiß nach Erhalt dieses Pakets, ob der Initiator den RFC3947 unterstützt, und packt im positiven Fall in sein erstes Paket neben der Vendor-ID Payload auch gleich seine NAT-D Payloads.

Der Initiator kann nun erkennen, ob er hinter einem NAT-Router angeordnet ist, und kann entsprechend reagieren. Er schickt anschließend sein letztes Paket (Authentifizierung), erweitert um seine NAT-D Payloads, zum Responder. Dieser kann nun auch ermitteln, ob er hinter einem NAT-System arbeitet.

Aktivierung der UDP Encapsulation

Um das Verhalten bestimmter NAT-Router, die versuchen, auf unterschiedliche Methoden mit IPSec klarzukommen, zu entschärfen, sollte sobald möglich auf die Ports 4500 gewechselt und NAT-Traversal nach RFC3948 durchgeführt werden. Im Main Mode kann dies ab dem vierten Austausch, im Aggressive Mode ab dem dritten geschehen.

Zu diesem Zeitpunkt, also nach Beendigung des Main oder Aggressive Mode, wissen beide Seiten, ob entlang der Übertragungsstrecke NAT durchgeführt wird oder nicht. Wenn dem so ist, verschickt die betroffene Seite bereits NAT-Keep-Alive-Pakete, und beide benutzen bereits UDP Encapsulation mit den Ports 4500.

Die Einkapselung erfolgt damit sowohl für IKE- als auch für IPSec-ESP-Pakete. Da in beiden Fällen die gleichen Ports verwendet werden, muss eine geeignete Methode dafür sorgen, dass IKE- und IPSec-Pakete getrennt verarbeitet werden.

NAT-OA

Um TCP- oder UDP-Prüfsummen verifizieren zu können, benötigen die Gegenstellen die IP-Adresse, die zum Zeitpunkt der Prüfsummenberechnung verwendet wurde. Da NAT diese Adresse jedoch geändert hat, gibt es hier im Fall von IPSec-Transportmode ein Problem. Um dieses zu lösen, wurde IKE erweitert, um im IKE Quick Mode die originalen IP-Adressen austauschen zu können.

Zu diesem Zweck wurde eine weitere Payload, die NAT-OA Payload (OA, Original Address), definiert. Falls der Initiator ausschließlich den IPSec-Transport Mode vorschlägt, müssen beide Seiten, sofern ein NAT-System zwischen Initiator und Responder erkannt wurde, ihre NAT-OA Payload austauschen. Dies muss im ersten und zweiten Paket des IKE Quick Mode geschehen.

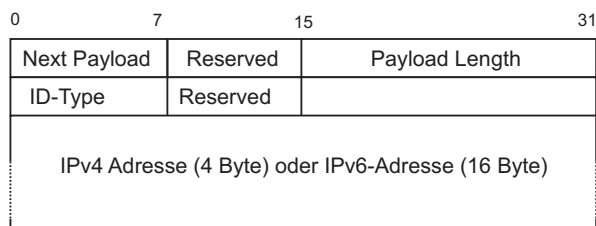


Abbildung 6.30: Die NAT Original Address (NAT-OA) Payload

Im Fall von IPSec-Tunnel-Mode sollten die Gegenstellen keine NAT-OA Payload austauschen, diese werden nicht benötigt, da der innere IP-Header geschützt ist und damit von NAT unberührt bleibt.

Falls der Initiator IPSec-Tunnel-Mode oder Transport-Mode vorschlägt, muss er auf alle Fälle seine NAT-OA Payload übertragen. In diesem Fall wählt der Responder den Tunnel-Mode aus und überträgt seine NAT-OA Payload nicht.

Automatische NAT-Erkennung und Mobile IKE (MobIKE)

Normalerweise arbeiten die meisten Implementierungen mittlerweile mit automatischer NAT-Erkennung, denn es macht keinen Sinn, UDP-Einkapselung einzusetzen, ohne dass ein NAT-Router im Übertragungsweg liegt. Es gibt jedoch bestimmte Fälle, in denen während des Verbindungsaufbaus (IKE) kein NAT-Router zwischen Initiator und Responder liegt, sich dies später jedoch (während einer existierenden SA) ändern kann. Wie kann das gehen? Ganz einfach, beim Roaming. Einige IPSec-Implementierungen unterstützen bereits einige Funktionen des kommenden MobIKE- Standards (Mobile IKE), der Roaming zwischen verschiedenen Medien und IP-Subnetzen unterstützt – ohne dass die IKE- oder IPSec-SA abgebrochen wird. In diesem Fall kann es vorkommen, dass eine IPSec-Verbindung in einem Netz ohne NAT (z.B. UMTS) gestartet wird und anschließend in eine Infrastruktur mit NAT (z.B. Heimbüro mit DSL und WLAN) übernommen wird. In diesem Fall würde die Verbindung trotz MobIKE abbrechen. Falls jedoch von IKE zwingend UDP-Encapsulation ausgehandelt wurde, unabhängig von der initialen Existenz oder Nichtexistenz eines NAT-Routers, bestehen die Security Associations weiter.

6.10.3 UDP Encapsulation von IPSec-ESP

Die UDP Encapsulation von IPSec-ESP-Paketen erfordert zwei Voraussetzungen

1. Der IKE-Responder hört auf einem von der IANA festgelegten, wohl bekannten UDP-Port.
2. Da sowohl für IKE- als auch für eingekapselten IPSec-ESP-Pakete die gleichen Portnummern verwendet werden, muss ein entsprechender Multiplex-Mechanismus implementiert werden.

Das erste Kriterium wird durch Standard-Portnummern (udp/4500, tcp/4500) erfüllt. Es gibt jedoch auch Implementierungen, in denen der Responder eigene, wahlfreie Portnummern benutzen kann und diese in einer IKE-Configuartion-Message der Gegenstelle übermittelt. Derartige Implementierungen arbeiten jedoch definitiv nicht konform zum RFC3947 und sind damit nicht interoperabel!

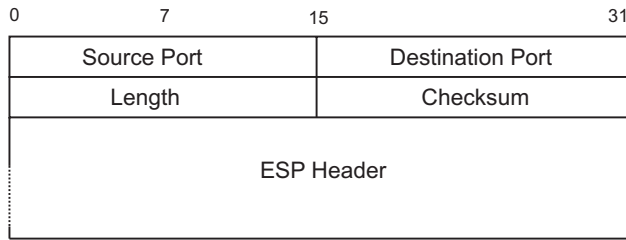


Abbildung 6.31: IPSec-ESP in UDP Encapsulation

Das Ziel des RFC3948, der die UDP Encapsulation von IPSec-ESP-Paketen beschreibt, ist es, diese zusätzliche Funktion möglichst leistungsfähig zu gestalten. So wurde nur ein einziger Port für IKE- und IPSec-Encapsulation gewählt, was den NAT/PAT-Router entlastet. Weiterhin werden die UDP-Prüfsummen nicht berechnet, außer im Fall von IKE-Paketen mit Port 4500, denn hier handelt es sich um normales UDP. Die NAT-Keep-Alive-Pakete werden, sobald sie als solche erkannt werden, vom Empfänger ignoriert. Sie dienen ausschließlich dazu, den entsprechenden Timer im NAT/PAT-Router zurückzusetzen.

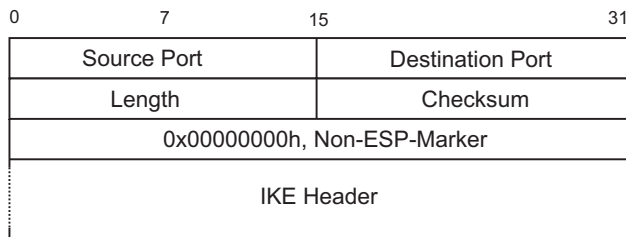


Abbildung 6.32: IKE in UDP-Port-4500-Paketen

IPSec-ESP-Einkapselung

Der UDP-Header entspricht hinsichtlich Größe und Format dem RFC786 (User Datagram Protocol). Source Port und Destination Port sind die gleichen, die auch vom IKE-Protokoll gemäß RFC3947 benutzt werden. Das UDP-Prüfsummenfeld sollte mit dem Wert 0 versehen sein. Falls ein UDP-Paket auf dem Port 4500 empfangen wird und der Wert des ersten, auf den UDP-Header folgenden Byte gleich 0xFFh ist, dann handelt es sich um ein NAT-Keep-Alive-Paket, das ignoriert werden kann. Andernfalls werden die vier Bytes nach dem UDP-Header ausgewertet. Ist der Wert des auf den UDP-Header folgenden 32-Bit-Wortes gleich 0, dann handelt es sich um ein IKE-Paket mit geänderten Portnummern und wird entsprechend weiter verarbeitet. Ist der 32-Bit-Wert ungleich 0, dann handelt es sich um den SPI eines IPSec-ESP-Headers und somit um ein eingekapseltes IPSec-ESP-Paket.

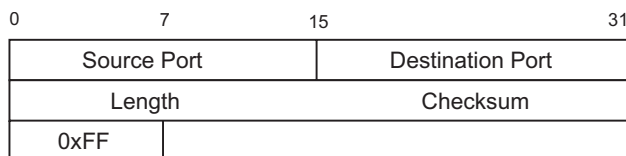


Abbildung 6.33: Ein NAT-Keep-Alive-Paket

Beim IPSec-Transport-Mode folgt dem ESP-Header in den meisten Fällen ein TCP- oder UDP-Header. Diese Header enthalten ein Prüfsummenfeld, in dem ein Wert steht, der unter anderem aus dem ursprünglichen IP-Header berechnet wurde. Diese Prüfsumme ist natürlich falsch, wenn ein NAT-Router Teile des IP-Headers verändert hat. In IPv4 muss die TCP-Prüfsumme zwingend ausgewertet werden, in IPv6 müssen TCP- und UDP-Prüfsumme ausgewertet werden. Damit dies in diesem Fall auch geschehen kann, kann im IKE Quick Mode vom Initiator die Original-IP-Adresse zum Responder übermittelt werden. Aufgrund dieser Information können jetzt die korrekten Prüfsummen berechnet werden. Im IPSec-Tunnel-Mode oder bei L2TP-over-IPSec-Transport (siehe Kapitel 8) gibt es diese Problematik nicht, hier ist ja das komplette private IP-Paket in einem anderen Paket enthalten.

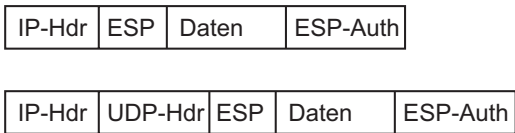


Abbildung 6.34: UDP Encapsulation im IPSec Transport Mode

NAT Keep Alive

Es wurde bereits mehrfach auf die NAT-Keep-Alive-Pakete verwiesen, die dafür sorgen sollen, dass die Umsetzungstabelle im NAT-Router den entsprechenden Eintrag nicht löscht, weil in einem bestimmten Zeitlimit keine Pakete übertragen werden. Die Funktion soll so implementiert werden, dass nicht unnötig Pakete geschickt werden. Der voreingestellte Wert ist ein Intervall von 20 Sekunden, aber nur dann, wenn nicht ohnehin schon Daten zur Gegenseite gesendet wurden. Kurzum, es dürfen nie mehr als 20 Sekunden ohne Datenverkehr zur Gegenseite vergehen. Der Wert sollte natürlich änderbar sein.

Probleme

Probleme kann es auch hier unter gewissen Bedingungen geben. So sind ironischerweise vor allem NAT/PAT-Router, die mit IPSec-ESP umgehen können, oft Schuld, wenn keine Verbindung aufgebaut werden kann. Denn deren von allen Standards abweichende, in jedem Modell anders implementierte „IPSec-Awareness“ gerät nicht selten in Konflikt mit NAT-Discovery und UDP-Encapsulation. Im Zweifelsfall sollte man diese Funktion deaktivieren und, wenn dies nicht geht, dann einen „dummen“ NAT/PAT-Router einsetzen, falls dies möglich ist. Manche Hersteller haben auch spezielle Konfigurationsoptionen, um mit solchen Systemen umgehen zu können, jedoch funktionieren diese auch nicht immer zuverlässig.

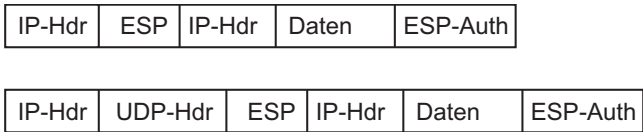


Abbildung 6.35: UDP Encapsulation im IPSec Tunnel-Mode

6.11 IKE Dead Peer Detection (DPD)

IPSec (ESP oder AH) kann aufgrund der Tatsache, dass es ein zustandsloses IP-Protokoll ist, nicht erkennen, ob eine Gegenstelle erreichbar ist, außer wenn diese Pakete sendet oder per IKE Delete Message mitteilt, dass sie die Verbindung beendet. Gesendete IPSec-Pakete werden nicht bestätigt, und wenn keine Pakete ankommen, kann dies bedeuten, dass entweder keine gesendet wurden oder dass die Gegenstelle nicht mehr erreichbar ist. Die SAs werden in letzterem Fall so lange aufrecht erhalten, bis ihre maximale Lebensdauer (Default: acht Stunden) abgelaufen ist.

6.11.1 DPD nach RFC3706

Da dieses Problem in der Vergangenheit mit verschiedenen herstellerspezifischen Verfahren gelöst wurde, hat die IETF einen informellen Standard, den RFC3706, verabschiedet, der empfiehlt, wie man eine interoperable Lösung implementieren sollte.

Die Ziele einer solchen Lösung sind folgende:

- Die Gegenstellen müssen gegenseitig erkennen, ob sie DPD unterstützen.

- Man möchte in einer frei konfigurierbaren Zeitspanne einen Ausfall der Verbindung oder der Gegenstelle erkennen können.

- Man möchte unnötigen Verkehr minimieren, damit die Lösung auch skalierbar ist.

- Die Lösung muss sicher gegen Angriffe sein, die z.B. die Erreichbarkeit einer in Wirklichkeit nicht mehr erreichbaren Gegenstelle vorspiegeln.

- Eine Gegenstelle soll, abhängig von ihrer IPSec/IKE-Implementierung, auch vor dem Senden von IPSec-Paketen prüfen können, ob die Gegenseite erreichbar ist.

DPD benutzt einen bidirektionalen Austausch von IKE-Nachrichten, deren Zeitlimit auf beiden Seiten einer Verbindung beliebig und asynchron konfigurierbar sein kann.

Um zuverlässig arbeiten zu können, müssen sich beide Seiten vorher über ihre RFC3706-Kapazität verständigt haben. Dies geschieht, wie in IKE üblich, durch gegenseitiges Zusenden der entsprechenden Vendor ID. Diese Vendor ID besteht aus der Bytefolge {0xAF, 0xCA, 0xD7, 0x13, 0x68, 0xA1, 0xF1, 0xC9, 0x6B, 0x86, 0x96, 0xFC, 0x77, 0x57}, einem Byte als Major Version Number und einem Byte als Minor Version Number. Die zur Drucklegung dieses Buches aktuelle Version von DPD ist die Version 1.0.

Ein normaler DPD-Austausch besteht aus zwei authentifizierten IKE Notify Exchanges, einer R-U-THERE- und einer darauf folgenden R-U-THERE-ACK-Nachricht. Das IKE-Notify-Paket enthält dabei folgende, aus dem privaten IANA-Wertebereich entnommene Notify-Nummern:

- 36136 (R-U-THERE)

- 36137 (R-U-THERE-ACK)

Die restlichen Felder entsprechen dem ISAKMP-Standard. Im Feld Notification Data ist der aktuelle Wert des Sequenzzählers eingetragen. Durch dieses Konstrukt werden keine Protokolländerungen an IKE notwendig, und es werden auch keine verbindlichen IANA-Nummern benötigt.

Der Austausch ist nicht rollengebunden und kann sowohl vom Initiator als auch vom Responder initiiert werden. Antwortet die Gegenseite nicht innerhalb einer implementierungsabhängigen Zeitspanne, sollte die Anfrage wiederholt werden. Wenn nach einer bestimmten Anzahl von Wiederholungen innerhalb eines Zeitintervalls keine Antwort kommt, sollte der Sender annehmen, dass die Gegenstelle nicht mehr erreichbar ist, und entsprechende Maßnahmen treffen. In der Regel werden dann immer die IKE- und IPSec-SAs gelöscht und, je nach Situation, versucht, die Verbindung neu aufzubauen. In den meisten IKE-Implementierungen sind die Anzahl der Wiederholungen, der Zeitabstand dazwischen und bei manchen Systemen auch der Timeout-Wert zur Entscheidung, ob ein Paket beantwortet ist oder nicht, frei konfigurierbar.

Der Standard empfiehlt, DPD so zu implementieren, dass nur dann R-U-THERE-Nachrichten versendet werden, wenn eine bestimmte Zeit lang keinerlei IPSec- oder IKE-Pakete von der Gegenseite ankommen. Damit werden unnötige Sendungen vermieden.

Wenn eine Gegenstelle mit aktivem DPD eine korrekte R-U-THERE-Nachricht erhält, muss sie diese mit einem R-U-THERE-ACK beantworten. Korrekt bedeutet hier, dass es sich um ein verschlüsseltes, authentifiziertes IKE-Paket mit einem korrekten Sequenznummerwert handeln muss. Nicht verschlüsselte Pakete, zum Beispiel mit einem nicht gesetztem E-Flag im ISAKMP-Header, müssen ignoriert werden.

Um sich gegen Denial-of-Service-Angriffe zu schützen, wird neben der Sicherheit von IKE zusätzlich noch ein Sequenznummer geführt. Damit hat man eine fast mit IPSec-ESP vergleichbare Sicherheit für die DPD-Nachrichten. Bisher sind für die IKE Dead Peer Detection keine Verwundbarkeiten oder Angriffe bekannt.

6.11.2 Andere DPD-Verfahren

Es existieren auch noch einige andere, herstellereigene DPD-Verfahren, die sich oft nicht so sehr voneinander unterscheiden. Das, was allen eigen ist: Bis auf eine Ausnahme sind sie nicht miteinander kompatibel!

Unidirektionale Verfahren

Diese Verfahren basieren darauf, dass eine Gegenstelle periodisch Pakete schickt. Dies wird auch als Heartbeat (Herzschlag) bezeichnet. Diese Verfahren bedingen entweder statische Implementierungen oder eine entsprechende manuelle oder per IKE-Config übertragene, zueinander passende Konfiguration auf beiden Seiten einer Verbindung.

Bidirektionale Verfahren

Hier erzeugt ebenfalls eine Seite Keep-Alive-Anfragen, wartet jedoch auf deren Beantwortung. Solche Verfahren funktionieren praktisch wie DPD nach RFC3706, haben jedoch bestimmte herstellerabhängige Spezifika, die sie nicht miteinander interoperabel machen.

Es gibt jedoch eine Ausnahme, die auf dem Re-Keying-Mechanismus beruht. Der Trick ist einfach der, dass jede IPSec-Gegenstelle auf eine Re-Keying-Nachricht mit einem vollständigen Schlüsseltausch reagieren muss. Das ist in den originalen Standards festgelegt. DPD arbeitet nun so, dass nach einer bestimmten Zeit, in der keine Pakete von der

Gegenstelle mehr kommen, eine ISAKMP-Re-Key-Nachricht zur Gegenseite geschickt wird. Auf diese muss reagiert werden, ansonsten ist die Verbindung unterbrochen oder die Gegenstelle nicht mehr existent. Der Vorteil ist, dass auf der Gegenseite eine beliebige standardkonforme IKE-Implementierung sein kann. Der Nachteil ist der, dass das Re-Keying einiges an Ressourcen benötigt und damit nicht besonders skalierbar ist.

6.12 Die Sicherheit von IKE

Wenn man sich dieses Themas annimmt, sollte man in jedem Fall zwei Dinge tun: Erstens die relevanten IKE-Standards lesen und auch verstehen und dann zweitens auch unterscheiden können, ob ein mögliches Sicherheitsproblem zwingend aus einem Standard, aufgrund einer standardfremden Erweiterung oder der Inkompetenz eines Herstellers herrührt.

Ziemlich kontraproduktiv in diesem Sinne sind einige ziemlich an den Haaren herbeigezogenen „Sicherheitsmängel“, die man in IKE gefunden haben will. Meist handelt es sich dabei um **mögliche** Probleme, auf die aber explizit im Standard hingewiesen wurde, oder um herstellerspezifische Erweiterungen von IKE, die potenzielle Angriffspunkte bieten können, jedoch für den Großteil der VPN-Anwender irrelevant sind, weil diese einfach ein anderes Produkt einsetzen. Leider tun sich auch gewisse Personen, deren Standardisierungsvorschläge seinerzeit von der IETF abgewiesen wurden, durch penetrante Nörgelei in verschiedenen Foren und Publikationen hervor. Und zu guter Letzt gibt es die ganzen Trittbrettfahrer, die mittlerweile auf den Security-Zug aufgesprungen sind, weil damit gerade ein gutes Geschäft gemacht werden kann. Da wird dann von IKE-Sicherheitsproblemen fabuliert, werden Leute wie Whitfield Diffie völlig aus dem Zusammenhang gerissen zitiert, und was wird dem dann völlig verunsicherten Kunden als Lösung angeboten? Ein simpler Router! Das ist nicht erfunden und leider kein Einzelfall.

Ich möchte deshalb an dieser Stelle, ausgehend von den aktuellen RFCs und Informationen der bekannten Security-Unternehmen und Security-Advisory-Organisationen, kurz die Schwächen und Stärken von IKE beleuchten, damit sich der Leser selbst ein Bild machen kann.

Der IKE Main Mode basiert auf der ISAKMP Identity Protection Exchange. Wie der Name andeutet, kann jemand, der einen IKE Main Mode vollständig mitliest, weder Informationen über die Identität der Gegenstellen erhalten, noch bekommt er für einen Offline-Wörterbuchangriff verwertbare Daten der IKE-Authentifizierung, denn diese erfolgt bereits verschlüsselt. Der Main Mode bietet vom Prinzip her eine sehr hohe Sicherheit, egal ob die Authentifizierung auf Basis von guten Pre-Shared Keys oder digitalen Signaturen erfolgt.

Wenn der PSK jedoch ein Benutzerkennwort ist, was typisch für Remote Access ist, dann kann IKE nicht sicherer sein als die Qualität und damit die Sicherheit des Passwortes. Insbesondere wenn mit IKE eine quasi-anonyme Authentifizierung erfolgt (siehe unten, XAUTH), dann ist einem Man-in-the-Middle-Angriff unter Umständen Tür und Tor geöffnet.

Kurzum, mit digitalen Signaturen oder guten Pre-Shared Keys ist der Main Mode sicher.

Nach wie vor ist bei den meisten Unternehmen keine PKI im Einsatz, also bleibt dann nur die Authentifizierung mit einem Pre-Shared Key. Bei Standortverbindungen (Gateway-zu-Gateway) ist das noch kein großes Problem, denn hier kann man bei statischen IP-Adressen mit dem Main Mode arbeiten. Auch gute PSK mit 20- oder 30-stelligen Hexzahlen, die jedem Wörterbuchangriff standhalten, sind in solch einem Szenario möglich.

Im Fall von Remote Access mit dynamisch zugewiesenen IP-Adressen (ISDN, DSL usw.) sieht es etwas schlechter aus, denn hier muss erstens der IKE Aggressive Mode benutzt werden, und zweitens ist das PSK in der Regel das Benutzerpasswort. Und dies kann natürlich keine 30-stellige Hexzahl sein, sondern eine vom Benutzer zu merkende Zeichenkette. Damit ist die Sicherheit von IKE auf die Sicherheit eines Benutzerpasswortes reduziert.

Das Hauptproblem beim IKE Aggressive Mode ist nun, dass es möglich ist, falls man eine vollständige Exchange aufzeichnen konnte, offline einen Wörterbuchangriff auf den Pre-Shared Key durchzuführen. Dies ist deshalb möglich, weil der HASH_R (vgl. Kap. 6.6.1) bis auf den PSK aus Komponenten berechnet wird, die vorher in den ersten beiden Nachrichten unverschlüsselt übertragen wurden. Und ein Offline-Wörterbuchangriff ist deshalb so gefährlich, weil er nicht erkannt werden kann. Allerdings kostet auch ein Offline-Wörterbuchangriff Ressourcen. Nur zum Vergleich, ein achtstelliges Benutzerpasswort (Bereich A–Z, a–z, 0–9) kann etwa $2,18 \cdot 10^{14}$ verschiedene Zustände annehmen, eine 30-stellige Hexzahl etwa $1,33 \cdot 10^{36}$. Ein Rechner mit einer Pentium-4-CPU mit 2,8 GHz würde für einen Wörterbuchangriff auf das achtstellige Kennwort alleine schon etwa 20 Jahre benötigen, das Erraten der Hexzahl dauert um ein Zigfaches länger.

Einige Hersteller versuchen, dies zu erschweren, indem sie die ID des Initiators verschleiern und als ID-Payload-Typ eine unstrukturierte Bitfolge (RFC2407, ID_KEY_ID) verwenden, in der die verschlüsselte ID enthalten ist. Leider sind der Verschlüsselung Grenzen gesetzt, da zu dem Zeitpunkt der Übertragung der ID noch gar keine Schlüssel ausgehandelt sind. Also behelfen sich viele mit Hash-Werten, gerne verwendet werden MD5 oder SHA1. Das ist im Prinzip auch in Ordnung, denn aus einem Hash-Wert kann man keinen Eingangswert zurückberechnen. Leider braucht man das auch nicht immer, denn in einigen Implementierungen kann man einfach raten und den Responder „fragen“, ob man eine gültige ID gefunden hat. Denn einige Implementierungen schicken unterschiedliche Antworten zurück, je nachdem, ob der Benutzername (Antwort: No_Proposal_Chosen) oder das Pre-Shared Secret (Antwort: Authentication_Failure) falsch ist.

Dieses Problem ist schon lange bekannt, es gibt sogar entsprechende Rate-Tools (z.B. ike-scan), und müsste eigentlich in allen aktuellen IKE-Implementierungen behoben sein. Leider ist dem nicht so.

Allerdings ist der Wind, den einige um Thema machen, ein bisschen überzogen, denn das Logfile des VPN-Systems, auf den ein IKE Scan losgelassen wird, quillt regelrecht über. Und Logfiles von Sicherheitssystemen gehören nun einmal regelmäßig und zeitnah ausgewertet.

XAUTH

Nun kommen wir zu einer richtig üblen Sache, dem unsäglichen XAUTH (Extended Authentication) Draft Standard. Das Ziel dieses Dokuments war ISAKMP und damit auch, IKE um weitere Authentifizierungsmechanismen zu ergänzen, die nach der IKE Phase 1 eingefügt werden können. Insbesondere wurde dabei an Radius, LDAP, TACACS, SecurID, MS-Domains und Ähnliches gedacht.

Leider ist dieser Draft, der mittlerweile von der IETF endgültig verworfen wurde, in verschiedenen Spielarten in fast jeder IPSec-Client-Implementierung zu finden. Im Prinzip ist XAUTH selbst eigentlich gar nicht so schlimm, nur hätten die Entwickler, Software-Designer und Produktmanager diesen Standard (draft-ietf-ipsec-isakmp-xauth-05.txt) vielleicht bis zu Ende lesen sollen. Denn dort findet sich folgendes Statement:

„The use of Extended Authentication does not imply that phase 1 authentication is no longer needed. Phase 1 authentication provides a higher level of user authentication by signing ISAKMP packets. Extended Authentication does not provide this service. The removal or weakening of phase 1 authentication would leave the IPSec session vulnerable to a man-in-the-middle attack and other spoofing attacks. Therefore, when using Extended Authentication with Pre-Shared keys, it is vital that the Pre-Shared key be well chosen and secure.“

Und genau diese Empfehlung wird in jeder mir bekannten IPSec-Client-Implementierung auf das Größte missachtet. Ja, es gibt sogar große Hersteller, die X-Auth als einzigstes Authentifizierungsverfahren außer digitalen Signaturen implementieren, mit praktisch keiner Sicherheit in der IKE Phase 1. Die meisten Anwender bekommen dies gar nicht mit, denn die meisten Clients verschleiern das Ganze unter Namen wie „Groups“, „Group Authentication“ oder Ähnlichem. Gemeinsam ist allen Spielarten, dass es eine Gruppe mit einer ID und einem Passwort gibt und innerhalb dieser Gruppe Benutzer mit UserID/Passwort, SecureID oder anderen benutzerspezifischen Attributen angelegt sind. Gruppen-ID und Gruppenpasswort sind für alle Mitglieder der Gruppe gleich und dienen als ID und Pre-Shared Key für die IKE-Phase-1-Aushandlung.

In sehr vielen realen Netzen gibt es für das gesamte Unternehmen eine einzige oder nur ganz wenige Gruppen-ID-/Passwort-Kombination, denn die sichere Verteilung von ID und PSK für jeden Benutzer, mit häufigem Wechsel, stellt ein ziemliches, wenn nicht sogar unlösbares administratives Problem dar. Also muss man das tun, wovor im XAUTH-Draft-Standard ausdrücklich gewarnt wird: Man verwendet eine einzige Gruppe für alle User und verlässt sich auf die Authentifizierung in XAUTH. Solch ein Pre-Shared Secret ist aber weder „well chosen“ noch wirklich „secret“ und macht damit eine Man-in-the-Middle-Attacke möglich.

XAUTH ist nur dann sicher, wenn jeder Benutzer eine eigene Gruppen-ID mit passendem PSK hätte oder man digitale Signaturen verwenden würde. Ersteres ist in der Regel nicht praktikabel, und im zweiten Fall braucht man XAUTH sowieso nicht.

Es ist dringend anzuraten, für Remote Access IKE mit digitale Signaturen einzusetzen. Erstens ist dieses Verfahren, insbesondere in Kombination mit SmartCards, extrem sicher, und zweitens kann man damit, trotz dynamischer IP-Adressen, den IKE Main Mode benutzen. Und wenn es irgendwie machbar ist: Hände weg von XAUTH!

Auch bei Site-to-Site VPN, also der Verbindung von Standorten über VPN, sollte man mit digitalen Zertifikaten und digitalen Signaturen arbeiten. Bei Standortverbindungen kann man allerdings auch noch ganz gut mit Pre-Shared Keys leben, denn erstens werden meistens feste IP-Adressen benutzt, und man kann somit auch mit PSK den IKE-Mainmode einsetzen. Die Pre-Shared Keys selbst können in einem Gateway hinreichend lang und komplex gewählt werden. An einer 30-stelligen Hexzahl mit ihren 2^{120} möglichen Werten beißen sich auch Programme wie PSK-Crack die Zähne aus.

Standortverbindungen, bei denen ein Standort dynamische IP-Adressen zugewiesen bekommt, also die typische DSL-Lösung, bedingt natürlich den IKE Aggressive Mode, allerdings kann man auch hier den Passwort-Rateprogrammen durch geeignete, lange HEX-Werte für den Pre-Shared Key einen Riegel vorschieben.

7 SSL-PN

Dem SSL-Protokoll wurde in der Vergangenheit oftmals nur eine geringe Bedeutung beigemessen, denn viele halten sich nicht vor Augen, dass SSL, fast sogar noch mehr als HTTP, das Protokoll ist, welches das Internet zu dem gemacht hat, was es heute ist. Denn ohne SSL gäbe es keine sicheren Webtransaktionen. Und damit kein eBay, kein Amazon, kein Home-Banking, keine Web-Shops und nicht all die anderen Webapplikationen, in denen Menschen ihre vertraulichen Daten übermitteln.

SSL hat mittlerweile sein auf HTTPS beschränktes Einsatzgebiet verlassen und wird auch in etlichen anderen Anwendungsbereichen eingesetzt, z.B. in der sicheren Netzwerkauthentifizierung und der Wireless-LAN-Sicherheit. Verfahren wie EAP-TLS, EAP-TTL oder PEAP benutzen SSL ebenso wie beispielsweise das LDAP-Protokoll, das ebenfalls SSL benutzen kann. Neu sind auch Einsatzgebiete zum sicheren Remote Access oder in Extranets – die SSL-VPN.

7.1 Geschichtliches

„The primary goal of the SSL Protocol is to provide privacy and reliability between two communicating applications.“ [Freier, Karton, Kocher; INTERNET-DRAFT SSL 3.0; Kapitel 1, Introduction]

„The primary goal of the TLS Protocol is to provide privacy and data integrity between two communicating applications.“ [Dierks, Allen; RFC 2246 The TLS Protocol Version 1.0; Kapitel 1, Introduction]

Hätten sich einige Leute wenigstens den ersten Satz des inhaltlichen Teils des SSL-Draft-RFC, des TLS-RFC und der IPsec Roadmap zu Gemüte geführt, dann würde es eine ganze Reihe von Missverständnissen, überflüssigen Diskussionen und vielleicht sogar den Begriff SSL-VPN gar nicht geben.

Denn vor wenigen Jahren begann die aggressive Vermarktung einer in dieser Spielart noch jungen Technologie, die von ihren Protagonisten als „SSL-VPN“ bezeichnet wurde. Alle diese Unternehmen hatten ausnahmslos SSL-VPN-Gateways als einziges Produkt in ihrem Portfolio und zielten mit ihrer Marketingstrategie darauf ab, IPsec durch ihre SSL-Plattformen abzulösen. Dass das Unsinn ist, haben natürlich Fachleute und Sicherheitsexperten sofort erkannt, denn die Ausrichtung der Protokolle ist nun wirklich so unterschiedlich, dass sie praktisch nicht miteinander vergleichbar und damit auch nicht austauschbar sind: IPsec arbeitet völlig infrastruktur- und applikationstransparent auf der Netzwerkebene, während SSL ebenso infrastrukturtransparent, aber völlig applikationsbezogen zwischen Transport- und Applikationsebene arbeitet. Damit unterstützt SSL selbst auch kein Tunneling, eine Hauptvoraussetzung von VPN.

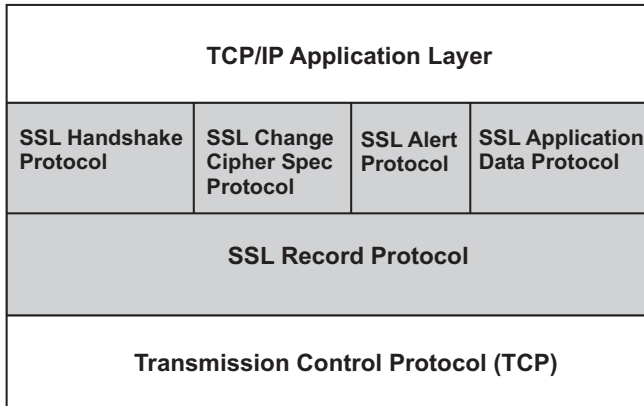


Abbildung 7.1: SSL legt sich zwischen TCP/IP-Applikationsebene und TCP.

Leider wurde die Diskussion „IPSec oder SSL“ in den letzten Jahren sehr polarisierend geführt, wodurch die wirklich wichtigen Dinge in den Hintergrund gedrängt wurden. All dies hat sehr viele potenzielle Anwender nach und nach ziemlich verunsichert und oft enttäuscht. Bei etlichen Kunden wird heute bei dem Thema SSL-VPN teilweise schon gleich abgewinkt – und das ist wirklich fatal, denn SSL bietet sich hervorragend als Ergänzung zu IPSec an und ist als Lösung für Extranets einfach unschlagbar!

Mittlerweile hat sich jedoch das Blatt gewendet, und die Fachpresse berichtet deutlich differenzierter über das Thema SSL-VPN. Auch die Analysten reden nicht mehr vom Ende von IPSec, sondern, wie einigen Vorträgen auf der SSL-Conference 2005 in Paris zu entnehmen war, davon, auf keinen Fall eine IPSec-Installation durch SSL abzulösen, sondern damit zu ergänzen. Das Thema ist auch nicht mehr „SSL oder IPSec“, sondern „SSL und IPSec“. Und mit dieser Lösung, der Kombination von SSL und IPSec, am besten noch in einem System, mit einer einzigen Benutzer- und Gruppenverwaltung, lassen sich wirklich sichere, flexible VPN aufbauen, die alle möglichen Funktionen bestmöglich abdecken. Es hat sich nämlich herausgestellt, dass sich die jeweiligen Schwächen der Einzelprotokolle in einer Gesamtlösung sehr gut ausgleichen lassen.

Bevor wir sehen, warum dies so ist, schauen wir uns die Grundlage dieser Technologie, das SSL- bzw. das TLS-Protokoll, zuerst einmal näher an, um deren Eignung für bestimmte Anwendungen besser verstehen zu können.

Secure Socket Layer (SSL)

SSL ist eine Entwicklung von Netscape mit dem Ziel, das HTTP-Protokoll durch eine optional aktivierbare Sicherheitskomponente zu erweitern, um sensible Geschäftsapplikationen über das Internet benutzen zu können. Hier wurden insbesondere sichere Transfers für Anwendungen wie E-Banking oder E-Commerce als Einsatzgebiet für SSL gesehen. So wurde denn SSL auch nicht etwa wie IPSec auf der Netzwerkebene implementiert, sondern als eigener Layer im oder genauer gesagt oberhalb des TCP-Protokolls. Das ist für die Zielsetzung von SSL auch der optimale Ansatz.

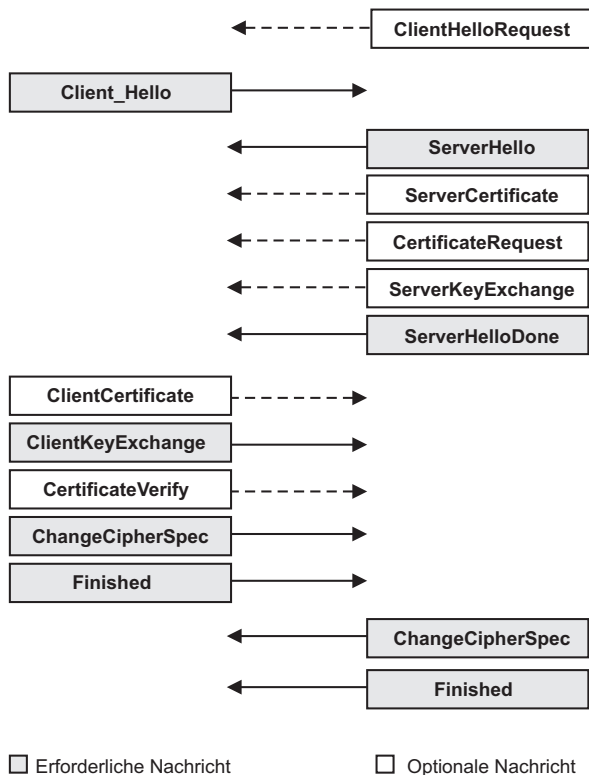


Abbildung 7.2: Der Aufbau einer SSL-Session

Die prinzipielle Funktion von SSL sieht vor, dass sich der Webserver gegenüber dem Browser des Benutzers stark authentifiziert. Hier baut man kompromisslos ausschließlich auf digitale Signaturen und Zertifikate hoher Sicherheitsstufen. Der Browser selbst authentifiziert sich nicht, denn eine Benutzerauthentifizierung findet bei SSL-Anwendungen in der Regel auf Applikationsebene statt, zum Beispiel per Kontonummer und PIN beim E-Banking oder auf anderen applikationsabhängigen Verfahren. Die Möglichkeit einer beidseitigen Authentifizierung per Signatur ist allerdings als mögliche Option in der SSL-Spezifikation verankert und wird vor allem in SSL-VPN genutzt.

Nach erfolgreicher Authentifizierung erfolgt eine sichere Kommunikation, innerhalb der alle HTTPS-Pakete verschlüsselt und gegen Veränderungen geschützt sind. Ebenso wie bei der Authentifizierung ist SSL hinsichtlich der einsetzbaren Verschlüsselungs- und Integritätsprüfungsalgorithmen sehr flexibel. Sowohl für PC optimierte Protokolle wie RC4 als auch Standards wie DES, Triple-DES oder AES sind neben anderen, nicht so verbreiteten Verfahren einsetzbar. Auch die leidigen Exportrestriktionen, die selbst beim Homebanking vor einigen Jahren außerhalb der USA nur eine SSL-Verschlüsselung von 40 Bit ermöglicht hatten, sind passé. Ärgerliche Sicherheitslöcher, wie z.B. die Erzeugung von Zufallszahlen aus der aktuellen Urzeit, sind Schnee von gestern, und eine korrekt programmierte SSL-Version-3-Implementierung erfüllt heute sehr hohe Sicherheitsstandards.

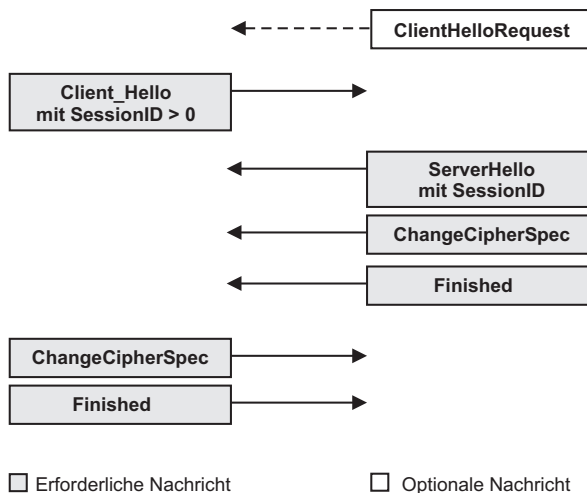


Abbildung 7.3: Der Aufbau einer SSL-Verbindung kann auch abgekürzt werden.

Der Ablauf von SSL sieht vor, dass zuerst eine Verständigung über die einzusetzenden Chiffrierverfahren und eine Serverauthentifizierung erfolgt. Anschließend erzeugt der Client einen Masterschlüssel, verschlüsselt diesen mit dem Public Key des Servers und schickt ihn zum Server. Der Server entschlüsselt das Ganze mit seinem privaten Schlüssel, und beide können sich jetzt ihre eigentlichen Session-Schlüssel erzeugen. Anschließend können die Nutzdaten sicher zwischen Client und Server übertragen werden. SSL ermöglicht es auch, eine vorgegangene Session wieder aufzunehmen oder weitere Verbindungen innerhalb einer Session zu erzeugen, ohne die aufwändigen Public-Key-Verfahren nochmals benutzen zu müssen.

Die Version 1 von SSL spielt heute überhaupt keine Rolle mehr, und auch die Version 2 wird mittlerweile als viel zu unsicher angesehen. Die Kinderkrankheiten und Sicherheitsmängel wurden seit der Version 3 des SSL-Protokolls endgültig überwunden.

Die Weiterentwicklung von SSL wurde mit der Version 3.0 beendet und ist in eine Arbeitsgruppe der Internet Engineering Task Force gemündet, die das Ganze unter dem Namen Transport Layer Security (TLS) weiter fortführt.

7.2.1 Transport Layer Security (TLS)

Die Internet Engineering Task Force hat sich schon seit einiger Zeit des SSL-Themas angenommen und basierend auf SSL Version 3.0 einen Standard verabschiedet. Dieser heißt nicht SSL, sondern TLS 1.0 (Transport Layer Security) und trägt die Nummer RFC2246. Die Unterschiede zu SSL sind marginal, aber ausreichend, damit TLSv1 und SSLv3 nicht miteinander kompatibel sind. Jedoch können praktisch alle TLS-Implementierungen auch in einem SSL-Kompatibilitätsmodus arbeiten.

Die weitere technische Beschreibung in diesem Kapitel beruht auf der SSL-Version 3.

Falls in irgendeinem Zusammenhang wissenswerte Unterschiede zwischen SSL Version 3 und TLS 1.0 existieren sollten, wird dies an der entsprechenden Stelle explizit erwähnt.

7.2.2 Kryptografie in SSL

In SSL werden viele Verschlüsselungs-, Schlüsselerzeugungs- und Hash-Verfahren eingesetzt, die auch im IPSec- und IKE-Protokoll Anwendung finden. So werden zum Beispiel DES, Triple-DES und, zumindest in neuen Implementierungen, AES zur Verschlüsselung der zu übertragenden Daten eingesetzt. Auch Algorithmen wie RC2 oder RC4 können eingesetzt werden. Insbesondere der Stromchiffrieralgorithmus RC4 zeichnet sich dabei durch hohe Performance und eine optimale Betriebsumgebung aus.

Die Erzeugung der Schlüssel erfolgt wahlweise über authentifizierte oder nicht authentifizierte (kein Spaß!) Diffie-Hellman-Varianten. Falls die Sitzungsschlüssel lokal im Client erzeugt werden, können sie mit RSA und dem Public Key des Servers verschlüsselt und anschließend übertragen werden. Die Integrität der Daten wird über HMAC-ähnliche Verfahren auf Basis von MD5 und SHA1 geschützt.

In SSL wird immer mit einer festen Kombination aus Verschlüsselungs-, Schlüsselerzeugungs- und Hash-Verfahren gearbeitet, die man dort als Cipher Suite bezeichnet. Im Gegensatz zu IPSec und IKE, die ihre Algorithmen völlig unabhängig voneinander festlegen können, sind hier immer bestimmte Verschlüsselungsverfahren mit bestimmten Schlüsselerzeugungsverfahren verkoppelt. Allerdings ist eine ganze Reihe verschiedener Kombinationen spezifiziert, welche die sinnvollen und sicheren Kombinationen weitgehend abdecken. Wer TLSv1 mit SSLv3 vergleicht, wird feststellen, dass Fortezza in TLS nicht mehr auftaucht. Das hat verschiedene Gründe, vor allem aber die Ablehnung von Fortezza durch viele namhafte Wissenschaftler im Bereich der Kryptologie. Das nicht etwa, weil Fortezza unsicher ist, sondern weil es nicht offen gelegt ist und keiner so ganz genau weiß, wie es funktioniert – oder was es für Hintertüren bietet. Nicht offen gelegte Algorithmen (man kann durchaus auch patentierte Algorithmen offen legen) können nämlich nicht von der Gemeinde der Kryptologen analysiert und getestet werden, sondern man kann sich nur auf das Wort des Herstellers verlassen.

Auch die Tatsache, dass alles, sowohl die Schlüsselerzeugungs- als auch die Verschlüsselungsalgorithmen, geheim gehalten werden, lässt bei vielen die Alarmglocke in den höchsten Tönen schrillen. Zu frisch sind die Erinnerungen an die Hintertüren, die Microsoft oder Lotus für die NSA eingebaut hatten, oder an der Clipper-Chip der NSA. So etwas braucht niemand, vor allem nicht, wenn es robuste, bewährte und seit Jahren vergeblich zu knacken versuchte Algorithmen wie Diffie-Hellman, RSA, Triple-DES oder AES gibt – die man obendrein auch noch frei von irgendwelchen Lizenzansprüchen einsetzen kann.

In Tabelle 7.1 sind die Cipher Suites von SSL Version 3 aufgelistet, eigentlich sollte für jedes Sicherheitsbedürfnis etwas dabei sein. Allerdings ist Vorsicht geboten, denn einige Kombinationen gilt es unter allen Umständen zu vermeiden. Das man von der Cipher Suite `SSL_NULL_WITH_NULL_NULL` am besten die Finger lässt, suggeriert schon deren Name. Eine Übertragung ohne Verschlüsselung, Authentifizierung und Integritätsprüfung ist bestenfalls im Labor zum Testen bestimmter Funktionen zu gebrauchen. Aber auch andere Cipher Suites, die mit zu kurzen Schlüsseln arbeiten, wie zum Beispiel RC4 oder DES mit 40 oder 56 Bit, gilt es zu vermeiden, ebenso wie RSA mit einem nur 512 Bit großen Schlüssel. Praktisch alles, was den String `EXPORT` im Namen der Cipher Suite führt, ist heutzutage untauglich für eine sichere Übertragung. Diese Cipher Suites stammen noch aus der Zeit, als die USA allen Ernstes glaubten, die Kryptografie weltweit entsprechend den Wünschen ihrer National Security Agency regulieren zu können.

Jemand, der SSL-VPN einsetzt, hat glücklicherweise die volle Kontrolle über das, was Client und Server an Cipher Suites aushandeln können und sollte einen entsprechend hohen Sicherheitslevel auswählen.

7.2.3 Die Schlüsselerzeugung in SSL

Alle Schlüssel in SSL werden von einem so genannten Pre-Master Secret abgeleitet. Dieses kann mit drei verschiedenen Verfahren erzeugt werden:

1. RSA
2. Diffie-Hellman
3. Fortezza in SSLv3 oder DSS (Digital Signature Standard) in TLS

In der Praxis findet man allerdings meist nur die ersten beiden Varianten, die auch hier etwas näher beleuchtet werden.

Cipher_Suite	Key Exchange	Cipher	Hash
SSL_NULL_WITH_NULL_NULL	–	–	–
SSL_RSA_WITH_NULL_MD5	RSA	–	MD5
SSL_RSA_WITH_NULL_SHA	RSA	–	SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5	RSA-512	RC4-40	MD5
SSL_RSA_WITH_RC4_128_MD5	RSA	RC4-128	MD5
SSL_RSA_WITH_RC4_128_SHA	RSA	RC4-128	SHA
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	RSA-512	RC2 40 Bit	MD5
SSL_RSA_WITH_IDEA_CBC_SHA	RSA	IDEA	SHA
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	RSA-512	DES 40 Bit	SHA
SSL_RSA_WITH_DES_CBC_SHA	RSA	DES 56 Bit	SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA	RSA	Triple-DES	SHA
SSL_DH_DSS_EXPORT_WITH_DES40_CBC_SHA	D/H-512, DSS	DES 40 Bit	SHA
SSL_DH_DSS_WITH_DES_CBC_SHA	D/H, DSS	DES 56 Bit	SHA
SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA	D/H, DSS	Triple-DES	SHA
SSL_DH_RSA_EXPORT_WITH_DES40_CBC_SHA	D/H-512, RSA	DES 40 Bit	SHA
SSL_DH_RSA_WITH_DES_CBC_SHA	D/H, RSA	DES 56 Bit	SHA
SSL_DH_RSA_WITH_3DES_EDE_CBC_SHA	D/H, RSA	Triple-DES	SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	Ephemeral D/H-512, DSS	DES 40 Bit	SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA	Ephemeral D/H, DSS	DES 56 Bit	SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	Ephemeral D/H, DSS	Triple-DES	SHA

SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	Ephemeral D/H 512 Bit, RSA	DES 40 Bit	SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA	Ephemeral D/H, RSA	DES 56 Bit	SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	Ephemeral D/H, RSA	Triple-DES	SHA
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	Anonymous D/H, no signature	RC4-40	MD5
SSL_DH_anon_WITH_RC4_128_MD5	Anonymous D/H, no sig.	RC4-128	MD5
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA	Anonymous D/H 512 Bit , no sig.	DES 40 Bit	SHA
SSL_DH_anon_WITH_DES_CBC_SHA	Anonymous D/H, no sig.	DES 56 Bit	SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	Anonymous D/H, no sig.	Triple-DES	SHA
SSL_FORTEZZA_KEA_WITH_NULL_SHA	Fortezza	Fortezza	SHA
SSL_FORTEZZA_KEA_WITH_FORTEZZA_CBC_SHA	Fortezza	Fortezza	SHA
SSL_FORTEZZA_KEA_WITH_RC4_128_SHA	Fortezza	Fortezza	SHA

Tabelle 7.1: Die Standard-Cipher-Suites von SSL Version 3

In diesem für HTTP typischen Szenario, in dem ein Browser eine SSL-Session startet und sich auf SSL-Ebene nur der Server authentifiziert, wird das Pre-Master Secret vom Client auf Zufallsbasis erzeugt und verschlüsselt zum Server übertragen. Hierzu wird der Public Key des Servers benutzt, den dieser in seinem Zertifikat, das in der Server-Hello-Message enthalten war, dem Client zur Verfügung gestellt hat. Der Client muss das Zertifikat natürlich vorher auf Echtheit und Gültigkeit prüfen. Aber Vorsicht, zu diesem Zeitpunkt ist noch keine Authentifizierung erfolgt, weder die des Servers noch des Clients noch des Schlüsseltausches selbst!

Hier wird das Standard Diffie-Hellman-Verfahren (D/H) benutzt. Im Gegensatz zu IPsec sind die öffentlichen Werte (Modulus und Generator) nicht im Standard festgelegt, sondern werden vom Server zur Verfügung gestellt. Dies geschieht, indem die Werte in einem signierten Zertifikat übertragen werden, wodurch deren Authentizität sichergestellt ist. Falls diese Werte nicht in der Server-Hello-Message übertragen werden, kann dies optional in einer folgenden Server-Key-Exchange-Message nachgeholt werden.

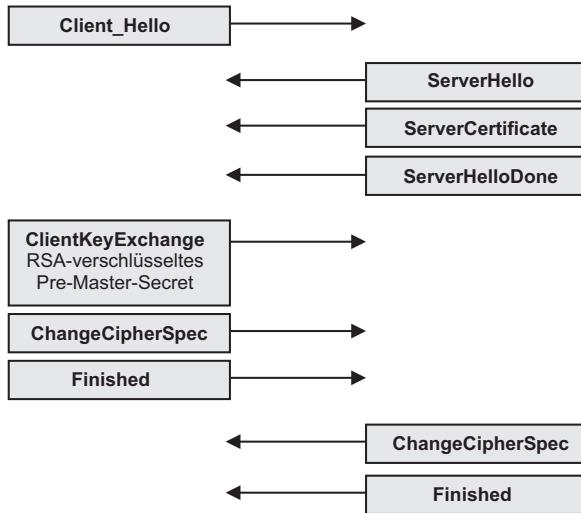


Abbildung 7.4: Der Aufbau einer SSL-Session mit RSA-Schlüsseltausch

An dieser Stelle sollte vielleicht erwähnt werden, dass sich ein Client und ein Server mit jeweils festen (Zertifikat) Diffie-Hellman-Parametern immer das gleiche Pre-Master Secret erzeugen. Dies liegt daran, dass alle drei Parameter in signierten Zertifikaten gespeichert und damit nicht veränderbar sind. Und wenn sowohl Client als auch Server entsprechende Zertifikate benutzen, dann ist bei immer gleichem Generator, Public-Value und gleicher Primzahl auch das Master Secret immer das gleiche.

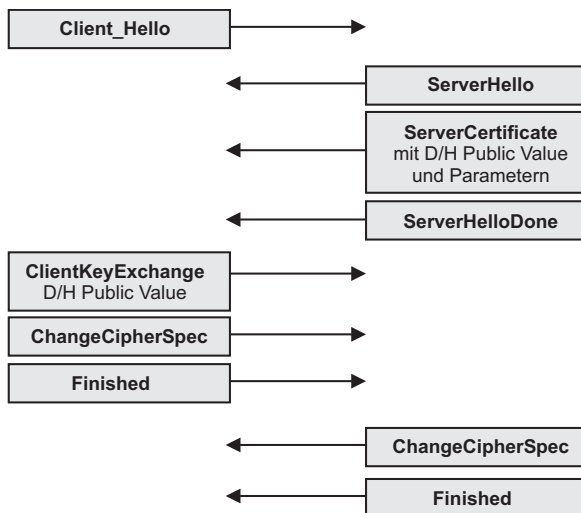


Abbildung 7.5: Ein Verbindungsaufbau mit Diffie-Hellman-Schlüsselerzeugung

Dies alles gilt nicht für Ephemeral Diffie-Hellman. In dieser Variante werden die Diffie-Hellman-Parameter weder vom Server noch vom Client in Zertifikaten übertragen, sondern in eigenen Key-Exchange-Nachrichten übermittelt. Der Vorteil dieses Verfahrens ist, dass man nur ein normales Signaturzertifikat für den Server benötigt. Hier werden auch immer neue Master Secrets erzeugt.

Nun sind sowohl Client als auch Server im Besitz des gleichen Pre-Master Secrets. Die Qualität des Pre-Master Secrets hängt entscheidend von der Güte der Zufallszahl ab, die sich der Client (RSA) oder Client und Server (D/H) erzeugt haben. Denn das Pre-Master Secret alleine kann in SSL nicht als Ausgangsbasis zur Erzeugung der Sitzungsschlüssel dienen. Es kann nämlich durchaus vorkommen (zum Beispiel dann, wenn sowohl Client als auch Server ein gültiges Zertifikat mit ihren öffentlichen Diffie-Hellman-Parametern benutzen), dass sich ein Client mit einem bestimmten Server immer das gleiche Pre-Master Secret berechnet! Um dies zu verhindern, wird in jeder Sitzung aus dem Pre-Master Secret und einigen Zufallszahlen (ServerHello.Random, ClientHello.Random) über Eingewegfunktionen das Master Secret berechnet.

Dies geschieht über folgendes HMAC-Konstrukt (Hash-based Message Authentication Code, vgl. Kap 4):

Master-Secret = MD5(Pre-Master-Secret + SHA(,A' + Pre-Master-Secret + ClientHello.Random+ServerHello.Random)) + MD5(Pre-Master-Secret+SHA(,BB'+ Pre-Master-Secret+ClientHello.Random+ServerHello.Random)) + MD5(Pre-Master-Secret+SHA(,CCC'+Pre-Master-Secret+ClientHello.Random+ServerHello.Random))

Diese Berechnungsweise ist extrem sicher. Die Verwendung zweier unterschiedlicher Hash-Algorithmen erlaubt, dass durchaus Schwächen in einem Algorithmus existieren können, ohne dass dadurch die Schlüsselerzeugung kompromittiert würde. Und ein HMAC-Konstrukt ist dann noch kollisionsresistent, wenn es Kollisionen für den verwendeten Hash-Algorithmus gibt. Im konkreten Fall von MD5 und SHA – hier wurden für beide Verfahren Kollisionen gefunden – bedeutet dies, dass das ganze Konstrukt trotzdem nach wie vor resistent gegen Kollisionen ist! Also besteht hier für SSL vorerst keine Gefahr, allerdings sollten im Lauf der Zeit neue oder zumindest zusätzliche Hash-Algorithmen eingesetzt werden.

Aus dem Master Secret werden nun die Session-Schlüssel erzeugt. Für Client und Server benötigt man jeweils einen MAC-Schlüssel, einen Schlüssel zur Datenverschlüsselung und einen Initialisierungsvektor für CBC. Diese Werte werden einem ausreichend langen Block von Schlüsselmaterial entnommen, der mit folgendem Verfahren erzeugt wird:

Key-Block=MD5(Master-Secret+SHA(,A'+Master-Secret+ClientHello.Random+ ServerHello.Random) + MD5(Master-Secret + SHA(,BB' + Master-Secret + ClientHello.Random+ServerHello.Random))+MD5(Master-Secret+SHA(,CCC'+ Master-Secret + ClientHello.Random + ServerHello.Random) + ...

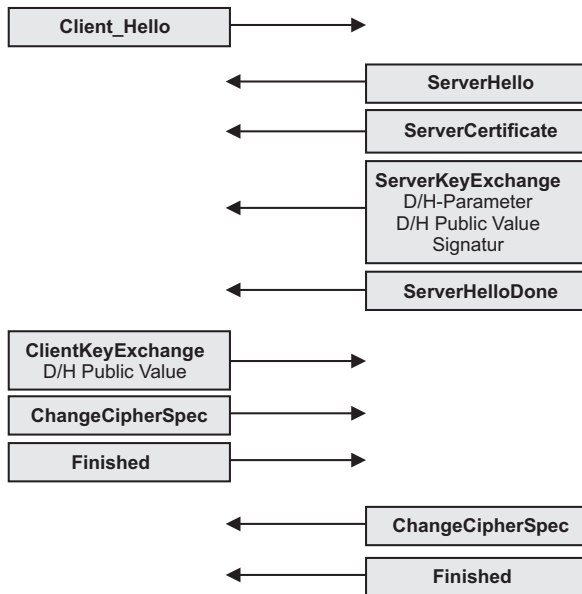


Abbildung 7.6: Ephemeral Diffie-Hellman benutzt keine Zertifikate zur Übertragung der notwendigen Parameter.

Die Verkettung wird so lange durchgeführt, bis genug Werte für die ausgewählten Algorithmen zur Verfügung stehen. Aus dem Schlüsselblock werden anschließend die einzelnen Schlüssel extrahiert. Nehmen wir an, wir benötigen 128 Bit für die MAC-Schlüssel, 128 Bit für den RC4-128-Algorithmus und 64 Bit für den Initialisierungsvektor. Dann sähe das Ganze auf Bit-Ebene so aus:

Client_MAC = Key_Block [0, 127]
 Server_MAC = Key_Block [128, 255]
 Client_Key = Key_Block [256, 383]
 Server_Key = Key_Block [384, 511]
 Client_IV = Key_Block [512, 575]
 Server_IV = Key_Block [576, 639]

Der Key-Block müsste also mindestens 640 Bit lang sein.

Die Erzeugung von Exportschlüsseln (40-Bit-Verschlüsselung bzw. 512-Bit-Verschlüsselung für die Schlüsselerzeugung) ist heute kein Thema mehr, insofern können wir uns deren etwas abweichende Erzeugung hier sparen.

In SSL findet eine ganze Reihe von Algorithmen Verwendung, von denen man den für die jeweilige Plattform günstigsten auswählen kann. So sind zum Beispiel RC4 oder AES sehr schnell in Software, während Triple-DES erst in entsprechenden Beschleunigerchips an Tempo gewinnt. Man sollte in jedem Fall versuchen, AES Cipher Suites zwischen

Client und Server aushandeln zu lassen, da diese sowohl für Soft- als auch Hardware-Implementierungen sehr schnell sind. Allerdings ist AES längst nicht in jedem System implementiert, insbesondere einige Serverplattformen sind da nicht ganz auf dem aktuellen Stand. SSL und TLS unterstützen vom ursprünglichen Standard her folgende Verfahren:

- RC2-40 (40 Bit Blockchiffre)
- RC4-40 (40 Bit Stromchiffre)
- RC4-128 (128 Bit Stromchiffre)
- DES-40 (40 Bit Blockchiffre)
- DES (56 Bit Blockchiffre)
- Triple-DES (168 Bit Blockchiffre)
- IDEA (128 Bit Blockchiffre)
- Fortezza (nur in SSL, 96 Bit Blockchiffre)

RC4 ist das einzige Stromchiffrierverfahren in dieser Liste. Alle Blockchiffrierverfahren außer Fortezza arbeiten mit CBC (Cipher Block Chaining), mit 64 Bit Blocklänge und mit einem 64 Bit großen expliziten Initialisierungsvektor. Fortezza arbeitet auch mit 64-Bit-Blöcken, aber mit einem 160-Bit-Initialisierungsvektor.

Die Initialisierungsvektoren (IV) werden durch das SSL Handshake Protocol in für den ausgewählten Blockchiffrieralgorithmus ausreichender Länge erzeugt. Für den ersten Record wird dieser IV benutzt, für folgende Records wird der letzte Chiffretextblock des vorhergehenden Records benutzt. Im Gegensatz zu IPSec-ESP werden die Initialisierungsvektoren nicht übertragen. Dies ist durch die Verwendung von TCP nicht notwendig, da die SSL-Records in jedem Fall in der korrekten Reihenfolge ankommen und beide Seiten daher, ausgehend vom (geheimen) ersten IV, ihre Vektoren synchron mitführen können.

Diese Funktion wird über MAC (Message Authentication Code) realisiert. Durch die Verwendung eines symmetrischen Schlüssels zusammen mit den einschlägigen Hash-Verfahren zur Erzeugung des MAC wird gleichzeitig sowohl die Authentifizierung als auch die Integritätssicherung des Records durchgeführt.

Als Hash-Verfahren werden MD5 und der SHA benutzt. Obwohl für die Algorithmen als solche bereits Kollisionen gefunden wurden, betrifft dies SSL (und auch IPSec) überhaupt nicht. Und dies gleich aus zwei Gründen:

1. Die Kollisionen wurden ermittelt, indem man beide Eingangsgrößen der jeweiligen Hash-Funktion so lange variiert hat, bis ein gleicher Hash-Wert erzeugt wurde. Bei SSL kann allerdings nur die Fälschung variiert werden, da der SSLCompressed Record natürlich feststeht. Das hat bisher noch niemand geschafft.
2. In SSL (und IPSec) werden die Hash-Algorithmen in der HMAC-Betriebsart benutzt, die gegen Kollisionen resistent ist, selbst wenn dies für die Hash-Verfahren selbst nicht gilt.

Somit ist für jeden SSL-Record ein wirksamer Schutz seiner Integrität gegeben.

Auch SSL schützt sich gegen Angriffe durch wiederholtes Senden von Records (Replay Attacks) und benutzt ebenso wie IPSec eine Sequenznummer zu diesem Zweck. Durch die Verwendung von TCP, dass die Zustellung der Records in korrekter Reihenfolge garantiert, kann SSL es sich allerdings viel einfacher als IPSec machen. Denn man benötigt keinen aufwändigen Window-Mechanismus, da die Nummern immer in der korrekten Reihenfolge ankommen. „Ankommen“ ist hier allerdings im übertragenen Sinn zu verstehen, denn die Sequenznummer selbst wird gar nicht mitgesendet, sondern fließt implizit in den MAC-Algorithmus mit ein. Der Zähler wird einfach bei jedem neuen SSL-Record um eins erhöht. Initialisiert (auf den Wert null) wird der Sequenznummernzähler durch eine Change-Cipher-Spec-Nachricht (s.u.).

Zusammenfassend lässt sich sagen, dass SSL aus kryptografischer Sicht ein sehr sicheres Protokoll ist – sofern man die richtigen Cipher Suite auswählt. Die Diskussion, ob SSL oder IPSec vom Protokoll her sicherer ist, ist ziemlich unsinnig. Beide Protokolle bieten bei korrekter Implementierung und Konfiguration ein sehr hohes Maß an Sicherheit.

Die Flexibilität der Schlüsselerzeugung ist höher als die von IKE, wodurch sich SSL in verschiedensten Umgebungen einsetzen lässt. Auch hat man erfreulicherweise, im Gegensatz zu IKE, auf die Verwendung von Pre-Shared Keys als einen der Parameter zur Schlüsselerzeugung gänzlich verzichtet, wodurch ein möglicher Angriffspunkt vermieden wird.

Die Verschlüsselungsalgorithmen und möglichen Schlüssellängen spiegeln den aktuellen Stand der Technik wider und sind sowohl für Soft- als auch für Hardware-Implementierungen geeignet. Aus Sicherheitssicht sind die Schlüssellängen auch für die nächsten Jahre mehr als ausreichend, neue Algorithmen können jederzeit in ergänzende Standards aufgenommen werden, ohne den SSL- oder TLS-Protokollstandard als solchen modifizieren zu müssen.

Die Verwendung von SHA1 und MD5 als Hash-Funktionen in SSL birgt zurzeit keinerlei Sicherheitsrisiko, da beide nicht generisch, sondern sowohl in beidseitiger Kombination als auch in einer speziellen, sicheren Betriebsart (HMAC) betrieben werden. HMAC-MD5 und HMAC-SHA sind beide kollisionsresistent, obwohl sowohl für MD5 als auch für SHA1 bereits Kollisionen gefunden wurden.

Die Authentifizierung sollte man vorerst einmal auf RSA-Signaturen beschränken, da der Digitale Signature Standard (DSS) auf SHA1 in seiner ursprünglichen Betriebsart (kein HMAC) basiert und damit anfällig gegen Kollisionsangriffe sein könnte. Dies ist zum gegenwärtigen Zeitpunkt zwar noch nicht abschließend beantwortet, aber sicher ist sicher, und es spricht wirklich nichts gegen RSA mit hinreichend langen Schlüsseln.

SSL-Funktionsblöcke

SSL besteht aus einigen voneinander abgegrenzten Funktionsblöcken, welche die komplette Funktionalität abdecken. Diese nennen sich:

- SSL Handshake Protocol
- SSL Change Cipher Spec Protocol
- SSL Alert Protocol
- SSL Record Protocol

Das SSL Record Protocol ist immer aktiv, wenn SSL gestartet wurde, denn die drei anderen Protokolle benutzen es als Vehikel für ihre Übertragung.

Client und Server unterhalten Datenstrukturen, die, vergleichbar mit den Security Associations von IPSec, den aktuellen Zustand von Sessions (SSL Session State) und Verbindungen (SSL Connection State) reflektieren. Eine Session kann verschiedene Connections unterhalten, ebenso kann ein Client oder Server auch mehrere Sessions betreiben. Der Aufbau der States ist wie folgt:

Session State

Der Session State enthält im Wesentlichen die Parameter zur Identifikation von Client und Server und die vom Handshake-Protokoll ausgehandelten Parameter zur Erzeugung von einer oder mehreren SSL Connections:

Session Identifier: Ein zufälliger Wert, der eine aktive oder wiederaufnehmbare Session eindeutig identifiziert.

Peer Certificate: Ein Server- oder Clientzertifikat im X.509v3-Format. Das Feld kann auch leer sein.

Compression Method: Der zur Datenkomprimierung verwendete Algorithmus.

Cipher Spec: Die Kombination aus Schlüsselerzeugungs-, Verschlüsselungs- und MAC-Algorithmus.

Master Secret: Ein 48-Byte-Wert, der nur Client und Server bekannt ist und aus dem die verschiedenen Schlüssel erzeugt werden.

Is_resumable: Dieses Flag kennzeichnet, ob eine Session wieder aufnehmbar ist.

Zu einer SSL Session existiert mindestens eine SSL Connection. Im SSL Connection State sind folgende Werte enthalten:

Server Random: Vom Server erzeugte Zufallszahl, die für alle Connection States unterschiedlich sein muss.

Client Random: Vom Client erzeugte Zufallszahl, die für alle Connection States unterschiedlich sein muss.

Server Write MAC Secret: Das Shared Secret zur Verwendung im MAC-Algorithmus im Server.

Client Write MAC Secret: Das Shared Secret zur Verwendung im MAC-Algorithmus im Client.

Server Write Key: Der Schlüssel, der vom Server zur Verwendung mit den ausgehandelten Verschlüsselungsalgorithmen benutzt wird.

Client Write Key: Der Schlüssel, der vom Client zur Verwendung mit den ausgehandelten Verschlüsselungsalgorithmen benutzt wird.

Initialisation Vector: Der IV wird von Blockchiffrierverfahren im CBC-Modus benötigt. Er wird durch das SSL Handshake Protocol erzeugt und anschließend durch letzten Chiffretextblock jedes SSS Ciphertext Records ersetzt.

Sequence Number: Die 64 Bit lange Sequenznummer wird von einer Change Cipher Spec Message auf null gesetzt und durch jeden eingehenden SSL-Record erhöht.

Das SSL Handshake Protocol leitet eine SSL Session ein oder nimmt eine vorhergegangene wieder auf. Verglichen mit den anderen SSL-Komponenten ist dieses Protokoll relativ komplex. Es ist von Aufbau und Funktion her mit dem IKE-Protokoll vergleichbar. Im SSL Handshake Protocol können verschiedene Nachrichtentypen versandt werden, die je nach Funktion verschiedene Parameter enthalten können.

Das Format der Nachrichten ist recht einfach, es beginnt mit einem Byte, das den Nachrichtentyp festlegt, 3 Bytes, welche die Länge der Nachricht festlegen, und der Nachricht selbst. Folgende Nachrichten sind spezifiziert:

Hello_Request (Typ = 0) Diese Nachricht mit der Typnummer 0 hat keine weiteren Parameter. Der Hello_Request ist eine Aufforderung des Servers an den Client, eine SSL-Verbindung für eine bereits existierende, unsichere Verbindung, meist HTTP, aufzubauen. Aufgrund der Tatsache, dass eine SSL-Session nur vom Client aufgebaut werden kann, ist diese Nachricht dann notwendig, wenn der Server eine existierende unsichere Verbindung sicher machen möchte. Ein Beispiel ist zum Beispiel das Homebanking: Man verbindet sich mit normalem HTTP auf der Bankenwebsite, und sobald man das Login-Fenster auswählt, schickt der Server einen Hello_Request zum Browser, der dann seine SSL-Funktionalität aktiviert.

Client_Hello (Typ = 1) Mit dieser Nachricht beginnt der Client den Aufbau einer SSL-Session. Sie wird entweder durch eine Benutzeraktion (z.B. Eingabe `https://...`) ausgelöst oder erfolgt als Reaktion auf eine empfangene Hello_Request-Nachricht. Sie besteht aus folgenden Parametern:

Client_Version gibt die höchste SSL-Version an, mit welcher der Client arbeiten will.

Random ist eine Datenstruktur, bestehend aus der aktuellen Zeit im Standard-32-Bit-Unix-Format und einem 28 Byte langen Pseudozufallswert, der von einem Zufalls-generator (PRNG, Pseudo Random Number Generator) erzeugt wurde. Bei SSL ist kein explizites Verfahren spezifiziert, bei TLS hingegen ist exakt festgelegt, wie (und damit auch wie gut oder schlecht) die PRN (Pseudo Random Number) erzeugt wird.

Session_ID ist ein Feld, das entweder „0“ ist und damit vom Server die Erzeugung einer vollständig neuen Session fordert oder in dem der Wert der aktuellen, einer vorhergehenden oder einer anderen aktiven Session eingetragen ist. Die drei letzten Optionen sollen vermeiden, dass jeweils ein komplett neuer Handshake durchge-

führt wird. Ist der Wert gleich der aktuellen Session_ID, können auf diese Weise verschiedene Parameter erneuert werden. Handelt es sich um die ID einer beendeten Session und der Server hat deren Zustand noch im Cache, kann auf diese Weise schnell die Session wieder aufgenommen werden.

Cipher_Suites ist eine Liste, in der mit absteigender Priorität die vom Client gewünschten Kombinationen der Kryptografieverfahren aufgelistet sind. Der Server wählt davon eine aus. Falls nicht, schickt er dem Client einen Handshake_Failure-Alarm und beendet die Verbindung. Die unterstützten Algorithmen findet man in den aktuellen Spezifikationen zu TLS oder SSL. TLS unterstützt bis auf Fortezza die gleichen Cipher_Suite-Kombinationen wie SSLv3. Der Wegfall von Fortezza macht aber niemanden richtig unglücklich, denn es handelt sich dabei um ein nicht offen gelegtes Kryptosystem, mit geheimen Algorithmen zur Ver- und Entschlüsselung, Schlüssel- und Initialisierungsvektorerzeugung.

So etwas ist für alle Kryptologen ein rotes Tuch und hat folgerichtig im RFC2246 keinen Einzug gefunden. Die Cipher_Suites mit „EXPORT“ im Namen sind die Verfahren, die nicht der Exportkontrolle des US-Außenhandelsministeriums oder des deutschen Kriegswaffenkontrollgesetzes unterliegen. Allerdings sind diesen Verfahren nur noch einige wenige, einschlägig bekannte „Schurkenstaaten“, die den Terrorismus unterstützen, betroffen.

Für TLS gibt es seit einiger Zeit zusätzliche Cipher_Suites, die Algorithmen wie AES (RFC3268) oder Verfahren wie Kerberos (RFC2712) für die Benutzung in TLS spezifizieren.

Compression_Methods ist eine nach absteigender Priorität sortierte Liste von Kompressionsverfahren, die der Client benutzen möchte. Wenn das Session-ID-Feld nicht 0 ist, dann muss mindestens das Verfahren der betreffenden Session in der Liste enthalten sein.

Nachdem der Client seine Client_Hello-Nachricht verschickt hat, wartet er auf die Antwort des Servers durch dessen Server_Hello. Kommt stattdessen irgendein anderes SSL-Handshake-Paket an, wird dies als schwerwiegender Fehler interpretiert.

Aufgrund der weitgehend eindeutigen Spezifikation des SSL- bzw. TLS-Protokolls auch hinsichtlich der Algorithmen, die zu implementieren sind, ist der Inhalt des Client_Hello-Pakets unter normalen Bedingungen für den Server akzeptabel. Er beantwortet daraufhin das Client_Hello- mit einem Server_Hello-Paket.

Server_Hello (Typ = 2) Der Server beantwortet ein Client_Hello mit seinem Server_Hello, wenn er den Vorschlag, den der Client hinsichtlich der Sicherheits- und Kompressionsverfahren gemacht hat, akzeptieren kann. Im anderen Fall schickt er einen Handshake_Failure_Alert zum Client und beendet die Session.

Die Server_Hello-Nachricht enthält die gleichen Felder wie das Client_Hello, jedoch mit anderem Inhalt, denn der Server wählt jeweils nur eine einzige Option aus den Vorschlägen des Clients aus:

Server_Version enthält die höchste SSL-Version des Servers, die auch gleichzeitig vom Client unterstützt werden kann.

Random enthält einen Zufallswert, der sowohl ungleich als auch unabhängig von dem des Client sein muss.

Session_ID ist die ID der aktuell aufzubauenden Session. Wenn der Client den Wert 0 geschickt hat, erzeugt der Server eine neue Session ID und fährt mit dem Handshake-Protokoll fort. Hat der Client eine ID ungleich 0 geschickt, prüft der Server, ob dieser Wert noch in seinem Cache ist. Wenn nicht, soll eine neue Session mit den Parametern der aktuellen Session aufgebaut werden, und der Server erzeugt eine neue ID, allerdings muss der SSL-Handshake nicht mehr komplett durchlaufen werden. Falls die ID im Cache ist und falls die Server-Policy eine Session-Wiederaufnahme überhaupt erlaubt, wird die alte Session wieder aufgenommen, und der Server antwortet mit der gleichen ID, die der Client geschickt hat. Beide überspringen dann alle weiteren Handshake-Schritte bis zur Change_Cipher_Spec- und Finish-Nachricht.

Falls der Server keine Session wiederaufnehmen will, schickt der Server eine ID gleich 0, und der Client weiß damit, dass er erneut den SSL-Handshake beginnen muss – nur diesmal mit 0 als Session-ID-Wert.

Die **Cipher Suite** ist die vom Server aus der Liste des Clients ausgewählte Kombination von Schlüsselerzeugung, Authentifizierung, Verschlüsselung und Hash-Verfahren. Es steht nur ein Wert in diesem Feld. Für wieder aufgenommene Sessions muss der Wert aus dem alten Session-State des Servers verwendet werden – unabhängig vom Inhalt von Cipher_Suite des Client_Hello.

Compression Method ist die vom Server aus der Liste des Clients ausgewählte Methode zur Datenkomprimierung. Es steht nur ein Wert in diesem Feld. Für wieder aufgenommene Sessions muss der Wert aus dem alten Session-State des Servers verwendet werden – unabhängig vom Inhalt des Compression_Method-Feldes Cipher_Suite des Client_Hello.

Certificate (Typ = 11) Diese optionale Nachricht enthält als einzigen Parameter das digitale Zertifikat des Servers, falls sich der Server authentifizieren muss, was praktisch in jeder sicheren Anwendung der Fall ist. Es handelt sich dabei um ein normales X.509v3-Zertifikat. In diesem Zertifikat ist auch der Public Key des Servers enthalten, mit dem der Client sein Pre-Master Secret verschlüsseln kann. Falls Diffie-Hellman als Schlüsselerzeugungsverfahren ausgewählt wurde, müssen die öffentlichen D/H-Parameter im Zertifikat hinterlegt sein.

Der Client benutzt ebenfalls diesen Nachrichtentyp, um sein eigenes Zertifikat zum Server zu übertragen, wenn dieser es angefordert hat. Falls es sich dabei um ein Zertifikat mit Diffie-Hellman-Parametern handelt, müssen diese mit den Werten (Modulus und Generator) des Serverzertifikats übereinstimmen.

Server_Key_Exchange (Typ = 12) Diese optionale Message wird vom Server verschickt, wenn er kein eigenes Zertifikat besitzt, sein Zertifikat nur zur Authentifizierung benutzt werden kann oder darf oder wenn die Fortezza-Cipher-Suite benutzt wird. Die Struktur der Nachricht hängt natürlich vom ausgewählten Schlüsselerzeugungsverfahren ab. Im Fall von RSA werden folgende drei Parameter übertragen:

Der temporäre **RSA_Modulus**, also das Produkt aus den beiden geheimen Primzahlen des RSA-Algorithmus

Der **RSA_Exponent**

Signatur der Parameter

Im Fall von Diffie-Hellman werden vier Parameter zum Client gesendet:

Der Wert p ist die **große Primzahl (Modulus)**.

Der Parameter g ist der **Generator**.

Y_s ist der **öffentliche Diffie-Hellman-Wert**, gebildet aus der geheimen Zahl des Servers (s), p und g nach der Formel $Y_s = g^s \bmod p$.

Signatur der Parameter.

Um Man-in-the-Middle-Angriffe zu verhindern, werden die Werte digital signiert. Dazu wird ein Hash (MD5 oder SHA1) über eine Verkettung aus den Serverparametern, ClientHello.Random und ServerHello.Random gebildet, die mit RSA oder DSA signiert wird.

Certificate_Request (Typ = 13) Ein nichtanonymer Server kann optional vom Client ein Zertifikat und eine Authentifizierung per digitaler Signatur anfordern. Er übergibt dem Client dabei folgende zwei Parameter:

Die Liste **Client_Certificate_Type** besteht aus einer Reihe von Zertifikatstypen, die aufsteigend nach der Präferenz des Servers sortiert ist.

Distinguished_Name ist eine Liste mit eindeutigen Namen der Certification Authorities, die der Server akzeptiert.

Der Client_Certificate_Type kennt folgende Zertifikatstypen:

RSA Signatur (Typ 1)

DSA Signatur (Typ 2)

RSA fixed Diffie-Hellman (Typ 3)

DSS fixed Diffie-Hellman (Typ 4)

RSA ephemeral Diffie-Hellman (Typ 5)

RSA ephemeral Diffie-Hellman (Typ 6)

Fortezza KEA (Typ 20)

Der Client kann auf diese Anforderung sein Zertifikat und seine Signatur oder eine Alert-Nachricht des Typs No_Certificate zum Server schicken. Ist eine Client-Authentifizierung per Signatur zwingend, wird in letztem Fall das Handshake-Protokoll abgebrochen. Falls ein anonymer Server eine Certificate_Request-Nachricht verschickt, wird SSL mit einem fatalen Handshake-Protokollfehler abgebrochen.

Da die obigen drei Nachrichten optional sind und der Client wissen muss, wann er alle Nachrichten des Servers erhalten hat, schickt der Server diese Nachricht, wenn er sein Server_Hello und die optionalen Nachrichten Certificate, Certificate_Request und Server_Key_Exchange verschickt hat. Da SSL das TCP-Protokoll benutzt, ist die Übertragung der Nachrichten und deren korrekte Reihenfolge garantiert. Server_Hello_Done enthält keine Parameter.

Diese optionale Nachricht wird vom Client zum Server geschickt, um sich per digitaler Signatur zu authentifizieren. Sie folgt der Nachricht, in welcher der Client sein Signaturzertifikat zum Server übertragen hat. Es wird nur ein einziger Parameter, nämlich die Signatur, übergeben. Die Signatur wird gebildet aus der Verschlüsselung des folgenden Wertes mit dem privaten Client-Schlüssel:

Anstelle von MD5 kann auch der SHA1-Algorithmus verwendet werden. Die Art der Signatur, RSA oder DAS, hängt vom Typ des Clientzertifikats ab.

client_key_exchange (Typ = 16) Diese Nachricht ist zwingend erforderlich und der Anteil des Clients am Schlüsselerzeugungsprozess. Das Format hängt vom ausgewählten Verfahren zur Schlüsselerzeugung ab. Es werden drei Fälle unterschieden:

1. Falls **RSA** eingesetzt wird, erzeugt der Client ein 48 Byte langes Pre-Master Secret und verschlüsselt es mit einem Public Key, den er entweder aus dem Serverzertifikat entnommen oder vom Server in der Server_Key_Exchange-Nachricht bekommen hat.
2. Im Fall von **Diffie-Hellman** wird unterschieden, ob der Client bereits in seinem Clientzertifikat seine öffentlichen Diffie-Hellman-Werte übertragen hat oder ob sie in dieser Nachricht noch übertragen werden müssen. Im ersten Fall wird ein leerer Parameter übertragen, da in jedem Fall eine Nachricht gesendet werden muss. Im anderen Fall wird der aus der geheimen Zufallszahl des Clients und den D/H-Parametern aus dem Serverzertifikat berechnete, öffentliche Diffie-Hellman-Wert übertragen.
3. Im Fall von Fortezza wird es komplex. Hier wird eine ganze Reihe von Parametern übertragen, auf die an dieser Stelle wegen der Irrelevanz von Fortezza für moderne und gegen US-Geheimdienste abhörsichere VPN nicht weiter eingegangen wird. Näheres findet man im aktuellen SSLv3-Dokument, in TLS wird Fortezza ohnehin nicht mehr eingesetzt.

Finished (Typ = 20) Diese Nachricht besagt, dass für den betreffenden Endpunkt das Handshake-Protokoll beendet ist. Es dient dazu, die im aktuellen Handshake-Prozess ausgehandelten Algorithmen, Initialisierungsvektoren usw. zu überprüfen. Finished ist die erste Nachricht, die mit diesen Funktionen und Werten arbeitet und die Daten verschlüsselt überträgt. Vor jeder Finished-Message muss eine Nachricht des Change_Cipher_Spec-Protokolls zur Gegenseite geschickt werden. Falls eine Finished-Nachricht empfangen wird, ohne dass der betreffende Sender vorher eine Change_Cipher_Spec-Nachricht gesendet hat, wird dies als fataler Fehler behandelt. Die Finished-Nachricht enthält zwei Parameter:

Der Wert Sender ist bei der Finished-Nachricht des Clients 0x434c4e54 und bei der des Servers 0x53525652. Die Hash-Werte werden über alle Nachrichten, außer Finished selbst, gebildet.

Ablauf des SSL Handshake Protocols

Das Handshake Protocol benutzt das SSL Record Protocol für den Austausch der Nachrichten, das SSL Alert Protocol zur Fehlerbehandlung sowie das SSL Change Cipher Spec Protocol zur Aktivierung der ausgehandelten Algorithmen und Parameter. Da beim ersten Aufbau einer SSL-Session noch keine Algorithmen und Parameter zur Verschlüsselung usw. ausgehandelt wurden, werden vom SSL-Record-Protokoll so genannte NULL-Algorithmen verwendet, welche die Daten auf sich selbst abbilden. Alle Nachrichten

sind damit unverschlüsselt und unauthentifiziert. Schutz gegen Man-in-the-Middle-Angriffe muss auf Ebene des Handshake-Protokolls implementiert werden, z.B. durch signierten Schlüsseltausch und ähnliche Methoden, wie in den obigen Nachrichtenformaten beschrieben.

Eine SSL-Session wird immer durch ein Client_Hello initiiert. Will jedoch ein Server eine SSL-Session aufgebaut haben, so animiert er den Client durch einen Hello_Request zu deren Aufbau. Der Client_Hello muss durch einen Server_Hello beantwortet werden. Diese beiden Nachrichten handeln praktisch alle Aspekte der SSL-Session aus: Algorithmen zur Verschlüsselung, Authentifizierung, Integritätsprüfung und Schlüsselerzeugung, die Session-ID, verschiedene Zufallswerte sowie das Kompressionsverfahren. Dadurch ist auch der weitere Ablauf des Handshake-Protokolls weitgehend vorbestimmt. Grundsätzlich sind beim Ablauf zwei Fälle zu unterscheiden:

1. Der Client will eine vorhergehende Session wieder aufnehmen oder die Parameter einer bereits existierenden Session für eine neue SSL-Verbindung verwenden.
2. Der Client will einen kompletten Neuaufbau einer Session.

Im ersten Fall kann der Handshake stark abgekürzt werden, sowohl von der Anzahl der Nachrichten her als auch aus Sicht der Verarbeitungszeit. Angenommen der Client wünscht die Wiederaufnahme einer Session und der Server akzeptiert diesen Wunsch und hat die entsprechenden Werte noch in seinem Cache, dann folgt dem entsprechenden Client_Hello ein Server_Hello gefolgt von gegenseitigem Change_Cipher_Spec und Finished. Alle weiteren Nachrichten und vor allem die rechenaufwändigen Schlüsselerzeugungsverfahren entfallen.

Auch bei einem vollständigen Aufbau der Session ist die Anzahl und Art der Nachrichten abhängig von dem, was in den beiden Hello-Nachrichten vereinbart wurde. Je nach vereinbarten Schlüsselerzeugungs- und Authentifizierungsverfahren sendet der Server optional sein Zertifikat (Certificate), Werte für die Schlüsselerzeugung (Server_Key_Exchange) oder eine Anforderung für ein Clientzertifikat (Certificate_Request). Nachdem alle notwendigen Nachrichten vom Server verschickt wurden, sendet dieser sein Server_Hello_Done zum Client.

Der Client hat nach seinem Client_Hello gewartet, bis vom Server ein Server_Hello, irgendwann gefolgt von einem Server_Hello_Done, gekommen ist. Nun schickt er optional, falls vom Server angefordert, sein Clientzertifikat (Certificate) zum Server. Es folgen in jedem Fall ein Client_Key_Exchange und ein optionales Certificate_Verify des Clients, falls vom Server eine Client-Authentifizierung per Signatur gefordert wurde. Nun sendet der Client eine Change_Cipher_Spec-Nachricht zum Server, aktiviert die ausgehandelten Parameter und sendet eine verschlüsselte SSL-Nachricht des Typs Finished zum Server. Der Server aktiviert, nachdem er die Change_Cipher_Spec-Nachricht erhalten hat, ebenfalls die neuen Session-Parameter und schickt seinerseits ebenfalls ein Change_Cipher_Spec und Finished zum Client.

Damit ist das SSL Handshake Protocol abgeschlossen, und es können sicher Nutzdaten zwischen Client und Server transportiert werden.

7.3.2 SSL Change Cipher Spec Protocol

Dieses Protokoll verdient den Namen Protokoll eigentlich nicht, denn es besteht nur aus einer einzigen Nachricht mit einem einzigen Parameter, einem Byte mit dem Wert 1. Diese Nachricht signalisiert, dass alle folgenden Records mit den Parametern und Algorithmen der zuvor ausgehandelten Cipher Suite geschützt werden. Nach dieser Protokollnachricht folgt praktisch immer eine Finished-Nachricht des SSL Handshake Protocols.

7.3.3 SSL Alert Protocol

Das SSL Alert Protocol dient zur Signalisierung von Fehlerzuständen. Eine Alert-Nachricht besteht aus zwei Parametern, der Dringlichkeit und der Beschreibung des Problems. Es gibt zwei Dringlichkeitsstufen:

Warning ist in der Regel ein Hinweis auf eine unerwartete Situation, die nicht zwingend den Abbruch einer Session erfordert.

Fatal bedeutet, dass die Verbindung sofort abubrechen ist und deren Parameter auch nicht mehr zur Erzeugung neuer Sessions benutzt werden dürfen.

Der SSL-Standard definiert folgende Beschreibungen von Problemen und Fehlern in SSL:

Close_Notify signalisiert der Gegenstelle die Beendigung einer Session. Beide Seiten müssen sich über den Zustand (aktiv oder beendet) einer Verbindung im Klaren sein, um so genannte Truncation Attacks zu vermeiden. Wenn eine Seite einen Closure_Notify empfängt, sendet sie ebenfalls einen Closure_Notify zurück und beendet die Session. Die Gegenseite muss normalerweise nicht auf diese Bestätigung warten.

Unexpected Message ist immer ein fataler Fehler. Er lässt auf eine inkorrekte Implementierung des SSL-Protokolls auf der Gegenseite oder einen Denial-of-Service-Angriff schließen, da dieser Zustand ansonsten nicht auftreten kann.

Bad_Record_MAC bedeutet, dass der Message Authentication Code inkorrekt ist, was ebenfalls immer ein schwerer Fehler ist.

Decompression_Failure spricht für sich selbst und ist selbstverständlich ebenfalls immer fatal.

Handshake_Failure bedeutet, dass die Verbindung nicht aufgebaut werden kann, was generell als schwerer Fehler behandelt wird.

No_Certificate ist die Antwort auf einen Certificate_Request, wenn kein passendes Zertifikat zur Verfügung steht. In der Praxis tritt dieser Fall dann ein, wenn ein Server vom Client ein Zertifikat anfordert und dieser kein passendes besitzt. Die Schwere dieses Fehlers wird von der Anwendung bzw. der Konfiguration des Servers bestimmt.

Bad_Certificate kennzeichnet ein korruptes Zertifikat oder eines, dessen Signatur nicht verifiziert werden kann.

Unsupported_Certificate wird gemeldet, wenn der Typ eines zu benutzenden Zertifikats nicht unterstützt wird.

Certificate_Revoked wird durch ein zu benutzendes Zertifikat erzeugt, dass von der CA gesperrt wurde.

Certificate_Expired ist der Fehler, den ein zeitlich abgelaufenes Zertifikat zur Folge hat.

Certificate_Unknown wird erzeugt, wenn ein Zertifikat nicht benutzt werden kann, auf das die anderen zertifikatsbezogenen Fehler aber nicht zutreffen.

Illegal_Parameter ist immer ein fataler Fehler, da in diesem Fall der Wert eines Feldes im Handshake Protocol außerhalb des erlaubten Bereichs liegt oder mit einem anderen Feld in Konflikt steht. Dieser Fehler weist ebenfalls auf eine inkorrekte Implementierung von SSL hin.

Die Nachrichten des Alert-Protokolls sind, falls das Handshake-Protokoll abgeschlossen wurde, über das darunter liegende SSL Record Protocol geschützt.

Das SSL Record Protokoll entspricht im Wesentlichen der Funktionalität von ESP in der IPSec-Protokollfamilie, so dass die grundsätzliche Funktion derartiger Sicherheitsprotokolle im Kapitel 5 nachgelesen werden kann. Allerdings ist eine SSL-Session im Gegensatz zu einer IPSec Security Association ein zustandsbehaftetes (stateful) Protokoll, da es zwingend TCP voraussetzt, während IPSec-ESP direkt auf IP aufsetzt.

0	7	15	23	31
Type	Major Version	Minor Version	Length (1st Byte)	
Length (2nd Byte)	Data			

Abbildung 7.7: Der SSL-Record-Header

Im SSL Record Protocol werden die zu übertragenden Datenpakete komprimiert, verschlüsselt und mit einer Integritätsprüfsumme versehen. Der Ablauf der Verarbeitung eines ausgehenden Pakets ist in Abbildung 7.9 zu sehen. Die Applikationsdaten werden bei Bedarf fragmentiert und anschließend komprimiert. Für jedes Fragment wird eine Integritätsprüfsumme berechnet und angehängt. Diese Prüfsumme ist ein verschlüsselter Hash-Wert des Datenfragments.

Anschließend erfolgt eine Verschlüsselung mit der zuvor durch das SSL Handshake oder Change Cipher Spec Protocol ausgehandelten Methode.

Das Ergebnis dieser Verschlüsselung bekommt den SSL-Header vorangestellt und wird dem TCP-Stack mit den entsprechenden Portnummern zur weiteren Verarbeitung übergeben.

Der SSL Record Layer erhält Daten von Anwendungen auf höheren Ebenen in Blöcken beliebiger Größe. Die Daten können entweder von SSL-Protokollen (Handshake, Alert usw.) oder von anderen Applikationen im TCP/IP-Application-Layer kommen. Die Daten werden von SSL nicht weiter interpretiert.

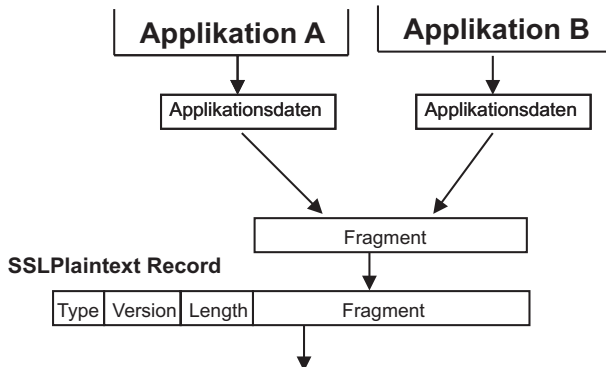


Abbildung 7.8: SSL-Fragmente können auch Daten verschiedener Applikationen enthalten.

Die Daten werden in SSLPlaintext Records in einer Größe von maximal 2^{14} Byte fragmentiert. Es wird keine Interpretation der Nutzdaten vorgenommen, in einem SSLPlaintext Record können somit auch mehrere Datenblöcke gleichen Typs enthalten sein. Der Record (vgl. Abbildung 7.7) besteht aus folgenden Feldern:

Type identifiziert den Typ der Daten im Fragment. Definiert sind `Application_Data` (Typ 23), `Handshake` (22), `Alert` (21) und `Change_Cipher_Spec` (Typ 20). Dieses Feld ist ein Byte lang.

Version, ein zwei Byte großes Feld, gibt die benutzte SSL-Version in der Form (Major Version, Minor Version) an. Für SSL 3.0 wäre der Wert 3,0, und für TLS 1.0 enthält das Feld 3,1.

Length gibt die Länge des folgenden Klartextsegments (Fragment) an. Das Feld ist zwei Byte groß, und der Wert darf 2^{14} nicht überschreiten.

Fragment enthält die Applikationsdaten, die für SSL transparent sind. Die Länge kann zwischen 2^0 und 2^{14} Byte liegen.

Der SSL Record Layer vergibt den zu übertragenden Daten unterschiedliche Prioritäten, die niedrigste haben dabei die Applikationsdaten, damit die SSL-Protokolle möglichst verzögerungsfrei miteinander kommunizieren können.

Alle SSLPlaintext Records werden mit dem im aktuellen Session Status festgelegten Kompressionsalgorithmus komprimiert bzw. dekomprimiert. Ist noch kein Algorithmus festgelegt, wird ein NULL-Verfahren benutzt, das den Inhalt nicht verändert. Die von dieser Funktion erzeugte Datenstruktur wird als `SSLCompressed Record` bezeichnet.

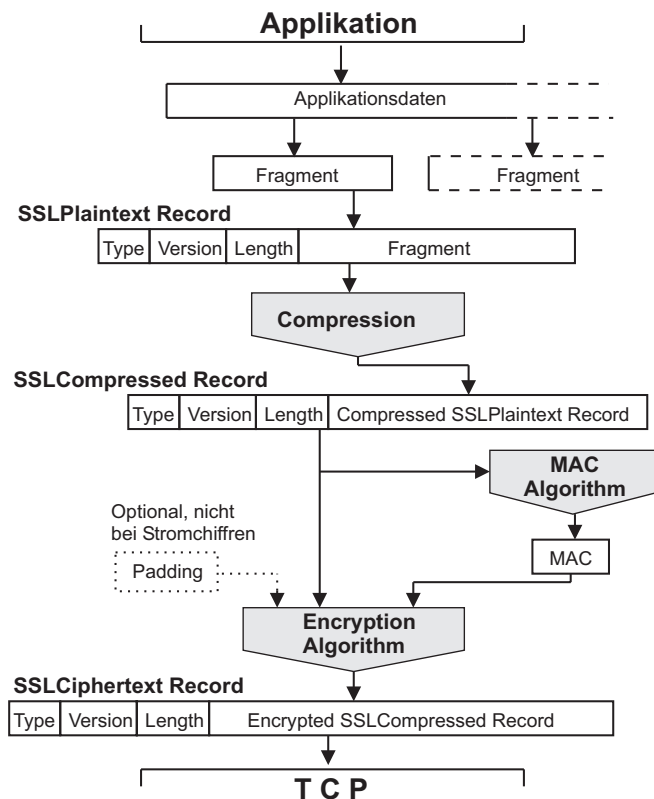


Abbildung 7.9: Die SSL-Verarbeitung ist ganz ähnlich der von IPSec, allerdings auf einer anderen Ebene des TCP/IP-Schichtenmodells.

Dieser Record ähnelt von seiner Struktur sehr dem SSLPlaintext Record und ist folgendermaßen aufgebaut:

Type identifiziert den Typ der Daten im Fragment. Dieses Feld ist ein Byte lang. Der Wert ist identisch zum gleichen Feld im SSLPlaintext Record.

Version, ein zwei Byte großes Feld beschreibt die benutzte SSL-Version in der Form (Major Version, Minor Version). Der Wert ist identisch zum gleichen Feld im SSLPlaintext Record.

Length gibt die Länge der komprimierten Daten an. Das Feld ist zwei Byte groß, und der Wert darf $2^{14} + 1024$ nicht überschreiten.

Fragment enthält den komprimierten SSLPlaintext Record. Die Länge kann zwischen 2^0 und $2^{14} + 1024$ Byte liegen.

Die Kompression muss selbstverständlich verlustfrei sein und darf die Länge des Records um nicht mehr als 1024 Byte vergrößern. Bei der Implementation der Kompressionsfunktion ist besonders darauf zu achten, dass beim Dekomprimieren keine Buffer-Overflows auftreten können. Es wird der komplette SSLPlaintext Record komprimiert.

Diese Funktion ist neben dem SSL Handshake Protocol essenziell für die Sicherheit von SSL und TLS. Welche Algorithmen in welcher Kombination und mit welchen Parametern zum Einsatz kommen, wird von der aktuellen Cipher Suite festgelegt. In SSL ist immer irgendeine Cipher Suite aktiv. Im Fall des Aufbaus einer Session, wenn vom Handshake Protocol noch keine Algorithmen usw. ausgehandelt wurden, wird automatisch SSL_NULL_WITH_NULL_NULL benutzt. Diese Cipher Suite bietet noch keinerlei Sicherheit, weder Vertraulichkeit noch Datenintegrität.

Die Reihenfolge der Verarbeitung sieht so aus, dass zuerst eine kryptografische Prüfsumme (MAC, Message Authentication Code) gebildet wird und dann die Datenstruktur aus MAC und SSLCompressed Record mit dem ausgewählten Verschlüsselungsalgorithmus zum SSLCiphertext Record transformiert wird. Dieser Record hat folgenden Aufbau:

Type kennzeichnet den Typ der verschlüsselten Daten. Das Feld ist ein Byte lang und wird von SSLCompressed.type übernommen.

Version, ein zwei Byte großes Feld beschreibt die benutzte SSL-Version in der Form (Major Version, Minor Version). Der Wert ist identisch zum gleichen Feld im SSL-Compressed Record.

Length gibt die Länge der verschlüsselten Daten an. Das Feld ist zwei Byte groß, und der Wert darf $2^{14} + 2048$ nicht überschreiten.

Fragment enthält die Ausgabe des aktiven Chiffreverfahrens. Die Länge darf $2^{14} + 2048$ nicht überschreiten.

SSL ist ein Protokoll, das sowohl Strom- als auch Blockchiffrierverfahren unterstützt. Dadurch differiert der interne Aufbau des SSLCiphertext Records, der an das TCP-Protokoll übergeben wird. Das liegt daran, dass Blockchiffrierverfahren in der Regel mit festen Blockgrößen arbeiten und deshalb Klartextblöcke auf die notwendige Länge mit beliebigen Füllzeichen (Padding) aufgefüllt werden müssen. Stromchiffren benötigen kein Padding.

Zu Prüfung der Integrität und Authentizität der übertragenen Daten werden MACs verwendet, die auf Hash-Funktionen beruhen. In SSL werden Konstrukte eingesetzt, die sehr stark an HMAC-Funktionen angelehnt sind, in TLS werden standardisierte HMAC (RFC 2104, „Keyed Hashing for Message Authentication“) eingesetzt.

MAC in SSL 3.0 Der MAC wird aus verschiedenen Eingangsgrößen gebildet. Selbstverständlich ist das SSLCompressed.fragment eine dieser Größen, denn die übertragenen Daten müssen integer sein. Zusätzlich fließt das MAC_Write_Secret mit ein, um die Authentizität zu gewähren. Beide Seiten führen einen Sequenznummernzähler für ausgehende und ankommende Pakete. Diese Sequenznummer schützt SSL vor Angriffen durch wiederholtes Senden von Paketen. Da SSL auf ein Transportprotokoll (TCP) aufsetzt, das die Zustellung und die korrekte Reihenfolge der Daten garantiert, braucht diese Sequenznummer nicht mit übertragen zu werden. Die Initialisierung des Zählers auf den Wert 0 erfolgt durch eine Change_Cipher_Spec-Nachricht. Die vollständige MAC-Berechnung erfolgt durch:

```
sequence_
numberSSLCompressed.typeSSLCompressed.lengthSSLCompressed.fragment));
```

Die restlichen, bisher nicht beschriebenen Parameter in dieser Berechnungsvorschrift haben folgende Werte und Bedeutung:

Pad_1 wird, vereinfacht ausgedrückt, dazu verwendet, um extrem unterschiedlichen Eingangsgrößen mit einem gewissen Anteil an konstanten Werten zu verknüpfen. Dies ist ein Schutz vor Kollisionsangriffen. Der Wert von pad_1 ist ein String von 48 (MD5) oder 40 (SHA1) Byte mit dem Wert 0x36.

Pad_2 hat die gleiche Funktion wie pad_1, jedoch ist dessen Wert ein String aus 48 (MD5) oder 40 (SHA1) Byte mit dem Wert 0x5C.

Sequence_number ist die Sequenznummer, die von Client und Server gepflegt wird, um Replay Attacks zu unterbinden.

Da in TLS Hash-Funktionen generell als HMAC implementiert sind, sieht die Erzeugung des MAC etwas anders aus:

```
MAC = HMAC_Hash(MAC_write_secret, sequence_number + TLSCompressed.type +  
TLSCompressed.version + TLSCompressed.length + TLSCompressed.fragment)
```

Das Padding (inner_pad, outer_pad) und die Verschachtelung der Hash-Funktionen sind bereits im HMAC-Standard spezifiziert, so dass die Schreibweise etwas einfacher ist, das Ergebnis aber prinzipiell der MAC-Berechnung in SSLv3 entspricht.

Im SSLv3-Standard ist neben einigen Blockverschlüsselungsverfahren auch ein Stromchiffrierverfahren, RC4, in einigen Cipher Suites spezifiziert. Aufgrund der Tatsache, dass SSL auf TCP aufsetzt, also alle Pakete garantiert und in der richtigen Reihenfolge übertragen werden, kann RC4 in seiner optimalen und sichersten Betriebsart arbeiten, nämlich ohne Initialisierungsvektor oder andere Synchronisierungsmaßnahmen, die in vielen anderen Implementierungen (z.B. WEP) für fatale Sicherheitslücken verantwortlich sind. Denn TCP garantiert, dass der Klartext- und Chiffretext-Bytestrom synchron sind und somit der PRNG von RC4 praktisch ununterbrochen durchlaufen kann. Die Datenstruktur, die dem RC4-Algorithmus als Strom zu verschlüsselnden Bytes zugeführt wird, besteht aus zwei Feldern:

SSLCompressed [size] ist der vollständige SSLCompressed Record.

MAC [size] ist die Prüfsumme wie oben beschrieben. Die Länge (size) ist abhängig von ausgewählten Hash-Funktion.

Falls die Cipher Suite SSL_NULL_WITH_NULL_NULL aktiv ist, ist die Länge von MAC.size = 0, und SSLCompressed ist unverändert, also im komprimierten Klartext. Der NULL-Verschlüsselungsalgorithmus wird in SSL als Stromchiffrierverfahren behandelt.

Der Länge der verschlüsselten Daten berechnet sich einfach aus der Länge des SSLCompressed Records und der Länge des MAC.

Bei den Blockchiffrierverfahren, die mit Blöcken bestimmter Größe arbeiten müssen, müssen kürzere Klartextdatenblöcke auf die erforderliche Größe aufgefüllt werden. Damit bei der Entschlüsselung auch klar ist, welche Byte Nutz- und welche Byte Fülldaten sind, muss deren Länge der Gegenseite mitgeteilt werden.

Durch die Eigenschaften von TCP braucht, anders als zum Beispiel bei IPSec, der Initialisierungsvektor überhaupt nicht übertragen zu werden. Der initiale Vektor wird vom Handshake-Protokoll erzeugt, alle weiteren aus dem letzten Block des vorhergehenden Records. Die Datenstruktur, die dem Blockchiffreverfahren blockweise zugeführt wird, sieht wie folgt aus:

SSLCompressed [size] ist der vollständige SSLCompressed Record.

MAC [size] ist die Prüfsumme wie oben beschrieben. Die Länge (size) ist abhängig von ausgewählten Hash-Funktion.

Padding [size] ist ein Folge von Bytes zum Auffüllen auf die erforderliche Blocklänge. Die Länge kann zwischen 0 und der Blocklänge -1 liegen.

Padding length ist ein Ein-Byte-Feld, das die Anzahl der Padding-Bytes angibt.

Die Länge der verschlüsselten Daten berechnet sich aus der Länge des SSLCompressed Records, der Länge des MAC, der Länge des Paddings plus 1.

Die Verarbeitung von eingehenden Records erfolgt in genau umgekehrter Reihenfolge.

SSL baut für jede Applikation, zu der sich durch einen Client verbunden wird, eine eigene Session auf. Dies bedeutet, dass unter Umständen etliche SSL-Sessions zwischen einem Client und einem Gateway existieren. Vor allem aber der Aufbau der Sessions mit mehreren RSA-, Diffie-Hellman- oder DSS-Operationen kostet enorme Rechenleistung und kann zu einer spürbaren Beeinträchtigung der Gesamtperformance führen.

Allerdings kann man den SSL-Server bzw. das SSL-VPN-Gateway so konfigurieren, dass es dem Client erlaubt, falls dieser es bei entsprechender Konfiguration wünscht, die Parameter einer bereits existierenden Session zu benutzen und damit den kompletten SSL-Handshake zu überspringen. Dies kann sogar dann funktionieren, wenn gar keine Session mehr existiert, aber die Parameter der letzten Verbindung noch im Cache gespeichert sind. Bei dieser Methode wird allerdings nur der Aufwand für den Handshake reduziert.

Bei der Auswahl eines SSL-VPN-Gateways muss man in jedem Fall darauf achten, dass ein spezieller SSL-Chipsatz mit ausreichend CPU-Leistung zur Verarbeitung von SSL-VPN-Verkehr verwendet wird. Normale Server oder IPSec-Gateways, auch solche mit Beschleuniger-Modulen, sind für SSL-VPNs oft völlig ungeeignet! Überschlüssig kann man bei einem IPSec-Gateway (mit Hardware-Beschleuniger) egal welchen Herstellers davon ausgehen, dass ein SSL-Client auf dem Gateway etwa die 20fachen Ressourcen eines IPSec-Clients verbraucht. Mit anderen Worten: Auf einem IPSec-Gateway für 1.000 Benutzer können maximal 50 SSL-VPN-Benutzer arbeiten! Es sei denn, es wird ein spezieller SSL-Beschleunigungs-Chipsatz mit einem eigenem Rechner-Chipsatz verwendet.

Moderne Chipsätze, nehmen wir beispielhaft den NSP2004B-Beschleuniger der Firma Netoctave, erreichen laut Angaben des Herstellers Durchsätze bis zu 1,3 GBit/s mit starker Verschlüsselung (128 Bit), schaffen bis zu 4000 RSA-Operationen pro Sekunde bei gleichzeitig bis zu 256.000 SSL-Connections.

Die Auswahl der Authentifizierungsmethode hat unter bestimmten Umständen einen Einfluss auf die Performance von SSL. Denn wenn eine beidseitige SSL-Authentifizierung benutzt wird, dann verdoppelt sich die Anzahl der RSA-Operationen im SSL-Handshake. Andere Methoden wie Tokenkarten oder Passwortauthentifizierung sind aus Sicht von SSL reine Applikationsdaten und haben keine besonderen SSL-Aktivitäten zur Folge.

Bei Auswahl des richtigen SSL-VPN-Gateways und der optimalen Konfiguration der entsprechenden Parameter hat man eine mit IPSec vergleichbare Performance. Der leicht höhere Paket-Overhead von SSL dürfte nicht wirklich ins Gewicht fallen. Unbedingt hüten sollte man sich vor IPSec-Gateways, die lediglich per Software-Update SSL-VPN-tauglich gemacht sind, denn hier droht wirklich der Performance-GAU. Bei der Kombination von IPSec- und SSL-VPN müssen daher entweder zwei getrennte, auf die jeweilige Technologie spezialisierte Systeme eingesetzt oder ein IPSec-Gateway mit einem speziellen SSL-Hardware-Modul ausgerüstet werden. In diesem Fall kommt man in den Genuss optimaler Performance sowohl für SSL- als auch für IPSec-VPN.

Wenn man sich die Funktion und den technischen Hintergrund von SSL vergegenwärtigt, muss man sich fragen, wie SSL denn überhaupt für VPN eingesetzt werden. Aus der Einleitung dieses Kapitels ist hervorgegangen dass SSL prinzipiell kein Tunneling kennt, keine virtuellen Verbindungen oder Pfade abbilden und damit eigentlich keine echte VPN-Technologie sein kann.

Aber das gar nicht so schlimm, denn wenn man den Tunneling-Aspekt vorerst einmal außer Acht lässt und das Ganze Thema richtig auf eine sichere Übertragung zwischen zwei Applikationen reduziert, dann kann man SSL sehr gut für Remote-Access- oder Extranet-Anwendungen benutzen – wenn man es richtig macht und sich neben den Vorteilen auch möglicher Risiken und Probleme bewusst ist und diese damit schließlich vermeiden kann.

HTTP Application Proxy

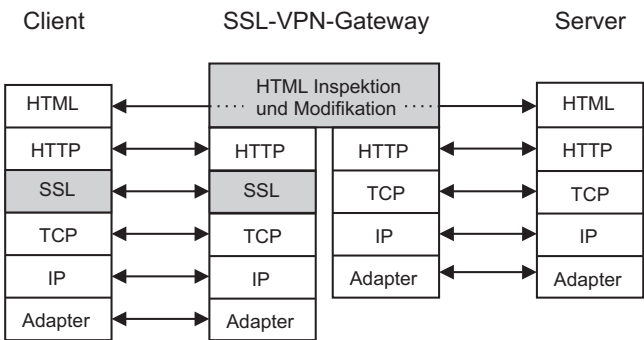


Abbildung 7.10: Selbst in der einfachsten Variante muss das SSL-VPN-Gateway die Applikationsdaten inspizieren und modifizieren.

Wie bereits gesagt, kann man SSL-VPN nicht für Standortverbindungen, sondern ausschließlich für Remote Access oder Extranets einsetzen. Das Szenario ähnelt dabei dem von IPSec Remote Access: Ein Benutzer baut von einem Rechner aus eine sichere Verbindung zu einem entsprechenden SSL-VPN Gateway auf, das den Zugriff auf die gewünschte Applikation im Unternehmensnetzwerk gewährt. An dieser Stelle ist es wichtig zu erwähnen, dass SSL für jede Applikation, auf die ein Client zugreift, eine eigene Connection aufbaut. Das ist ein fundamentaler Unterschied zu normalen IPSec-Clients, die für alle Daten nur eine einzige Verbindung, bestehend aus einer IKE-SA und zwei IPSec-SAs, aufbauen. Dies ist aber nicht etwa ein Nachteil von SSL, im Gegenteil, es erlaubt vielmehr auf der Gateway-Seite eine sehr differenzierte Zugangssteuerung auf Applikationsebene. Der Aufwand beim Aufbau von nachfolgenden Sessions kann in SSL dadurch stark reduziert werden, dass einfach die meisten Parameter der bereits zu einem Gateway aufgebauten Verbindung benutzt werden.

SSL-VPNs werden in der Regel in verschiedenen Modi betrieben:

Im einfachsten Fall arbeitet der Benutzer mit dem Webbrowser ohne Zusatzfunktionen oder ladbare Module. Die Funktionalität ist entsprechend eingeschränkt, und es stehen nur Applikationen zur Verfügung, die entweder selbst SSL benutzen, HTTP-basierend sind, also zum Beispiel Webmailer, Outlook Web Access usw., oder wenn im SSL-VPN-Gateway eine entsprechende Anwendungsübersetzung erfolgt. Dies ist auch gleichzeitig die flexibelste Möglichkeit, denn diese Funktionalität kann von überall benutzt werden, auf eigenen oder fremden Rechnern, mit jedem Browser, auch wenn auf diesem Java und ActiveX gesperrt sind.

Application Translation

z.B. Filesharing (CIFS <-> HTML)

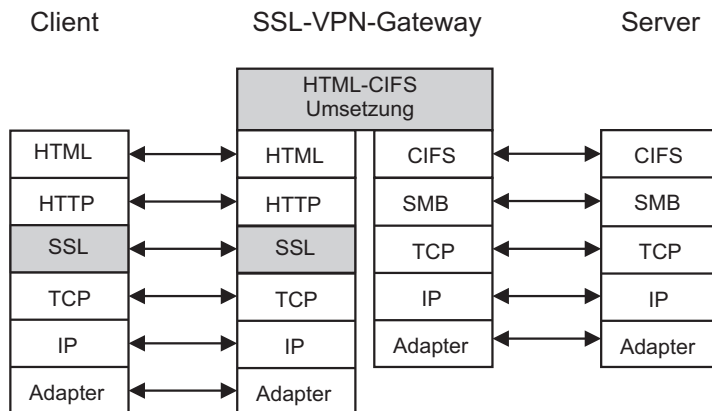


Abbildung 7.11: Für bestimmte Applikationen führt das SSL-VPN-Gateway eine vollständige Übersetzung einer Anwendung in HTML durch.

Dieser Modus ist allerdings nicht benutzertransparent. Der Anwender muss daher genau wissen, was er beziehungsweise sein Rechner tut, um die entsprechende SSL-VPN-Funktionalität manuell, meist über einen Portal-Link, zu starten. Dies bedingt eine entsprechende Schulung der Benutzer.

Hier werden über zusätzliche Funktionen, in der Regel Port Forwarder oder spezielle so genannte Application Level Proxies, weitere Anwendungen in den Genuss gebracht, SSL benutzen zu können. Bei den fortschrittlicheren Herstellern wird in dieser Betriebsart Applikationszugriff sowie File- und Printsharing über SSL benutzbar, was die Palette der benutzbaren Anwendungen schon deutlich erhöht. Hierfür werden vom Browser die für die jeweilige Applikation notwendigen Java-Applets oder ActiveX-Controls heruntergeladen und ausgeführt. Allerdings existiert auch hier keinerlei Transparenz: Es werden nur die Applikationen sicher übertragen, für die dies im SSL-VPN-Gateway explizit vorgesehen und technisch möglich gemacht wurde. Dies bedeutet je nach Anzahl der Anwendungen einen erheblichen Verarbeitungsarbeit, denn das Gateway übersetzt gewissermaßen die ganzen Applikationen in das http-Format.

Auch dieser Modus ist nicht benutzertransparent, hier gilt das Gleiche wie beim browser-based Modus.

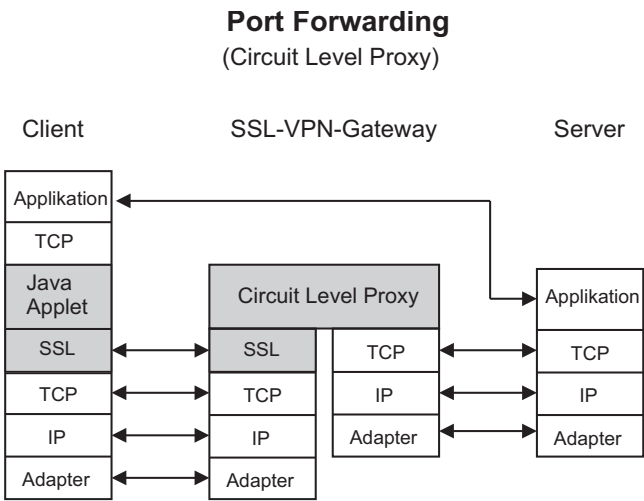


Abbildung 7.12: Im Enhanced Mode werden Java-Applets vom Gateway geladen, um TCP-Pakete per Port-Forwarding über SSL zu transportieren.

Die Lösung für dieses Problem, zumindest in Teilen, stellen die SSL-VPN-Clients dar. Dies sind Programme, die in der Regel auf Socket- oder Adapter-Ebene arbeiten und sich praktisch in die TCP/IP-Transportschicht einklinken und alle TCP-, UDP- und teilweise auch IP-Pakete zur SSL-Verarbeitung umleiten. Der Nachteil: Man muss diesen Client installieren und hat damit den Vorteil eines client-losen VPN wieder verspielt.

Einige SSL-VPN-Clients basieren auf Abwandlungen des SOCKS-Protokolls (SOCKS, RFC 1928), das als Sicherheitserweiterung für TCP-Verbindungen entwickelt wurde. Die Version 5 dieses Protokolls kann zusätzlich auch UDP-Pakete verarbeiten. Allerdings muss man sich vergegenwärtigen, dass UDP durch SSL in TCP eingekapselt wird und damit alle Eigenschaften von TCP aufgezwungen bekommt. Nun wird aber UDP oft dort eingesetzt, wo das Verhalten von TCP hinsichtlich Bandbreitensteuerung und wiederholtem Senden verloren gegangener Pakete für bestimmte Anwendungen völlig ungeeignet ist.

Zwei dieser Anwendungen sind zum Beispiel Voice over IP (VoIP) oder Video-Conferencing. Hierfür ist ein SSL-VPN nicht optimal, denn das zugrunde liegende Real Time Protocol (RTP) funktioniert nur mit UDP einwandfrei. Wiederholtes Senden verlorener Pakete ist dabei absolut tabu! Ebenso wirkt das Zwischenspeichern (Verzögerung) von Paketen, um diese in einem größeren Block zu versenden, der korrekten Funktion von RTP entgegen. Auch die Flusssteuerung von TCP führt eher zu Jitter, als dass sie für RTP einen praktischen Nutzen bringt.

Socks-based Tunneling

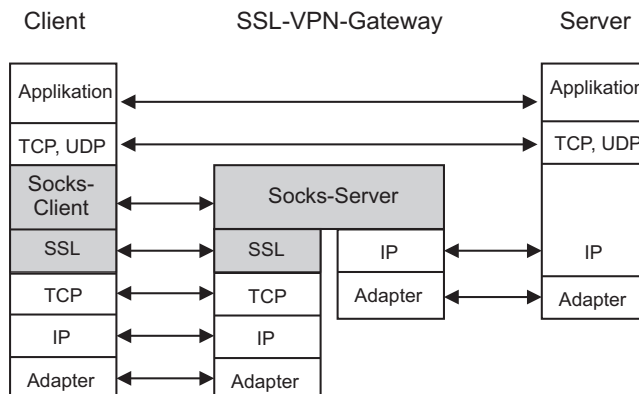


Abbildung 7.13: Socks-basierende Clients übertragen transparent alle TCP- und UDP-Pakete über SSL.

Andere Clients arbeiten auf Adapterebene, also unterhalb von IP, und bieten damit eine völlige Applikationstransparenz. Allerdings gibt man damit die Flexibilität der Zugangssteuerung im Gateway auf, denn es existiert zwischen einem Client-Rechner und dem SSL-VPN-Gateway nur eine einzige SSL-Session mit einer Connection.

Die möglichen Probleme bei Echtzeitanwendungen existieren auch hier, denn SSL bietet keine Möglichkeit für QoS-Signalisierung. Dieser Modus ist bei entsprechender Konfiguration des Clients benutzertransparent.

Virtueller Adapter

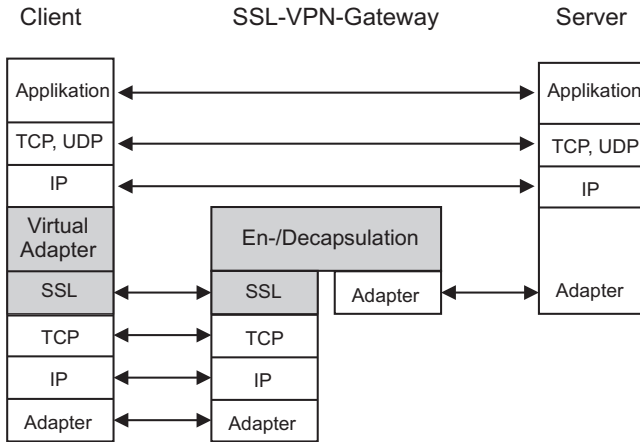


Abbildung 7.14: Fast schon Overkill: SSL-Clients auf Basis eines virtuellen Adapters arbeiten transparent.

SSL-VPNs sollen ohne die Installation von Clients arbeiten, aber andererseits ist für eine transparente TCP- und UDP-Übertragung ein Client notwendig. Für dieses Problem haben die findigen Entwickler eine Lösung parat: Der Client wird nicht mehr fest auf den Benutzerrechnern installiert, sondern zur Laufzeit bei Bedarf über den Browser vom Gateway als ActiveX-Control oder Java Applet heruntergeladen. Damit gelangt man wieder zu dem Vorteil, keine Clients flächendeckend installieren und verwalten zu müssen, und hat eine transparente Sicherheit für jegliche TCP- und UDP-basierende Kommunikation. Natürlich sind durch die ActiveX-Technologie andere Betriebssysteme als Windows ausgeschlossen, hier bietet sich Java als Alternative an. Aber auch diese Methode hat einen Wermutstropfen: Es wird nicht unbedingt von jeder Unternehmenssicherheitsrichtlinie erlaubt, dass über den Microsoft Internet Explorer ActiveX-Controls installiert werden, die mit Administratorrechten laufen müssen – wie es ein solcher Client, der in den TCP/IP-Stack eingreift, in der Regel erfordert.

Die verschiedenen Modi können weitgehend miteinander kombiniert werden bzw. ergänzen sich hervorragend. Insbesondere die Kombination aus dem Browser-, Enhanced-Browser- und Network-Modus bietet den größten Charme, denn hier muss wirklich nichts auf den Client-PCs installiert und gewartet werden. Und ein Browser ist mittlerweile auf jedem Betriebssystem und jedem Rechner vorhanden, vom Server bis zum Palmtop. Und falls der Hersteller sowohl Active-X- und/oder Java-Implementierungen anbietet, ist man obendrein auch ziemlich unabhängig vom Clientbetriebssystem.

Das SSL-VPN-Gateway spielt die zentrale Rolle in einem SSL-VPN. Bei den allgemeinen Funktionen wie Routing, Redundanz, Systemsicherheit, sicheres und flexibles Management, Leistung und Erweiterbarkeit gelten die gleichen Kriterien wie bei IPSec- oder beliebigen anderen VPN-Gateways, so dass diese Punkte hier nicht nochmals breitgewalzt werden müssen. Hier wird deshalb nur auf einige SSL-Spezifika eingegangen.

Heutige SSL-VPN-Gateways sollten (und haben meist auch) eine vollständige SSL- und TLS-Implementierung. Ein Algorithmus wie der AES gehört mittlerweile zum guten Ton und darf in keinem System fehlen, auch wenn er in den ursprünglichen SSL-Dokumenten (letzter Stand: 1996) wegen seines damaligen Nichtexistierens nicht erwähnt ist. Dass in einem SSL-VPN-Gateway als zentralem Konzentrador sehr vieler SSL-Sessions entsprechende Chipsätze zur Beschleunigung der SSL-Verarbeitung enthalten sein müssen, versteht sich von selbst. Eine reine Software-Lösung, egal auf welcher Serverplattform und unter welchem Betriebssystem, ist zum Scheitern verurteilt, wenn es mehr als 20 oder 30 Benutzer sein sollen. Beschleuniger-Hardware ist zum Teil mehr als 100- bis 1000-mal leistungsfähiger, insbesondere beim Aufbau von SSL-Sessions.

Da über SSL-VPN-Gateways Benutzer authentifiziert werden, muss eine breite Palette verschiedener Verfahren nutzbar sein, um in verschiedene Netzwerklandschaften integrierbar zu sein. Neben der ebenso unvermeidlichen wie unsicheren Client-Authentifizierung per UserID und Passwort sollten in jedem Fall folgende Verfahren implementiert sein:

- Digitale Signatur (innerhalb von SSL)

- Tokenkarten

Als Backend-Architektur müssen folgende Systeme unterstützt werden:

- Lokale Benutzerdatenbank

- Radius (CHAP, PAP, ggf. MS-CHAP)

- LDAP

- Tokenserver

- PKI

Optional können noch ältere oder proprietäre Verfahren wie NTLM, TACACS oder Siteminder unterstützt werden.

Diese Verfahren müssen vom SSL-Gateway fast alle als Applikation nach erfolgtem SSL-Verbindungsaufbau gestartet werden, denn das SSL-Protokoll erlaubt nur eine beidseitige oder einseitige Authentifizierung per digitaler Signatur – oder gar keine Authentifizierung.

Da das SSL-Protokoll definitionsgemäß Sicherheit zwischen zwei Applikationen bereitstellt, hat man hier einen viel größeren Spielraum als mit IPSec-Gateways. Aufgrund der Tatsache, dass für jede Applikation eine eigene SSL-Session benötigt wird, hat man die

Möglichkeit einer echten, differenzierten Zugriffssteuerung auf Applikationen. In Verbindung mit Benutzer- und Gruppenberechtigungen kann man hier sehr vielschichtig Berechtigungen vergeben. Das ist für normalen Remote Access eigener Benutzer meist nicht notwendig, dafür aber bei Extranet-VPN-Anwendungen um so mehr.

Beim Einsatz von SSL-VPN-Clients entfällt dieses Feature leider vollständig, denn hier existiert ebenso wie bei IPSec zwischen Client und Gateway eine einzige Verbindung, über die alle Applikationen transportiert werden.

Ich habe oft die Erfahrung gemacht, dass vielen Anwendern nicht so ganz klar ist, wie ein SSL-VPN eigentlich genau funktioniert. Meist wurde ihnen eingeredet, sie können mit SSL einfach IPSec ablösen, und das war es. Nun IPSec überträgt IP und damit jede Applikation, die IP benutzt. Und SSL überträgt http – und das war es. Und für jede andere Anwendung muss auf dem Gateway genau diese Anwendung in http übersetzt werden. Zwischen Client und Gateways läuft https und zwischen Gateway und Applikationsserver das Applikationsprotokoll. Mit anderen Worten, für jede Applikation muss auf dem Gateway eine geeignete Funktionalität implementiert werden. Ein SSL-VPN-Gateway ist denn auch tatsächlich ein echtes Gateway im klassischen Sinn, denn es arbeitet auf allen TCP/IP-Schichten.

Dies gilt nicht für SSL-VPN-Clients, hier hat man, abgesehen von Quality of Service, Applikationstransparenz.

Im Folgenden beschränke ich mich auf Sicherheitsbetrachtungen zu SSLv3. Ältere Versionen sind heute in SSL-VPN nicht mehr relevant. Hierbei sind zwei Dinge ganz klar voneinander zu trennen: die Sicherheit des Protokolls in seiner korrekten Implementierung und die Sicherheit der Umgebung, in der SSL eingesetzt wird.

SSL ist keine allein stehende Applikation, sondern eine Funktionalität innerhalb eines Protokoll-Stacks, der wiederum von verschiedenen Anwendungen benutzt werden kann. Hier gibt es wechselseitige Abhängigkeiten. Heutige Browser sind in der Regel gegen die meisten Angriffe gewappnet – allerdings sollte man das lieber auch überprüfen.

Hier versucht ein Angreifer, entweder indem er jemandem einen gefälschten Link schickt oder indem er eine Verbindung stiehlt, einen Benutzer auf eine gefälschte Website zu locken. Diese Webseite hat eine URL, die sich nur in einem, leicht verwechselbaren Zeichen von der korrekten unterscheidet. Üblich sind die Null (0) anstelle des Buchstabens O oder die Eins (1) anstelle des kleinen Buchstabens l und so weiter. Auf der

gefälschten Website, die zum Beispiel dem originalen SSL-VPN-Portal täuschend ähnlich sieht, wird versucht, dem Benutzer seine Anmeldeinformationen zu entlocken. Man-in-the-Middle-Angriffe können dies völlig unbemerkt tun, sie können unter Umständen sogar gültige Zertifikate einer im Browser hinterlegten öffentlichen CA benutzen, und der Inhaber des Zertifikats kann sogar mit dem Webserver des Angreifers übereinstimmen. Es erscheint zu keinem Zeitpunkt ein verdächtiges Warnfenster, nichts, was irgendwie verdächtig aussieht.

Technisch gesehen ist solch ein Angriff auf verschiedene Arten machbar. Wenn der Angreifer Zugriff auf das LAN des Clients oder des Servers hat, ist das Ganze sogar echt einfach. Denn hierfür gibt es einfach zu bedienende, zuverlässig arbeitende Public Domain Tools, die über LAN-Spoofing unbemerkt Ethernet-Frames umleiten können.

Üblicherweise reagiert ein korrekt arbeitender Browser auf Zertifikate einer unbekannten Certification Authority oder auf Zertifikate mit einem anderen als den in der URL angegebenen Namen durch eine entsprechende Warnmeldung, die vom Benutzer bearbeitet werden muss. Hier muss eine entsprechende Schulung der Benutzer die notwendige Sensibilität erzeugen, nicht einfach die Meldung zu ignorieren bzw. blind zu bestätigen und damit das obskure Zertifikat zu akzeptieren.

Das Problem ist, dass viele Benutzer durch die ständig auftauchenden Warnungen bei der Installation nicht von Microsoft signierter Software diese Warnmeldungen teilweise schon reflexartig mit OK bestätigen.

Die folgenden Angriffe sind alle auf hinsichtlich der SSL-Spezifikation korrekte Implementierungen möglich. Glücklicherweise sind die meisten Angriffe jedoch durch geeignete Maßnahmen bei der SSL-Implementierung zu verhindern, ohne dabei die SSL-Kompatibilität der jeweiligen Implementierung zu verlieren. Allerdings muss der Hersteller die Angriffe und die Verwundbarkeiten kennen und wissen, mit welchen Tricks und Workarounds diese neutralisiert werden können.

SSL bietet, wie viele andere Security-Protokolle, z.B. IKE (vgl. Kap. 6) auch, eine bestimmte Funktionalität, die es Angreifern ermöglicht, Pre-Shared Secrets, Benutzerkennungen oder Schlüssel zu ermitteln. Diese Funktionalität nennt man ein (informationstheoretisches) Orakel, und ein bekannter Angriff, der diese Orakel-Funktionalität bestimmter Algorithmen oder Protokolle ausnutzt, ist der Bleichenbacher-Angriff.

Dieser Angriff, eine Chosen-Ciphertext-Attacke, die versucht, das Pre-Master Secret einer SSL-Session zu erraten, macht sich im wesentlichen zwei Fakten zunutze:

1. Eine mit RSA verschlüsselte Nachricht lässt sich relativ leicht entschlüsseln, wenn man einige Bits der Nachricht entweder kennt oder mit einer gewissen Wahrscheinlichkeit vorhersagen kann. (Keine Panik, dann ist aber nur der Klartext zu diesem speziellen Chiffretext bekannt, nicht aber der private Schlüssel!)

2. Das SSLv3-Protokoll überträgt das Pre-Master Secret, von dem alle weiteren Schlüssel abgeleitet werden, in der Client-Key-Exchange-Nachricht in einer PKCS#1-codierten Form. Hierin sind insgesamt 40 Bit an festen Positionen bekannt.

Der Angreifer schickt nun eine Reihe ausgewählter Pakete zum Server und prüft dessen Antworten. Manche Server-Implementierungen prüfen nun zuallererst das Padding und schicken gegebenenfalls eine Meldung zurück, dass das Padding inkorrekt ist. Diese Meldung (die Antwort des Orakels) gibt dem Angreifer darüber Aufschluss, ob er mit seinem gewählten Chiffretext richtig liegt oder nicht. Über eine Reihe von Versuchen, maximal etwa 2^{16} , kann man das Pre-Master Secret einer belauschten und aufgezeichneten SSL-Session berechnen.

Es ist nun wichtig, zu prüfen oder sich vom Hersteller glaubhaft versichern zu lassen, dass die eingesetzte SSL-Server- oder SSL-VPN-Gateway-Implementierung vor diesem Angriff sicher ist. Dies kann man per Implementierung erreichen, indem man vor dem Prüfen des Paddings zuerst einmal den MAC-Wert berechnet und überprüft. Dieser ist in jedem Fall inkorrekt. Aber selbst auf einem verwundbaren System kann man einen Bleichenbacher-Angriff zumindest erkennen, und zwar an den Hunderttausendenden von Meldungen, die der Server erzeugt, weil das Padding in der Client-Key-Exchange-Nachricht fehlerhaft war.

Dieser Angriff funktioniert nur mit SSL 2.0 oder falls ein SSL 3.0-Server und -Client im 2.0-Kompatibilitätsmodus arbeiten. SSL 2.0 bietet kaum Integritätsschutz beim Handshake-Protokoll, vor allem fehlt die Finished-Nachricht, die den kompletten Handshake nachträglich verifiziert. Hier kann ein Man-in-the-Middle eine Liste von Cipher Specs gegen eine eigene, mit schwachen Verfahren belegte Liste austauschen. Der Benutzer bemerkt das normalerweise nicht.

Um einen Client und einen Server, die beide SSL 3.0 unterstützen, zum Arbeiten im SSL 2.0-Kompatibilitätsmodus zu verleiten, kann ein Man-in-the-Middle den Version-Rollback-Angriff benutzen.

Dieser Angriff, wie alle anderen möglichen Angriffe auf SSL auch, ist ein Man-in-the-Middle-Angriff. Der Angreifer ändert ein Client-Hello im Version-3-Format in ein Version-2-Format und schickt dies zum Server. Der Server glaubt, dass der Client kein SSL 3 kennt, und antwortet mit seinem Server-Hello im Version-2-Format. Der Client bekommt dieses Paket, nimmt an, dass der Server kein SSL 3 unterstützt, und fährt mit einem Handshake der SSL Version 2 fort. Damit hat der Man-in-the-Middle sowohl Server als auch Client zur Verwendung von SSL Version 2 gebracht und kann die Schwächen dieser Protokollversion ausnutzen.

Dieser Angriff kann nur erfolgen, wenn der Server so konfiguriert ist, dass er abwärtskompatibel zu SSL Version 2 arbeiten kann. Heutige SSL-VPN-Gateways oder SSL-Server können jedoch meist auch so konfiguriert werden, dass sie nur SSLv3 oder TLS unterstützen. Das Manko, dass damit uralte Browser, die kein SSLv3 oder TLS können, nicht mehr unterstützt werden, wird durch eine entsprechend hohe Sicherheit wieder

wettgemacht. Im SSL-VPN hat man in der Regel auch Kontrolle über die Clients (Browser) und kann generell nur mit TLS oder SSLv3 arbeiten. In Extranets hat man diese Kontrolle natürlich nicht.

Falls unbedingt eine Abwärtskompatibilität zu SSLv2 notwendig ist, kann man sich mit einem kleinem, aber wirkungsvollen Trick behelfen: Der Client schreibt die höchste von ihm unterstützte SSL-Version in das Padding des PKCS#1-Block der Client-Key-Exchange-Nachricht, der durch eine RSA-Verschlüsselung geschützt ist. Diese Nachricht kann vom Man-in-the-Middle nicht gefälscht werden, ohne RSA zu brechen. Damit wird der Angriff spätestens dann entdeckt, wenn der Server die Client-Key-Exchange-Nachricht erhält. Dieser Trick ist voll konform zur SSLv3- und TLS-Protokollspezifikation.

Dieser Angriff beruht darauf, dass der Angreifer dem Client als Server vorspiegelt, RSA zu benutzen, während dem Server vorgemacht wird, der Client wolle Diffie-Hellman benutzen. Beide schicken sich nun die jeweiligen Key-Exchange-Nachrichten zu. Nachdem der Angreifer die Key-Exchange-Nachricht des Clients abgefangen hat, der mit Diffie-Hellman-Parametern eine RSA-Verschlüsselung durchgeführt hat, kann er das Pre-Master Secret entschlüsseln!

Denn der Server sendet eine große Primzahl n , einen Generator g (gerne verwendet man hier die Zahl 2) und seinen öffentlichen D/H-Parameter zum Client. Dieser nimmt an, der erste Wert (n) sei der RSA-Modulus (keine Primzahl) und der zweite Wert (g) sei der RSA-Exponent. Nun führt der Client eine RSA-Verschlüsselung des Pre-Master Secrets m durch: $k = m^g \pmod n$. Der Man-in-the-Middle fängt k ab berechnet einfach die g -te Wurzel aus k . Dies funktioniert, da der Wert n im Gegensatz zum echten RSA-Modulus, der das Produkt zweier großer Primzahlen und damit nicht prim ist, eine Primzahl ist. Damit können vom Angreifer über das Pre-Master Secret (m) die Sitzungsschlüssel berechnet, der Handshake vervollständigt und letztendlich die Finished-Nachricht gefälscht werden.

Diesem Angriff kann man unter Beibehaltung der SSL-Kompatibilität verhindern, indem die SSL-Implementierung die Anzahl der übertragenen Parameter prüft. Diffie-Hellman benötigt drei, RSA nur zwei Werte. Wenn der Client auf einen RSA Public-Key wartet und drei anstelle von zwei Parametern bekommt, kann der Handshake sofort abgebrochen werden, und der Angriff läuft ins Leere.

Gegen all diese Angriffe gibt es Workarounds oder gehärtete Implementierungen, die einen gewissen Schutz bieten, ohne dass man die Spezifikationen von SSL oder TLS ändern oder verletzen müsste. Man muss nur sicherstellen, dass diese oder vergleichbare andere Maßnahmen im SSL-VPN implementiert sind. Das ist bei den SSL-VPN-Gateways meist das kleinere Problem. Das Hauptproblem sind die ganzen unterschiedlichen Browser mit ihren verschiedenen Versionen, Service Packs, Patches und nachinstallierbaren Modulen. Hier ist absolute Versionskontrolle und vor allem absolute Restriktion angesagt.

Ein wohlmeinender Benutzer, der eigenmächtig Patches, Service Packs oder andere ihm sinnvoll erscheinende Erweiterungen für den Browser auf seinem Rechner installiert, kann dabei massive Sicherheitslöcher einbauen. Hier ist mit den entsprechenden Mitteln der Betriebssysteme, eventuell flankiert von Client-Policy-Enforcement-Anwendungen wie dem Nortel TunnelGuard, Sygate (Symantec) Enterprise oder ähnlichen Applikationen, dafür Sorge zu tragen, dass der Browser in seiner getesteten und frei gegebenen Version verbleibt; so lange, bis eine neue Version von der Administration ausgegeben wird.

Aufgrund der aus Sicherheitssicht unzulänglichen Qualität heutiger Browser-Software – betroffen sind, wenn auch in unterschiedlichem Maß, alle Hersteller – muss man leider bei jeder neuen Version, jedem neuen Service Pack und jedem neuen Patch prüfen, ob all die bekannten Angriffe gegen SSL in der vorliegenden Version tatsächlich noch abgeblockt werden oder nicht. Und man muss prüfen, ob nicht neue, bis dato unbekannte Schlupflöcher für böswillige Angreifer geöffnet wurden.

8

L2TP/IPSec-Transport

L2TP/IPSec-Transport, oft auch als L2TP-over-IPSec oder einfach L2TP/IPSec bezeichnet, ist eine Kombination aus einem reinen Sicherheitsprotokoll, IP Security (IPSec), und einem speziell für Remote Access entwickelten Tunneling-Protokoll, dem Layer 2 Tunneling Protocol (L2TP). L2TP/IPSec hat mittlerweile eine gewisse Bedeutung gewonnen, da es von Microsoft seit Windows 2000 als VPN-Protokoll favorisiert wird. Viele Hersteller von VPN-Routern tragen dem Rechnung, indem ihre Systeme neben den eigen entwickelten IPSec-Clients auch die L2TP/IPSec-Clients von Microsoft terminieren können.

Das Layer 2 Tunneling Protocol wurde entwickelt, um PPP (Point-to-Point Protocol, ein Verfahren zum Transport von Netzwerkprotokollen über Punkt-zu-Punkt-Verbindungen) die Datenkommunikation über andere als Punkt-zu-Punkt-Verbindungen zu ermöglichen. Sein vorrangiges, aber nicht ausschließliches Einsatzgebiet ist daher der „virtuelle“ Remote Access über ein IP-Netzwerk.

L2TP ist als Verbesserung und Weiterentwicklung zweier älterer, nicht standardisierter Verfahren, des Point-to-Point Tunneling Protocols (PPTP) und des Layer 2 Forwardings (L2F), zu verstehen. Die generelle Struktur und die jeweiligen Vorteile beider Verfahren wurden in das Layer 2 Tunneling Protocol, das im RFC 2661 beschrieben ist, übernommen. Obwohl sowohl PPTP als auch L2F heute noch in bestehenden Installationen verwendet werden, findet keine Weiterentwicklung der beiden Verfahren mehr statt.

Die Entwicklung von L2TP ist zweigleisig, getrieben von zwei völlig unterschiedlichen Einsatzszenarien: dem Einsatz in Enterprise Ende-zu-Ende-VPN und als Protokoll mit dem Carrier und Service Provider Dial-Verbindungen in ihren POPs (Point-of-Presence) terminieren, aggregieren und durch einen Tunnel über IP zum Endkunden schleusen können. In diesem Buch konzentrieren wir uns auf erste Variante und deren Client-Implementierung durch Microsoft.

8.1 Das Point-to-Point Protocol (PPP)

Um L2TP richtig verstehen zu können, muss man auch die Arbeitsweise von PPP kennen. Das Point-to-Point Protocol wurde entwickelt, um verschiedenste Netzwerkprotokolle wie IP, IPX, Appletalk usw. über eine Punkt-zu-Punkt-Verbindung zu übertragen. Derartige Verbindungen kommen bei leitungsvermittelnden Netzen oder Festverbindungen vor. Vor der Entwicklung von PPP wurden diese Verbindungen mit proprietären, meist auf Basis von HDLC (High-Level Data Link Control) arbeitenden Verfahren betrieben. PPP ermöglichte erstmals Punkt-zu-Punkt-Verbindungen auch zwischen Produkten verschiedener Hersteller. Mittlerweile ist dieses Protokoll in praktisch jedem Router und jedem Rechner (z.B. Microsoft DFÜ-Netzwerk, Unix-pppd) implementiert.

Der Zugriff auf das Internet über Wählverbindungen, den die Internet Service Provider (ISP) weltweit anbieten, basiert fast ausschließlich auf PPP; auch bei Festverbindungen wird es sehr oft benutzt.

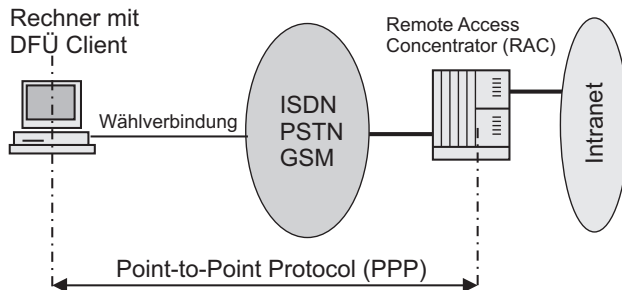


Abbildung 8.1: Ein typisches Remote-Access-Szenario mit PPP über Wählverbindungen

8.1.1 Remote Access mit PPP

In Abbildung 8.1 sehen Sie ein typisches Unternehmensszenario für einen Remote Access, in dem sich ein Rechner per Wählverbindung über einen Remote Access Concentrator (RAC) Zugriff auf ein Intranet verschafft. Sowohl im PC als auch im RAC übernimmt das PPP-Protokoll die Aufgabe, diese Punkt-zu-Punkt-Verbindung zu initialisieren, aufrechtzuerhalten und zu beenden. In PPP sind alle Gegenstellen (Peers) gleichwertig, es gibt weder eine dedizierte Server- oder Client-Funktionalität noch ein rollenbasierendes Verhalten wie Initiator oder Responder. Es ist auch nicht unbedingt notwendig, dass die PPP-Gegenstellen immer den gleichen Funktionsumfang unterstützen. Die Kommunikationsoptionen und -parameter können während der Initialisierungsphase untereinander ausgehandelt werden. Lediglich wenn bestimmte Funktionen von einer Seite als zwingend erforderlich angesehen werden und die Gegenstelle diese nicht in ausreichend erscheinendem Maße unterstützen kann, wird ein Verbindungsaufbau abgebrochen. Dies ist aber nur in ganz wenigen Fällen gegeben.

8.1.2 Die Komponenten von PPP

PPP basiert auf dem HDLC-Protokoll, allerdings nur insoweit, als es dessen Rahmenstruktur benutzt. Die HDLC-Adressierung wird nicht benutzt, da nur zwischen zwei Endpunkten kommuniziert wird. Die Adressfelder dienen nur zur Identifikation von PPP-Rahmen, indem dort feste, definierte Werte eingesetzt werden.

PPP kann man – in Abhängigkeit von ihrer Funktion – in verschiedene, in der PPP-Terminologie als Sublayer bezeichnete Schichten untergliedern. Wie Sie in Abbildung 8.2 sehen, gibt es für jede dieser Funktionen in den verschiedenen Schichten Steuerprotokolle zur Aushandlung und Konfiguration dieser Dienste und die implementierten Dienste selbst, welche die zu übertragenden Pakete entsprechend umformen oder Prozeduren wie die Authentifizierung durchführen.

Verarbeitung	Steuerungs- protokoll
Multiplex-Schicht - Netzwerkprotokoll Verteilung	IPCP IPXCP
Dienste-Schicht - Kompression - CHAP, PAP - Verschlüsselung	CCP Auth. ECP
Medienabhängige Schicht - HDLC Adressierung - Framing, Escaping - Frame Check Sequence (FCS)	LCP

Abbildung 8.2: Die verschiedenen Verarbeitungsschichten und Komponenten von PPP

Die Multiplex-Schicht

In dieser Schicht konfigurieren die Steuerprotokolle die spätere Behandlung und den Transport von Paketen der zu übertragenden Netzwerkprotokolle (wie IP oder IPX). Dies betrifft die Bekanntgabe oder Zuweisung von Netzwerkadressen, die Übermittlung notwendiger Parameter für bestimmte Protokolle usw. Die beiden wichtigsten Protokolle sind das IPCP (IP Control Protocol) und das IPXCP (IPX Control Protocol). PPP ermöglicht die Übertragung einer ganzen Reihe weiterer Netzwerkprotokolle, jedoch hat sich IP derart stark durchgesetzt, dass die anderen praktisch keine Rolle mehr spielen.

In der Datenphase wird in dieser Schicht das Multiplexen und Demultiplexen der Netzwerkpakete vorgenommen. Da PPP die gleichzeitige Übertragung verschiedener Protokolle ermöglicht, müssen ausgehende Pakete gekapselt und gekennzeichnet werden. Ankommende Pakete müssen aufgrund der Kennzeichnungen in den PPP-Frames zur korrekten Weiterverarbeitung in den richtigen Netzwerk-Stack geleitet werden.

Die Dienste-Schicht

In diesem Bereich werden Dienste angesiedelt, welche die zu übertragenden Daten umformen oder Funktionen wie die Authentifizierung durchführen. Üblicherweise sind im PPP die Authentifizierungsprotokolle PAP, CHAP und MS-CHAPv2 verfügbar; mittlerweile bietet das EAP (Extensible Authentication Protocol) auch weiterreichende Funktionen und die Möglichkeit, herstellerspezifische Verfahren einzubinden. Weitere Dienste sind die Datenkompression und in manchen, seltenen Fällen auch die Datenverschlüsselung auf PPP-Ebene.

Entsprechende Steuerprotokolle handeln zwischen den PPP-Gegenstellen die notwendigen Optionen und Parameter aus. Als wichtigstes Protokoll in diesem Bereich ist das CCP (Compression Control Protocol) zu nennen, das die Datenkomprimierung steuert und damit Bandbreite und dadurch Übertragungskosten einsparen kann.

Die mediumabhängige Schicht

Hier spielt als Steuerprotokoll das LCP (Link Control Protocol) die Hauptrolle. Sobald über eine Punkt-zu-Punkt-Verbindung eine PPP-Session gestartet wird, erwacht das LCP zum Leben und konfiguriert das Verhalten der Gegenstellen. LCP definiert bestimmte Dienste, wie zum Beispiel die Authentifizierung, die in der Dienste-Schicht angesiedelt sind. Somit erstreckt sich der Wirkungsbereich des Link Control Protocols über die Grenzen der medienabhängigen Schicht hinaus.

Die Dienste in dieser Schicht übernehmen die Konstruktion von PPP-Frames, die Berechnung von Prüfsummen und alles andere, was notwendig ist, um ein PPP-Paket an die darunter liegende Übertragungsschicht weiterzuleiten.

8.1.3 PPP-Steuerprotokolle und -Dienste

Die Steuerprotokolle haben eine zentrale Bedeutung, denn sie handeln die gewünschten Dienste zwischen zwei PPP-Gegenstellen aus und konfigurieren sie anschließend auch interaktiv. Man unterscheidet dabei grundsätzlich zwischen Link-, Dienst- und Netzwerksteuerprotokollen, je nachdem, auf welcher Ebene sie angesiedelt sind. Im Folgenden werden einige wichtige und in praktisch allen PPP-Implementierungen vorhandene Protokolle erläutert:

- Das Link Control Protocol (LCP)
- Das Password Authentication Protocol (PAP)
- Das Challenge Handshake Authentication Protocol (CHAP)
- Das MS-CHAPv2 Protocol
- Das Compression Control Protocol (CCP)
- Das IP Control Protocol (IPCP)

Das Link Control Protocol (LCP)

Das LCP (Link Control Protocol) ist das Protokoll, das immer in der Startphase jeder PPP-Verbindung ausgeführt wird. Es legt die eigentlichen, medienabhängigen Parameter fest und handelt anschließend die im Weiteren auszuführenden Steuerprotokolle und Dienste aus. Dies sind in der Praxis Authentifizierungsverfahren, Datenkompression und Netzwerksteuerprotokolle zur Konfiguration der Übertragung von IP, IPX usw. über die PPP-Verbindung. Diese Konfiguration wird in Form einer Reihe von Anfragen, Antworten, Änderungsvorschlägen oder Ablehnungen durchgeführt. Dieser Vorgang kann auch asynchron erfolgen; eine Gegenstelle wartet mit ihrer nächsten Anfrage nicht, bis ihre vorherige beantwortet ist. In einer Anfrage können auch mehrere Optionen enthalten sein. Im PPP-Standard sind Configure Request (REQ), Configure Acknowledge (ACK), Configure Negative Acknowledge (NAK) und Configure Reject (REJ) als LCP-Botschaften zwischen PPP-Peers spezifiziert.

Das Password Authentication Protocol (PAP)

PAP ist ein relativ einfach aufgebautes Authentifizierungsprotokoll. Es wird von der zeitlichen Abfolge her zwischen der LCP- und der NCP-Phase ausgeführt und entscheidet, ob eine Verbindung weiter aufgebaut oder unmittelbar beendet wird. Es ist in einem eigenen Standard, dem RFC1334, beschrieben. Die Authentifizierung erfolgt, indem die Seite, die ihre Identität nachweisen muss, der Gegenstelle einen Authenticate Request schickt, in dem die Peer ID (üblicherweise der Benutzername) und ein Passwort im Klartext enthalten sind. Die Gegenstelle überprüft das Passwort und schickt, falls es richtig ist, ein Authenticate Acknowledge, ansonsten ein Authenticate Negative Acknowledge zurück und beendet die Verbindung.

Dies ist natürlich vom Standpunkt der Sicherheit aus betrachtet nicht gerade berauschend, denn jeder, der Zugriff auf die Übertragung oder Leitung hat, bekommt eine User ID samt zugehörigem Passwort praktisch auf dem Präsentierteller geliefert. Dies hat zur Entwicklung eines weiteren Authentifizierungsprotokolls geführt, des Challenge Handshake Authentication Protocols.

Das Challenge Handshake Authentication Protocol (CHAP)

Dieses Protokoll ist etwas fortschrittlicher als PAP, denn hier erfolgt die Authentifizierung in einem dreiphasigen Handshake-Verfahren, ohne dass dabei überhaupt ein Passwort übertragen wird. Es wird stattdessen ein Hash-Wert übertragen, der aus dem so genannten CHAP-Challenge-Wert, einer Zufallszahl, und dem Benutzerpasswort berechnet wurde. Die Gegenseite, die ebenfalls das Benutzerpasswort kennen muss, kann ebenfalls diesen Hash-Wert berechnen und beide Werte vergleichen. Sind sie identisch, dann hat der Benutzer das korrekte Kennwort eingegeben und ist authentifiziert.

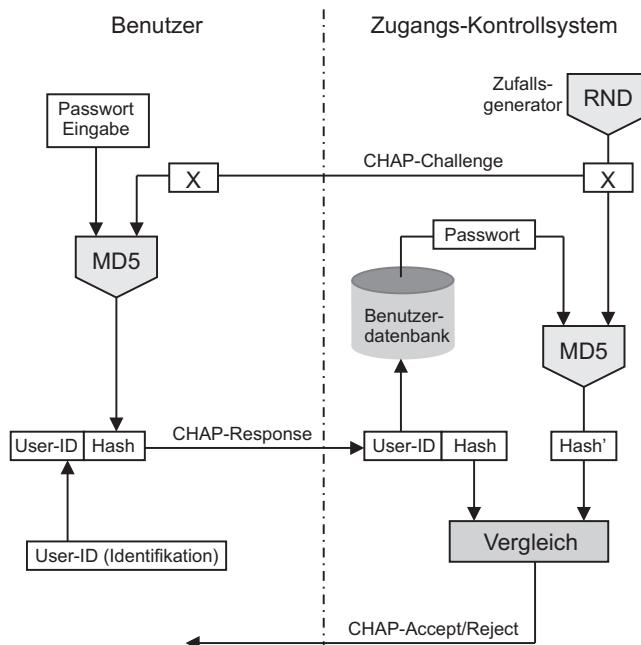


Abbildung 8.3: CHAP ist eine der Authentifizierungskomponenten in PPP.

CHAP arbeitet dreiphasig. In der ersten Phase übermittelt der Authenticator (in unserem Fall der L2TP Network Server) dem Client eine Zufallszahl, das CHAP-Challenge. Im zweiten Schritt erzeugt der Client einen Hash-Wert aus seinem vom Benutzer eingegebenen Passwort und dem CHAP-Challenge. Dieser Hash-Wert wird zusammen mit der Benutzer-ID zum Authenticator gesendet. Dieser schaut in seiner Benutzerdatenbank nach, welches Passwort zu der empfangenen Benutzer-ID gespeichert ist, und bildet daraus und aus dem CHAP-Challenge ebenfalls einen Hash-Wert. Stimmen beide Hash-Werte überein, ist der Benutzer authentisch, und der Client erhält eine CHAP-Accept-Nachricht. Im negativen Fall sendet der Authenticator eine CHAP-Failure-Nachricht und bricht den Verbindungsaufbau ab.

Das Wesentliche bei CHAP ist, neben seiner höheren Sicherheit, auch die Option, das Protokoll während einer bestehenden PPP-Session mehrfach in unregelmäßigen, für Dritte nicht vorherbestimmbaren Zeitabständen auszuführen. Damit kann man verhindern, dass ein Angreifer die Verbindung „stiehlt“ und mit gefälschten IP-Adressen die Kommunikation weiterzuführen versucht. Dies kann er nur bis zum nächsten CHAP-Zyklus tun, denn dann muss er als PPP-Peer einen Hash-Wert berechnen, ohne dass er das CHAP-Secret (Passwort) kennt. Der Angreifer kann dann entweder gar keine oder nur eine falsche CHAP-Response generieren, und die Verbindung wird beendet.

CHAP, und das ist weniger schön, bedingt allerdings, dass Authenticator das Benutzerkennwort in der gleichen Form zur Verfügung hat, in der es vom Benutzer eingegeben wurde. Im Klartext! Das bedeutet, das Benutzerkennwort muss entweder im Klartext oder mit einer reversiblen Verschlüsselung gespeichert sein, beides in keiner Weise besonders sicher. Hier müssen auf den an der Authenticator-Funktionalität beteiligten Systemen knallharte Sicherheitsrichtlinien existieren und durchgesetzt werden. Ansonsten baut man sich eine Hintertür in sein Netzwerk ein.

CHAP ist eine unidirektionale Authentifizierung. Müssen sich beide Gegenseiten authentifizieren, dann muss CHAP zweimal in beiden Richtungen ablaufen.

Das Microsoft Challenge Handshake Authentication Protocol Version 2 (MS-CHAPv2)

MS-CHAPv2 ist ein Protokoll, das erstens eine etwas höhere Sicherheit als CHAP aufweist und zweitens gleichzeitig eine bidirektionale Authentifizierung durchführt.

Der Ablauf ist im Prinzip der gleiche wie bei CHAP, jedoch sind die Nachrichten anders aufgebaut. In der ersten Nachricht vom Authenticator stehen eine Session ID und der Client-Challenge-Wert. Der Client antwortet mit seiner User ID, einem Peer-Challenge-Wert und einem Hash aus dem Client-Challenge, dem Server-Challenge der Session ID und seinem Kennwort.

Der Authenticator überprüft nun den Hash anhand des Benutzerkennwortes. Ist er nicht korrekt, dann wird eine Failure-Nachricht gesendet und der Verbindungsabbau abgebrochen. Im anderen Fall schickt der Authenticator eine Success-Nachricht und fügt in diese seine eigene Authentifizierung ein. Diese besteht aus einem Hash über den beiden Challenge-Werten, dem Hash des Clients und dem Kennwort.

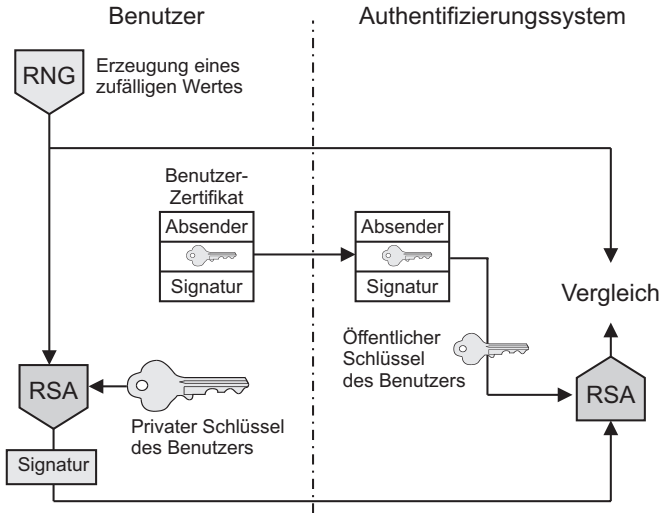


Abbildung 8.4: EAP kann PPP um die Möglichkeit der Authentifizierung per digitaler Signatur erweitern.

Der Client berechnet nun ebenfalls diesen Hash-Wert mit seinem eigenen Benutzerkennwort. Ist das Ergebnis identisch mit dem empfangenen Hash, dann weiß der Client, dass der Authenticator ebenfalls sein Kennwort kennt und damit ebenfalls authentifiziert ist.

MS-CHAP erzeugt darüber hinaus noch zwei Sitzungsschlüssel, die aber für L2TP/IPSec-Transport belanglos sind. Darüber hinaus ist deren Erzeugung, falls man den Verbindungsaufbau abhören kann, mit einem einfachen Wörterbuchangriff zu knacken, da die Schlüssel vom Challenge-Wert (bekannt) und vom Kennwort (unbekannt, aber vom Benutzer zu merken und in der Regel mehrere Wochen gültig) abgeleitet werden. Diese unselige Erzeugung von Sitzungsschlüsseln ist übrigens der Hauptgrund für die Schwäche von PPTP in der Microsoft-Variante.

Extensible Authentication Protocol (EAP)

EAP ist eine Erweiterung (RFC2284) des PPP-Protokolls, um auch andere, aktuellere und sicherere Verfahren als MD5 oder CHAP einzusetzen. Insbesondere wurde dabei an digitale Signaturen, Zwei-Faktor-Authentifizierungen und biometrische Verfahren gedacht. EAP ist im Prinzip ein Framework, das dynamisch um weitere Verfahren erweiterbar ist und deren generelle Einbettung ein PPP spezifiziert. Es steht dem Hersteller frei, die für seine Anwendungen notwendigen Verfahren zu implementieren.

EAP wurde aufgrund seiner Flexibilität mittlerweile auch für LAN-Anwendungen (EAP over LAN, EAPoL) und Wireless-LAN-Sicherheit (z.B. EAP-TLS, EAP-Tunneled TLS) angepasst.

Das Compression Control Protocol (CCP)

Mit dem CCP handeln PPP-Gegenstellen einen Algorithmus zur Datenkompression aus. Es ist eine ganze Reihe von Algorithmen spezifiziert, jedoch werden von den meisten PPP-Implementierungen nur wenige unterstützt. Die häufigsten sind die Algorithmen

von Stack Electronics (Stack LZS) und Microsoft (MPPC), eine Variante von LZS. Im Remote-Access-Bereich sollte aufgrund der Dominanz von Microsoft bei den Clients in jedem Fall dessen Point-to-Point Compression (MPPC) verfügbar sein.

Das IP Control Protocol (IPCP)

Das wichtigste Protokoll im Bereich der Netzwerksteuerprotokolle ist das IP Control Protocol (IPCP), das die Konfiguration des Internet-Protokolls innerhalb der PPP-Session steuert. Neben der Publizierung oder Zuweisung von IP-Adressen und Netzwerkmasken können optional auch DNS- oder WINS-Adressen zugewiesen werden. Wie in allen Steuerprotokollen werden hier keine IP-Pakete selbst übertragen, dies erfolgt erst in der Datenphase.

8.1.4 Der PPP-Verbindungs Aufbau

In Abbildung 8.5 ist der zeitliche Ablauf des Aufbaus einer PPP-Verbindung von der Initialisierung durch eine Seite bis hin zur Datenphase dargestellt. In der LCP-Phase werden die physikalischen Verbindungsparameter und weitere Steuerprotokolle ausgehandelt. Sobald sich die Gegenstellen geeinigt haben, wird das spezifizierte Authentifizierungsprotokoll, im beschriebenen Fall CHAP, gestartet. Falls die Authentifizierung erfolgreich ist, werden die weiteren Protokolle gestartet, falls nicht, wird die PPP-Verbindung sofort beendet. In unserem Beispiel konfiguriert das CCP (Compression Control Protocol) anschließend das Datenkompressionsverfahren, das verwendet werden soll. Falls sich die beiden Gegenstellen nicht einigen können, wird hier, im Gegensatz zur fehlgeschlagenen Authentifizierung, die Verbindung nicht terminiert, sondern die Daten werden eben unkomprimiert übertragen.

Anschließend wird das IPCP (IP Control Protocol) ausgeführt, das die beiden Gegenstellen für die Übertragung von IP-Paketen über die PPP-Verbindung konfiguriert. Es können auch während einer Session mehrere Netzwerkprotokolle wie IPX, NetBEUI, Vine-SIP usw. übertragen werden.

PPP ist üblicherweise so implementiert, dass nur in ganz wenigen Fällen keine Verbindung zustande kommt. Sobald beide Gegenstellen in der Lage sind, PPP-Pakete zu übertragen, und die Authentifizierung erfolgreich war, gibt es immer einen kleinsten gemeinsamen Nenner, auf den sich beide Gegenstellen einigen können. Wenn bestimmte Funktionen oder Parameter nicht möglich sind, ihr Fehlen aber keine fatalen Folgen hat, werden sie ignoriert, oder es werden andere Werte ausgehandelt – Hauptsache, man hat eine Verbindung. Es gibt in der Praxis auch nur wenige Fälle, in denen ein Abbruch erfolgt, obwohl eine Übertragung von PPP-Paketen möglich wäre: bei fehlgeschlagener Authentifizierung, falls überhaupt kein Netzwerksteuerprotokoll ausgehandelt werden konnte oder falls eine Seite PPP-Verschlüsselung will und die Gegenseite diese nicht unterstützt.

Somit stellt sich PPP als wirklich robustes Protokoll auf OSI-Ebene 2 dar, das aus diesem Grund eine weite Verbreitung im Bereich von Wähl- und zunehmend auch von Festverbindungen gefunden hat. Insbesondere Einwahlkonzentratoren benutzen fast ausschließlich PPP. Es hat sich daher auch als integraler Bestandteil von Betriebssystemen wie Windows, Mac OS, Linux oder OS/2 durchgesetzt, so dass dort keine zusätzliche Software für Remote-Access-Funktionalität mehr notwendig ist.

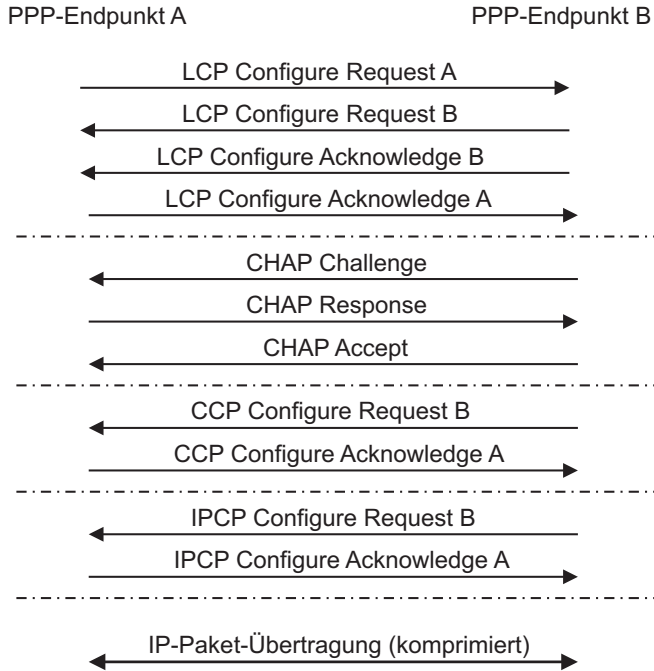


Abbildung 8.5: Der Verbindungsaufbau in PPP

8.2 L2TP

L2TP wurde ursprünglich mit dem Ziel entwickelt, Carriern und Service Providern ein Tunneling-Protokoll zur Verfügung zu stellen, bei dem sie alle Analog- und ISDN-Calls terminieren konnten, die PPP-Verbindungen aber bis in das Unternehmensnetz ihrer Kunden reichten. Für die Strecke zwischen Einwahlknoten (POP, Point of Presence) und dem Router des Kunden wurde ein Tunneling-Protokoll benötigt, L2TP.

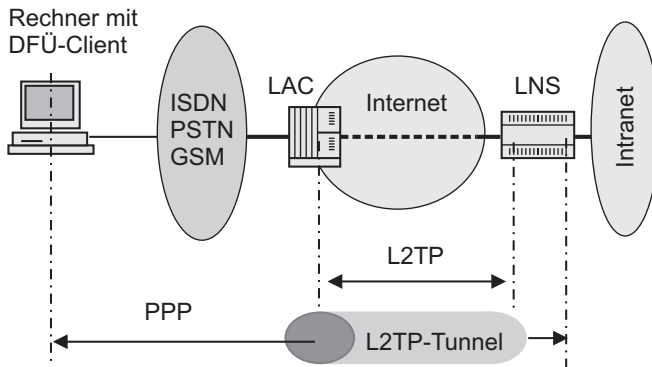


Abbildung 8.6: Beim Remote Access mit L2TP sind der Punkt der Einwahl und des Netzzugangs nicht mehr in einem System vereint, sondern auf LAC und LNS verteilt.

8.2.1 Virtueller Remote Access mit L2TP

In Abbildung 8.7 sind schematisch die Komponenten von L2TP dargestellt. Durch den L2TP-Tunnel werden sowohl die PPP-Sessions als auch eine Steuerverbindung übertragen, über die der LAC und der LNS miteinander kommunizieren. Als Tunnelmedium dient heutzutage fast ausschließlich UDP/IP; dies ist jedoch nicht zwingend, man kann mit L2TP auch direkt auf niedrigeren Protokollen wie ATM oder Frame Relay aufsetzen – oder auch auf Ethernet, denn auf diese Weise, nämlich L2TP über Ethernet, realisiert man in der Regel die PPP-Multilink-Multichassis-Funktionalität in Einwahl-Konzentratoren. In einem L2TP-Tunnel können mehrere PPP-Sessions existieren. Sobald die letzte Session abgebaut wurde, baut die Steuerverbindung auch den Tunnel selbst ab.

In der Spezifikation des Layer 2 Tunneling Protocols werden bestimmte Begriffe verwendet, deren Bedeutung im L2TP im Folgenden geklärt werden sollte.

Call

Als Call bezeichnet man eine PPP-Verbindung innerhalb eines L2TP-Tunnels. Der Name kommt daher, dass es sich bei der physikalischen Verbindung meist um eine Wählverbindung (Telefonie, Switched Virtual Circuits usw.) handelt.

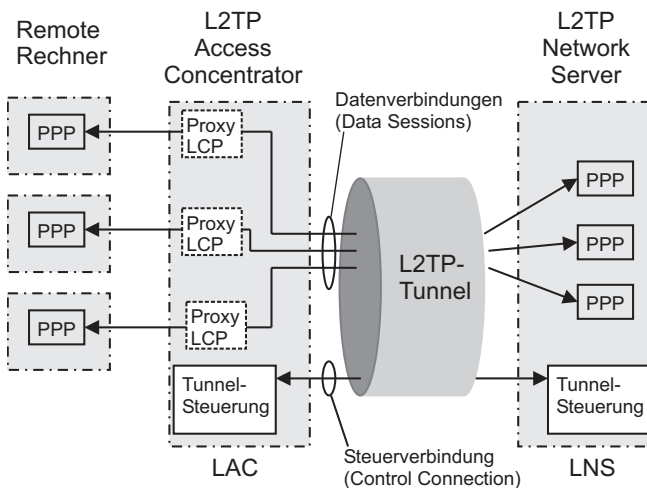


Abbildung 8.7: Die verschiedenen Komponenten des L2TP

Innerhalb eines L2TP-Tunnels können ein, mehrere oder auch gar kein Calls aktiv sein. In der Microsoft-Variante von L2TP/IPSec existiert jeweils ein Tunnel mit einer Control Connection und einem Call.

Control Connection

Innerhalb eines L2TP-Tunnels existiert genau eine Steuerverbindung zwischen LNS und LAC. Diese Verbindung steuert den Auf- und Abbau von Calls und Tunnels und überwacht den Zustand eines existierenden Tunnels.

Tunnel

Im Sinne von L2TP existiert ein Tunnel zwischen LAC und LNS, sobald eine Control Connection zwischen beiden besteht. Wenn diese existiert, ist der Tunnel vollständig aufgebaut.

8.2.2 Der LAC (L2TP Access Concentrator)

Der L2TP Access Concentrator (LAC) ist für die Verarbeitung der medienabhängigen PPP-Schichten verantwortlich und entscheidet, ob eine eingehende Verbindung als normale PPP-Verbindung lokal terminiert wird oder ob ein Tunnel zu einem LNS aufgebaut werden muss. Je nachdem, auf welche Art diese Entscheidung getroffen wird – aufgrund des DNIS (Diald Number Information Service) oder des Benutzernamens –, führt der LAC zusätzlich noch einen Teil der Authentifizierung durch. Wie bereits erwähnt wurde, ist die LAC-Funktionalität meist auf Dial-in-Systemen in den POPs von Service Providern und Carriern implementiert. Solche Systeme sind in der Lage, sowohl selbst PPP zu terminieren als auch PPP-Verbindungen zu tunneln. Die zusätzliche L2TP-LAC-Funktionalität bringt in den meisten Implementierungen keine Performanceprobleme mit sich, da meist nur ein Tunnel zu den Kundennetzen aufgebaut werden muss, durch den dann mehrere Calls gleichzeitig übertragen werden können. Außerdem findet der größte Teil der PPP-Verarbeitung dabei nicht mehr auf dem LAC, sondern auf dem LNS statt. Dies spart ebenfalls Speicher- und Rechenzeitressourcen auf dem Einwahlsystem.

Virtual LAC

In Sonderfällen, wie dem Microsoft L2TP/IPSec VPN Client, kann die vollständige LAC-Funktionalität auch auf dem Client liegen. Damit sind nur der Clientrechner und das VPN-Gateway des Kunden in die L2TP-Funktionalität involviert. Somit sind Gateway und Client gleichzeitig Endpunkte für PPP, L2TP und gegebenenfalls IPSec.

Damit wird diese Variante von L2TP zu einem Ende-zu-Ende-Modell. Carrier oder Service Provider sind nicht mehr involviert.

8.2.3 Der LNS (L2TP Network Server)

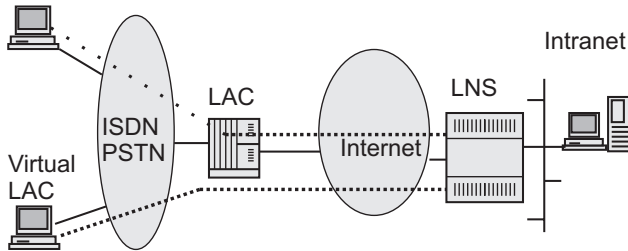
Der L2TP Network Server (LNS) ist das Gerät, auf dem die PPP-Verbindungen der Remote-Access-Clients terminiert werden. Hier arbeiten alle Dienste zum Ver- und Entpacken der Netzwerkprotokolle, der Datenkomprimierung, der Authentifizierung und der Kontrolle der Verbindungsqualität. Lediglich die medienabhängigen Dienste werden vom LAC abgewickelt. Zusätzlich zur LNS-Funktionalität muss eine Schnittstelle zu Authentifizierungs- und Accounting-Systemen existieren, meist zu RADIUS-Servern, die sich als dominierender Standard hierfür durchgesetzt haben.

8.2.4 Die L2TP-Tunneling Modelle

Das Layer 2 Tunneling Protocol kann in zwei verschiedenen Modellen (vgl. Kapitel 3) betrieben werden, dem Provider-Enterprise-Modell (Compulsory Tunneling) und dem Ende-zu-Ende-Modell (Voluntary Tunneling).

In Abbildung 8.8 sind beide Modelle dargestellt. Die beiden Modelle unterscheiden sich durch den Punkt, an dem der L2TP-Tunnel initiiert wird. Der Remote-Access-Client A wählt sich in den Einwahlkonzentrator eines Service Providers ein und wird an dieser Stelle zum LNS seines Zielnetzwerks getunnelt. Der Client selbst hat keinen Einfluss auf das Tunneln, dies wird allein vom LAC gesteuert. Aus diesem Verhalten ist auch die englische Bezeichnung Compulsory Tunneling, zwangsmäßiges Tunneln, abgeleitet. Mit diesem Modell kann der Service Provider einem Kunden garantieren, dass dessen Clients jedes Mal, wenn sie sich anmelden, unbedingt zu einem entsprechend eingerichteten LNS getunnelt werden und keinen direkten Zugriff auf das Netzwerk des Providers oder das Internet haben. Das Entscheidungskriterium für den LAC ist hier die angerufene Nummer (DNIS) oder ein Zusatz im Benutzernamen.

Compulsory Tunneling



Voluntary Tunneling

..... L2TP-Tunnel PPP-Session

Abbildung 8.8: L2TP unterstützt sowohl zwangsweises Tunneling (oben) als auch von Benutzer initiiertes Tunneling (unten).

Voluntary Tunneling, freiwilliges Tunneln, ist das andere Modell, das mit L2TP realisiert werden kann. In dieser Variante ist der Remote Access Concentrator des Service Providers in keiner Weise in das Tunneln involviert. Stattdessen ist die LAC-Funktionalität auf dem Client, Client B in Abbildung 8.8, implementiert. Der Tunnel beginnt also bereits im Clientrechner (virtueller LAC) und wird im entsprechenden LNS terminiert. Der Einwahlkonzentrator selbst überträgt dabei nur IP-Pakete zwischen Client und LNS. In diesem Fall kann der Client selbst entscheiden, ob und wie getunnelt wird.

Dieses Modell wird insbesondere bei Microsoft VPN Clients eingesetzt.

8.2.5 L2TP-Paketformate

In L2TP existieren zwei funktionell grundsätzlich verschiedene, aber ähnlich konstruierte Arten von Paketen. Dies sind zum einen Datenpakete, die PPP-Frames innerhalb eines Calls zwischen LAC und LNS transportieren, und zum anderen Steuerungspakete, die innerhalb einer Steuerverbindung zwischen LAC und LNS verschickt werden.

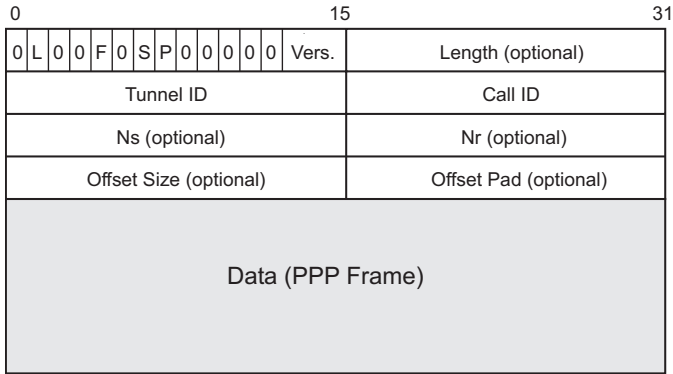


Abbildung 8.9: Das Format des L2TP-Datenpakets

In Abbildung 8.9 ist die Struktur eines L2TP-Datenpakets zu sehen. Die Hauptfunktion ist die Kapselung eines PPP-Rahmens in ein L2TP-Paket und die Zuordnung des Pakets zu einem Tunnel und einem Call innerhalb dieses Tunnels. Weiterhin können optional Sequenzzähler zur Steuerung der Paketreihenfolge und Füllzeichen (Padding) enthalten sein. Die Felder in einem Datenpaket haben folgende Bedeutung:

T-Bit Das T-Bit ist das erste Bit in einem L2TP-Paket und zeigt an, ob es sich um ein Datenpaket (0, wie in Abbildung 8.9) oder um ein Steuerungspaket (1, wie in Abbildung 8.10) handelt.

L-Bit Dieses Bit zeigt an, ob in dem Paket ein Längenfeld (1) enthalten ist oder nicht (0). In Steuerungspaketen muss ein Längenfeld enthalten sein, in Datenpaketen kann in vielen Fällen darauf verzichtet werden.

F-Bit Wenn dieses Bit gesetzt (1) ist, dann sind im Paket die Sequenzzähler Ns und Nr enthalten. In Steuerungspaketen sind diese obligatorisch, in Datenpaketen sind sie optional.

S-Bit Falls das Offset-Size-Feld vorhanden ist, muss das S-Bit gesetzt werden. In L2TP-Steuerungspaketen muss dieses Bit immer 0 sein, da dieses Feld dort nicht existiert.

Version Dies ist die Versionsnummer von L2TP; der augenblicklich einzig gültige Wert ist 2.

Length Dieser 16-Bit-Wert gibt die vollständige Länge des L2TP-Pakets in Byte an. Ein Paket kann somit maximal 65.535 Byte groß sein.

Tunnel ID Dies ist der eindeutige Bezeichner eines bestimmten L2TP-Tunnels. Er wird gemäß der L2TP-Spezifikation auf Zufallsbasis erzeugt. Mit Hilfe der Tunnel ID kann der Empfänger des Pakets dieses einem bestimmten Tunnel zuordnen.

Call ID Die Call ID ist ein innerhalb eines Tunnels eindeutiger Bezeichner eines Calls, also einer PPP-Session. Auch er wird auf Zufallsbasis erzeugt. Durch die Größe des Bezeichners von 16 Bit ist die maximale Anzahl von Calls, die in einem Tunnel existieren können, auf 65.535 beschränkt.

Ns Dieser Wert (Ns, Next Send) stellt die Sequenznummer eines Pakets dar. In Steuerungspaketen muss dieser Wert enthalten sein, in Datenpaketen ist er optional und dient dort zur Steuerung der korrekten Paketreihenfolge.

Nr Next Received (Nr) ist die Sequenznummer des Steuerungspakets, das der Sender des vorliegenden Pakets als Nächstes empfangen möchte. In Datenpaketen hat dieser Wert keinen Sinn und wird daher auf 0 gesetzt.

Offset-Size Diesen Wert gibt es nur in Datenpaketen, falls dort das S-Bit gesetzt wurde. Er gibt an, wie viele Byte zwischen dem L2TP-Header und dem PPP-Frame liegen.

Offset-Pad In den Zwischenraum zwischen L2TP-Header und PPP-Frame wird, falls der Wert Offset-Size größer 0 ist, die Anzahl von Füllzeichen geschrieben, die dem Wert von Offset-Size entspricht. Die Füllzeichen sind in der Regel Bytes mit dem Wert 0x00.

8.2.6 L2TP Attribute Value Pairs (AVP)

In Steuerungspaketen sind so genannte AVPs (Attribute Value Pairs) enthalten, die diejenigen Befehle und Optionen enthalten, die zwischen LAC und LNS ausgetauscht werden. Diese Datenstrukturen bestehen aus einem festen Header, der auch den Bezeichner des jeweiligen Attributs enthält, und einem Attributwert variabler Länge. In Abbildung 8.10 folgen zwei verschiedene AVPs dem L2TP-Steuerungspaket-Header.

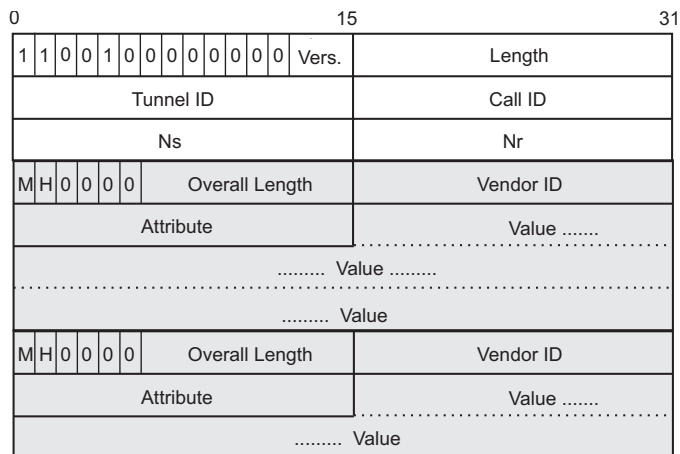


Abbildung 8.10: So sieht ein L2TP-Steuerungspaket mit zwei AVPs (Attribute Value Pair) aus.

Im AVP-Header finden wir verschiedene Flags und Felder mit folgender Bedeutung:

M-Bit Dieses Flag (M, Mandatory) zeigt an, ob der AVP zwingend ist, und steuert damit das Verhalten des Empfängers, falls dieser ein AVP mit gesetztem M-Bit empfängt, aber mit dem Inhalt des AVP nichts anfangen kann. In diesem Fall, der dann vorliegt, wenn entweder die Vendor-ID, der Attributbezeichner oder beide unbekannt sind, muss der Empfänger den Tunnel beenden.

H-Bit Das H-Bit (Hide-Bit) wird dann gesetzt, wenn der Wert des AVP versteckt sein soll. Dies wird durch die Verschlüsselung des Value-Feldes erreicht. Die ersten sechs Bytes (Flags, Overall Length, Vendor-ID und Attribute) eines AVP sind immer unverschlüsselt.

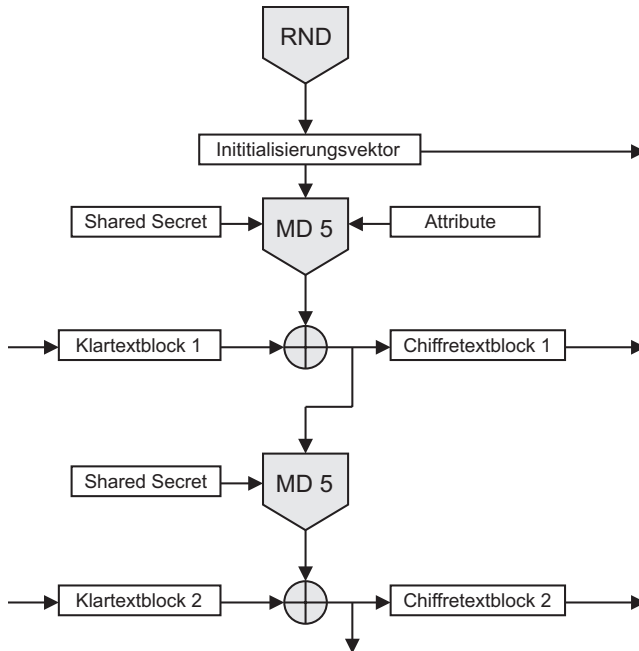


Abbildung 8.11: Die (optionale) Verschlüsselung eines AVP

Eine praktische Anwendung für versteckte beziehungsweise verschlüsselte AVPs ist die Proxy-Authentifizierung. Dabei schickt der LAC die User ID und das Passwort des PAP-Protokolls zum LNS. Da diese Informationen im Klartext vorliegen, sollten sie beim Transport über ein öffentliches Netz wie das Internet verschlüsselt werden.

Die in L2TP spezifizierte Verschlüsselung ist allerdings vom Sicherheitsstandpunkt aus gesehen nicht gerade umwerfend. Der Algorithmus zum Ver- und Entschlüsseln ist eine schlichte Exklusiv-Oder-Verknüpfung des Klartextes mit einem 128 Bit großen Schlüssel. Dieser Schlüssel wird aus einer Zufallszahl, die mit einem CBC-Initialisierungsvektor vergleichbar ist, dem Attribut und einem Shared Secret mit Hilfe der MD5-Funktion berechnet. Dieses Shared Secret wird auch zur Authentifizierung von LNS und LAC beim Aufbau eines L2TP-Tunnels benutzt. Die MD5-Funktion ist als Basis des CHAP-Verfahrens ebenfalls bereits im System vorhanden.

Die Verschlüsselung arbeitet, wie in Abbildung 8.11 zu sehen ist, in einem Output-Feedback-Modus mit einem expliziten Initialisierungsvektor. Aus dem Shared Secret, dem Attributwert und dem Initialisierungsvektor wird über MD5 ein 128 Bit großer Wert erzeugt, der mit den ersten 16 Byte des originalen AVP-Subformats exklusiv-oder-verknüpft wird. Damit sind die ersten 16 Bytes verschlüsselt. Aus diesem Wert und dem Shared Secret wird über MD5 wieder ein 128-Bit-Wert erzeugt, mit dem die nächsten 16 Bytes verschlüsselt werden. Dies wird so lange weitergeführt, bis der originale Wert vollständig verschlüsselt ist.

Aus Sicherheitsgründen wird die Länge des originalen Wertes im Value-Feld verschleiert, indem man ein so genanntes Hidden-AVP-Subformat erzeugt, das aus einem Längenfeld, dem Originalwert und Füllzeichen besteht. Dieses Format wird mit dem beschriebenen Verfahren verschlüsselt und als Wert in das Value-Feld des AVP eingetragen.

An dieser Stelle sei auch bereits darauf hingewiesen, dass die Verschlüsselung nur auf Steuerungspakete angewendet werden kann, nicht jedoch auf L2TP-Datenpakete.

Overall Length Dieses Feld gibt die Gesamtlänge des AVP in Byte an.

Vendor ID Für AVPs, die von der IETF spezifiziert worden sind, ist dieser Wert 0. Herstellerspezifische AVPs enthalten eine Vendor ID, die der von der IANA (Internet Assigned Numbers Authority) zugewiesenen Nummer gemäß dem RFC1700 entspricht. Somit ist die Kombination von Attribut und Vendor ID weltweit eindeutig, denn selbst wenn unterschiedliche Hersteller die gleichen Attributbezeichner verwenden, so unterscheiden sie sich doch durch ihre Vendor ID.

Attribute Dieses Feld enthält den Bezeichner des Attributs als 16-Bit-Wert.

Value In diesem Feld steht der Wert des Attributs. Er kann, innerhalb der maximalen Größe des AVP, beliebig lang sein, und sein Wert wird vom Empfänger anhand des Attribute- und Vendor-ID-Feldes interpretiert.

8.2.7 Auf- und Abbau von Tunneln und Calls in L2TP

Wir betrachten hier einen Tunnelaufbau im Voluntary Tunneling Model, also mit einem virtuellen LAC auf dem Client, wie es auch bei Microsoft L2TP-Clients der Fall ist.

Der LAC versucht, einen Tunnel zum LNS aufzubauen, indem er einen Start-control-connection-request an den LNS schickt. Der LNS antwortet mit einem Start-control-connection-reply, falls er dem Wunsch des LAC, einen Tunnel zu erzeugen, folgt. Sobald der LAC diese Antwort erhält, komplettiert er seinen Tunnelaufbau, und beide Seiten, LAC und LNS, sind über eine Control Connection miteinander verbunden. Im Sinne von L2TP ist damit ein Tunnel aufgebaut.

Innerhalb dieses dreiphasigen Handshake-Verfahrens kann auch die Authentifizierung der beiden Tunnelendpunkte erfolgen. Dazu wird ein dem CHAP-Verfahren analoger Mechanismus auf Basis von MD5 mit Challenge-, Response- und Accept-/Reject-Paketen unter Verwendung eines Shared Secret auf LAC und LNS eingesetzt.

Die LCP-Aushandlungen können jetzt durch den Tunnel zwischen dem Client und dem LNS direkt erfolgen, ohne dass die virtuelle LAC-Funktion noch in LCP involviert ist. Nachdem die Übertragungsparameter und Authentifizierungsprotokolle ausgehandelt wurden, kann die Authentifizierung des Benutzers (Clients) erfolgen. Nun können die Netzwerksteuerungsprotokolle gestartet werden, und nach deren Abschluss können L2TP-Datenpakete und damit auch Netzwerkpakete übertragen werden.

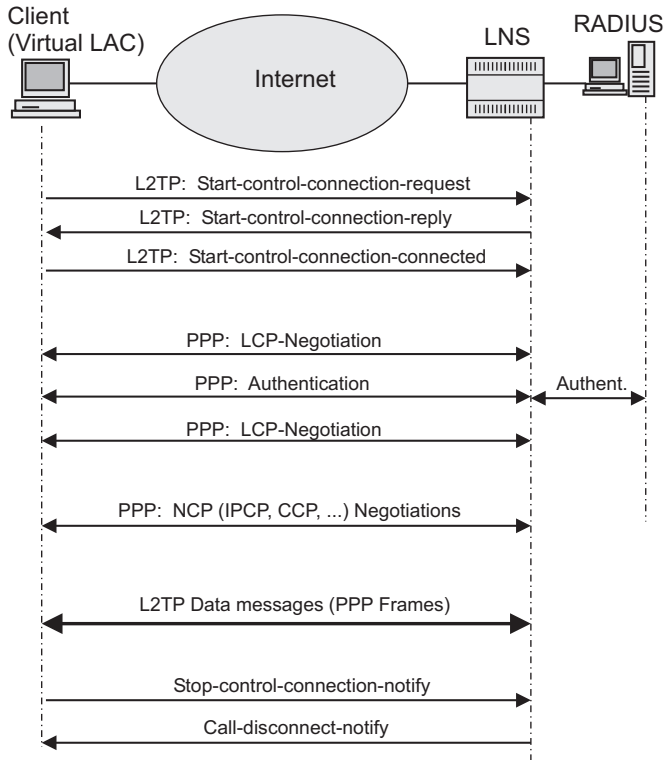


Abbildung 8.12: Der Aufbau einer Verbindung ist ein mehrstufiger Prozess, indem zuerst L2TP und anschließend PPP involviert sind.

Wird die Verbindung vom Client unterbrochen, also der Call terminiert, kann auch der Tunnel selbst abgebaut werden, wenn sonst keine weiteren Calls in diesem Tunnel mehr aktiv sind. Der Abbau wird ebenfalls über entsprechende Steuerungspakete geregelt.

8.2.8 Die Sicherheit von L2TP

Genau genommen müsste die Überschrift zu diesem Unterkapitel eigentlich lauten: Die Unsicherheit von L2TP. Bevor sich Ihnen beim Lesen dieses Abschnitts aber der Gedanke aufdrängt, L2TP sei eigentlich ein völlig unbrauchbares Protokoll zum Einsatz in öffentlichen Netzen, sollten Sie sich vor Augen halten, dass L2TP kein Sicherheitsprotokoll ist, dies niemals sein sollte und dies niemals sein wird. L2TP ist, der Name sagt es eigentlich schon, ein reines Tunneling-Protokoll. Sicherheitsmaßnahmen werden auf anderen Ebenen eingesetzt, auf der Transport- oder der Netzwerkebene (L2Sec, IPSec usw.) oder auf applikationsnahen Ebenen wie SSL, PGP usw.

Aus diesem Grunde sind auch in L2TP nur minimale Sicherheitsmechanismen eingebaut, die sich hauptsächlich auf den Bereich der Control Connection und der Tunnelauthentifizierung beschränken. Im Vergleich zu Sicherheitsprotokollen wie IP-Security bietet L2TP keinen Schutz der Datenvertraulichkeit und der Datenintegrität, und es fin-

det auch keine Paketauthentifizierung statt. Falls hierfür Bedarf vorhanden ist, dann wird vom L2TP-Standardisierungsgremium empfohlen, die Sicherheitsdienste der darunter liegenden Transportprotokolle zu benutzen, also im Falle von L2TP/UDP beispielsweise IPSec im Transport-Modus einzusetzen. Auf diese Weise wird die Verbindung zwischen LNS und LAC vollständig – also Steuerungs- und Datenkanäle – geschützt.

L2TP selbst bietet eine zugegebenermaßen recht simple Authentifizierung von LNS und LAC beim Aufbau eines Tunnels. Diese Authentifizierung basiert auf einem dreiphasigen Handshake-Verfahren, ähnlich wie CHAP, das die Existenz eines Shared Secrets auf den beteiligten Gegenstellen voraussetzt. Dieses Shared Secret ist auch gleichzeitig der Schwachpunkt des Verfahrens, denn es wird in den meisten Implementierungen eine Kombination aus Buchstaben, Ziffern und Sonderzeichen im Klartext gespeichert. Somit besteht die Gefahr von Wörterbuchangriffen, Brute-Force-Attacken oder eines Angriffs auf den LAC oder LNS selbst, da das Shared Secret dort entweder direkt im Klartext gespeichert oder zumindest einfach rekonstruierbar ist.

Als weitere Sicherheitsmaßnahme ist die Verwendung von Hidden-AVPs innerhalb der Steuerungsverbindung möglich, wie bei der Vorstellung des H-Bit erläutert wurde. Diese im Vergleich zu Algorithmen wie DES oder IDEA eher rudimentäre Verschlüsselung ist leider mit einem überschaubaren Aufwand zu „knacken“. Man muss sich lediglich das Shared Secret beschaffen (siehe oben) und kann damit den benutzten 128-Bit-Schlüssel berechnen, oder man startet eine Kryptoanalyse auf das Ergebnis der Exklusiv-oder-Verknüpfung, die in L2TP als Ver- und Entschlüsselungsalgorithmus verwendet wird.

Im Gegensatz beispielsweise zum PPTP hat man beim Design von L2TP darauf verzichtet, bestimmte Dienste oder Funktionen an darunter liegende Protokolle zu binden. So werden Pakete des Steuerungskanals im PPTP aus Gründen der Verlässlichkeit über TCP transportiert, Datenpakete dagegen über UDP. L2TP ist dagegen ein so genanntes „Self Containing Protocol“, also ein Protokoll, das völlig unabhängig von den darunter liegenden Schichten arbeitet. L2TP kann daher auf UDP/IP, Frame Relay, ATM usw. aufsetzen. Die Zuverlässigkeit, die PPTP durch die Verwendung von TCP erreicht, ist im L2TP-Protokoll selbst implementiert. Der Verzicht auf TCP birgt zugleich aber auch einen wichtigen Sicherheitsvorteil. Denn das TCP-Protokoll ist gegen eine Reihe von protokollspezifischen Angriffen anfällig: die so genannten Denial-of-Service-Angriffe. Beim Einsatz von UDP ist man davor viel besser geschützt.

8.3 L2TP-over-IPSec

Falls man L2TP in IP-Umgebungen wie dem Internet oder den Backbones von Service Providern einsetzt, ist IPSec das empfohlene Verfahren, um die notwendige Sicherheit zu gewährleisten. Durch die Entscheidung von Microsoft, dieses Verfahren (IPSec secured L2TP oder kurz L2TP/IPSec-Transport) als Basis seiner VPN-Technologie seit Windows 2000 einzusetzen, ist es mit einem Schlag sehr populär geworden und befreit denjenigen, der es einsetzen mag, auch von der Verteilung der Client-Software anderer Hersteller. Zugleich markiert es auch die Abkehr Microsofts vom jahrelang bevorzugten PPTP und eine Hinwendung zu offenen Standards wie IP-Security und L2TP. Das Verfahren ist im RFC3193 (Securing L2TP using IPSec) beschrieben.

Durch diese Kombination von IPSec und L2TP, und damit auch PPP, beseitigt man entscheidende Schwachstellen in beiden Verfahren. IPSec ist ein sehr sicheres Protokoll, aber von seinem Design her ausschließlich auf Sicherheit ausgelegt – mit der Option, einzig und allein IP zu tunneln. IPX und andere Protokolle bleiben außen vor. Außerdem ist Standard-IPSec ziemlich ungeeignet für Remote Access, da die ganzen Funktionen zur Konfiguration der Verbindung (IP-, DNS-, WINS-Adresszuweisung usw.) fehlen und von vielen Herstellern durch eigene Erweiterungen nachgerüstet werden müssen.

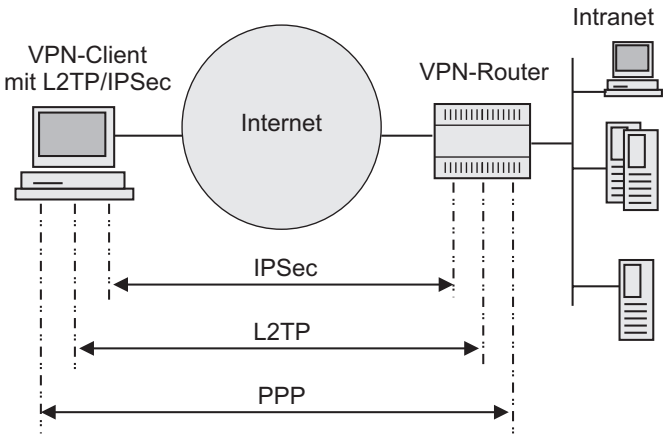


Abbildung 8.13: L2TP/IPSec kombiniert die Stärke von L2TP beim Tunneling und Remote Access mit der Sicherheit von IPSec.

L2TP andererseits kann aufgrund seines Designs alle möglichen Protokolle tunneln, sofern sie in PPP einzukapseln sind. Aber L2TP bietet praktisch keine nennenswerten Sicherheitsfunktionen. Durch die Kombination beider Protokolle heben sich beide Schwächen gegenseitig auf, und man hat ein sehr flexibles Tunneling-Protokoll mit höchster Sicherheit. Durch die Verwendung von PPP in L2TP kommt man auch in den Genuss aller Funktionen, die PPP für den Remote Access bietet.

	PPP Sessions	L2TP-Tunnel	IKE-SA	IPSec-SA

Tabelle 8.1: Die Ressourcen von IPSec und L2TP/IPSec auf einem VPN-Router bei 1000 Verbindungen

Allerdings erkaufte man sich diesen Vorteil zu einem gewissen Preis. Sehen wir uns einmal in Tabelle 8.1 den Vergleich zwischen IPSec und L2TP/IPSec als Basis eines einfachen VPN an, und überschlagen wir den Aufwand (Rechenzeit, Speicher), den die beteiligten Systeme damit haben. Zum Rechnen nehmen wir als praxisnahes Beispiel einmal ein mittelgroßes Unternehmen an, das im Schnitt 1.000 Benutzern gleichzeitig Remote Access gewähren will. Beim Vergleich der Protokolle sieht man Folgendes:

IPSec Auf dem Gateway im Kundennetz werden 2.000 IPSec-SAs und 1000 IKE-SAs terminiert. Da man hier in der Regel den IPSec-Tunnel-Mode benutzt, kommt noch etwas Aufwand durch die Einkapselung hinzu.

L2TP/IPSec Auf dem Gateway werden 1.000 L2TP-Tunnel, 1.000 PPP-Sessions, 2.000 IPSec-SAs und 1000 IKE-SAs terminiert. Da L2TP für das Tunneling zuständig ist, kann IPSec hier im Transport Mode betrieben werden, falls keine Verkehrsfluss-Vertraulichkeit benötigt wird.

8.3.1 Die Performance von L2TP/IPSec

Es ist klar, dass die Terminierung einer größeren Anzahl von L2TP/IPSec-Verbindungen einiges an Ressourcen verschlingt. Selbst spezielle VPN-Hardware benötigt einiges an zusätzlichem Arbeitsspeicher und vor allem Hardware-Beschleuniger, wenn nicht IPSec-Tunnelmode, sondern L2TP/IPSec terminiert werden soll. Die Hardware-Beschleuniger haben vor allem den Sinn, im VPN-Router Ressourcen für die PPP- und L2TP-Terminierung frei zu machen. Den Speicher braucht man, weil PPP und L2TP einiges an Speicher für die Verbindungs- und Tunnelendpunkte benötigen.

Ab einer gewissen Anzahl von Verbindungen benötigt man in jedem Fall spezielle Hardware (VPN-Gateway, VPN-Router) und insbesondere ein Echtzeitbetriebssystem. Wer etwas technisches Verständnis mitbringt, der wird erkennen, dass ab einer gewissen Anzahl von Verbindungen als VPN-Gateway kein normaler Server auf Time-Sharing-Basis, egal ob Windows oder Linux, eingesetzt werden kann – auch nicht mit Kryptokarten. Das ist auch nicht notwendig, denn alle namhaften Hersteller leistungsfähiger VPN-Router sind in der Lage, neben ihren eigenen IPSec-Clients auch Microsoft VPN Clients mit L2TP/IPSec zu terminieren.

8.3.2 Die Erzeugung von L2TP/IPSec-Paketen

Nachdem nun klar ist, worauf man sich einlässt, folgt hier ein kurzer Überblick, wie L2TP/IPSec nun eigentlich funktioniert. Zwischen dem L2TP/IPSec-Client und dem Gateway muss zuerst eine IPSec-Verbindung aufgebaut werden. Anschließend kann über diese sichere Verbindung ein L2TP-Tunnel aufgebaut werden, durch den nun die privaten Netzwerkpakete in PPP-Frames transportiert werden können.

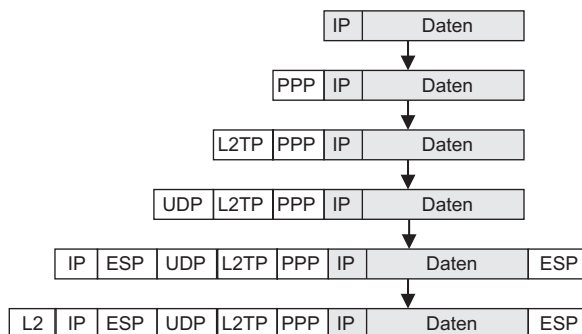


Abbildung 8.14: Das sieht auf den ersten Blick pompös aus, aber der Overhead von L2TP ist nur wenig höher als der von IPSec im Tunnel Mode.

Die in der Abbildung 8.14 dargestellte Paketverarbeitung läuft nun wie folgt ab: Die IP-Pakete werden in PPP-Frames eingekapselt. Diese PPP-Pakete werden anschließend in L2TP-Datenpakete gekapselt und diese wiederum in UDP/IP-Pakete eingepackt. Um diese Pakete herum wird ein IPSec-Header und -Trailer konstruiert, die Daten werden verschlüsselt, und es wird ein Message Authentication Code (MAC) berechnet. Anschließend wird das Paket nochmals auf dem Layer 2 (PPP, PPPoE, Ethernet usw.) eingekapselt und kann nun zum VPN-Router geschickt werden. Dies ist nicht nur eine recht aufwändige Verarbeitung, auch der Paket-Overhead kann sich im Vergleich zur Nutzlast, dem originalen IP-Paket, sehen lassen. Aber wie gesagt, bei Kenntnis des Sachverhalts und einer davon ausgehend vernünftigen Auslegung der Komponenten und ihrer Leistungsfähigkeit hat man die Möglichkeit, in den Genuss der Sicherheit von IPSec zu kommen, trotzdem auch Nicht-IP-Pakete zu übertragen und vor allem die Remote-Access-Funktionen von PPP nutzen zu können.

Im Kapitel zu den Lösungsbeispielen (Kapitel 11) finden Sie eine ausführliche Beschreibung zum Design von Microsoft L2TP/IPSec Remote Access VPN.

9 Quality of Service in VPN

9.1 Quality of Service (QoS)

Insbesondere in konvergenten Netzen, in denen Sprache, Daten und Video über die gleiche Netzwerkinfrastruktur und das gleiche Netzwerkprotokoll übertragen werden, ist das Thema Quality of Service (QoS) extrem wichtiges Kriterium geworden, weil mit der nach wie vor teuren Ressource WAN möglichst effizient umgegangen werden muss.

Viele Carrier und Service Provider haben die unterschiedlichen QoS-Verfahren, wenn überhaupt, bisher nur intern eingesetzt, um ihre Netze selbst effizient nutzen zu können. Nun beginnen diese aber auch ihren Kunden mehr und mehr abgestufte Service Level Agreements (SLA) anzubieten, zunehmend auch mit der Möglichkeit verschiedener Dienstgüteklassen für unterschiedliche Applikationen.

Neben den klassischen Merkmalen wie Verfügbarkeit und Bandbreite werden in diesem Abschnitt insbesondere auch Ansätze wie klassenbasierter QoS und verschiedene Techniken aus dem Bereich Warteschlangen und Flusssteuerung aufgrund ihrer Wichtigkeit etwas eingehender beleuchtet.

In früheren Zeiten existierten in vielen Unternehmen gleich mehrere Netze, oft sogar mit eigener Verkabelungstechnik, als da waren: Infrastrukturen für Telefonie (klassische analoge Zweidrahttechnik), Netze für Hostanwendungen (Koax- oder Twinax-Systeme), PC-Netze (auf Basis von Ethernet, Token Ring oder Arcnet) und Anbindungen an das Internet oder andere öffentliche Netze über das IP-Protokoll. Dieser Tohuwabohu verschiedenster Technologien hatte aber auch einen Vorteil: Die unterschiedlichen Netze waren von ihrer Funktionalität und Qualität optimal auf die speziellen Anwendungen abgestimmt, für die sie eingesetzt wurden.

Die in den letzten Jahren stattgefundene Konsolidierung der Netze hat dazu geführt, dass in den meisten Fällen nur noch zwei Netze übrig blieben, ein Datennetz mit IP als Netzwerkprotokoll auf verschiedensten Infrastrukturen und ein Sprachnetz für die klassische Telefonie, das auch oft für Videokonferenzen genutzt wird. Und diese Applikationen werden in neuester Zeit zunehmend auch auf das Datennetz und IP (Voice over IP, Video over IP) verlagert. Gerade solche Anwendungen reagieren jedoch empfindlich auf große Verzögerungen oder Verzögerungsvarianz (Jitter).

Nun gibt es nicht wenige, die auf dem Standpunkt stehen, man brauche keine komplizierten und komplexen QoS-Mechanismen um die nötigen Qualitätskriterien zu erfüllen, wenn man einfach nur genug Bandbreite zur Verfügung stellt. Das ist auch nicht grundsätzlich falsch, denn wenn nirgendwo Bandbreitenengpässe existieren, dann werden auch alle Datenpakete mit einer vorhersagbaren Geschwindigkeit und Verzögerung übertragen. Aber ist so etwas überhaupt in der realen Netzwerkwelt umsetzbar?

Diese Strategie des Bandbreitenüberangebots wird in den meisten LANs schon länger angewendet. Dort fand in den letzten Jahren eine deutliche Steigerung der mittleren Bandbreiten von 10 Mbit/s Shared Media bis zu 100 MBit/s bis zum Arbeitsplatz statt, in absehbarer Zeit werden es 1 GBit/s sein. Im Backbone-Bereich und zur Serveranbindung wird zunehmend Gigabit-Technologie eingesetzt, und 10 Gigabit Ethernet steht schon vor der Tür.

So weit, so gut – nur was passiert, wenn Netzwerke standortübergreifend sind oder werden? Kann man seine Verbindung zwischen München und Frankfurt genau so wie im LAN-Bereich auslegen – also mit Multilink-Trunking und mehreren 1.000-MBit/s-Fasern zwischen den verschiedenen Ethernet-Switchen in den Standorten? Theoretisch ja, denn die Fasern sind vorhanden, und deren Besitzer, die Carrier, würden sich ehrlich freuen, allerdings wären die Kosten weit jenseits dessen, was sich die meisten Unternehmen, egal welcher Größe, erlauben können oder wollen. Also heißt die Realität im WAN: geringere Bandbreite zu tragbaren Kosten – und eben mögliche, in den meisten Fällen auch sehr wahrscheinliche, Überlastsituationen in Kauf zu nehmen.

Nun kommt hier natürlich die Frage auf, welche Applikationen denn besonders empfindlich auf die Folgen von Überlast wie Paketverluste, Verzögerungen und niedrige Bandbreiten reagieren und welche nicht? Für Erstere müssen nämlich geeignete Maßnahmen getroffen werden, um bestimmte Qualitätskriterien für eine Verbindung zwischen zwei Endgeräten in einem Unternehmensnetz, bestehend aus LAN und WAN, zu erfüllen.

Verkehrskategorie	Beispielanwendungen

Tabelle 9.1: Die verschiedenen Verkehrskategorien und typische Anwendungen daraus

Üblicherweise unterscheidet man aufgrund der damit verbundenen Applikationen folgende Kategorien von Netzwerkverkehr, die sowohl subjektives Empfinden von Endbenutzern als auch technologische Anforderungen von Anwendungen berücksichtigen:

Netzwerksteuerung (Network Control). Das Hauptmerkmal von kritischen Applikationen dieser Kategorie ist, dass sie Priorität vor allen anderen Anwendungen haben, da sie für die Steuerung des Netzwerks unabdingbar sind. Protokolle, die zum Aufbau von Hochverfügbarkeitsumgebungen benutzt werden, sowie Routing- oder Signalisierungsprotokolle fallen in diese Sparte.

Interaktiv (Interactive). In diese Kategorie fallen Anwendungen, wenn zwei oder mehr Personen aktiv mit Hilfe dieser Applikationen auf irgendeine Weise kommunizieren oder zusammenarbeiten. Telefonie oder Videokonferenzen sind typische Beispiele hierfür. Qualitätskriterien für eine derartige Kommunikation zwischen den Beteiligten sind Echtzeitfähigkeit, also minimale Verzögerung und minimale Laufzeitunterschiede sowie eine Mindestqualität hinsichtlich Bandbreite und Informationsverlust (Packet Loss).

Bedarfsgesteuert (Responsive). Falls die Anforderungen an Verzögerung, Laufzeitunterschiede und Informationsverlust nicht ganz so hoch sind, spricht man von bedarfsgesteuertem Netzwerkverkehr. Hier fehlt die interaktive Komponente, in der Regel handelt es sich um Datentransfers, die vom Benutzer von einem System angefordert und durch den Anwender gesteuert werden können, z.B. Video-Streaming, Musikwiedergabe über Netze usw.

Zeitgerecht (Timely). Falls die zeitnahe Reaktionsfähigkeit gar kein Kriterium mehr ist und auch Laufzeitunterschiede oder Paketverluste in Kauf genommen oder durch Re-Transmits (wiederholtes Senden verloren gegangener Pakete) ausgeglichen werden können, spricht man von zeitgerechtem Netzwerkverkehr. E-Mail, Datenbankabfragen oder Dateiübertragungen sind die typischen Anwendungen, hier kommt es auf ein paar Sekunden oder Minuten nicht so sehr an. Lediglich Paketverluste dürfen hier nicht vorkommen, was aber durch wiederholtes Senden verloren gegangener Pakete, wenn auch mit kleinen Verzögerungen, keinerlei Probleme bereitet.

Interessanterweise werden aber auch gerade im LAN-Bereich, zumindest den meisten neueren Ausschreibungen zufolge, gezielt QoS-Funktionen in den Switchen gefordert, trotz der vorhandenen, enormen Kapazität dieser Systeme. Das ist nicht zuletzt eine Folge davon, dass die Systeme, welche die Netzwerklast erzeugen, auch nicht auf dem Stand von 1990 geblieben sind. Die Rechner haben eben keine 486er-CPU mit einem 10-MBit/s-Halb-Duplex-Interface, sondern einen Pentium 4 mit einem 100 oder 1000-MBit/s-Full-Duplex-Ethernet-Interface. Und entsprechend hoch ist auch die Zugriffsrate auf das Netz geworden.

Ebenso ist die Zahl der Netzwerkanwendungen sehr schnell gewachsen, so dass selbst in heutigen, schnellen Netzen durchaus an einigen Punkten eine Überlast auftreten kann – selbst bei Infrastrukturen mit 100 MBit/s bis zum Arbeitsplatz und einem Multi-Gigabit Backbone.

Aber das LAN ist in der Regel nur ein kleiner Teil des Übertragungswegs, den Pakete innerhalb einer VPN-Verbindung nehmen. Und eine Kette ist nur so stark wie ihr schwächstes Glied. Und die schwächsten Glieder in der Netzwerkkette sind von der Geschwindigkeit her fast immer die WAN-Verbindungen.

In Folge dieser Entwicklung, Konvergenz und Konsolidierung der Netze auf der einen Seite und den nach wie vor starken Einschränkungen hinsichtlich WAN-Bandbreiten und nach wie vor möglichen Engpässen im LAN auf der anderen Seite, haben sich im Laufe der letzten Jahre verschiedene QoS-Verfahren für die unterschiedlichen LAN- und WAN-Technologien entwickelt:

- ATM Service Categories

- LAN (Ethernet) IEEE 802.3p

- PPP Class Numbers

- Frame Relay Discard Eligibility (DE)

- Resource Reservation Protocol (RSVP), Integrated Services

- IP Differentiated Services (DiffServ)

- MPLS EXP-Bits und MPLS Traffic Engineering

- UMTS Wireless Traffic Classes (Conversational, Streaming, Interactive, Background)

- IEEE802.11e (Wireless LAN Quality of Service)

Netzwerkweiter Quality of Service erfordert nun, dass diese verschiedenen Technologien und Verfahren miteinander kombiniert werden, denn heutige Netze sind ein Mix aus verschiedensten Infrastrukturen, die nur eins gemeinsam haben: Sie transportieren fast ausschließlich nur noch IP-Pakete.

In diesem Buch und ganz besonders in diesem Kapitel wird sehr oft und teilweise sehr detailliert auf das Thema Voice over IP (VoIP) und anderen echtzeitähnlichen Verkehr eingegangen. Das hat seinen Grund darin, dass in den nächsten Jahren sehr viele Unternehmen in unterschiedlichem Umfang VoIP einsetzen werden. Und die Netze müssen den Qualitätsanforderungen interaktiver Sprachkommunikation gewachsen sein, und zwar in ihrer Gesamtheit. Denn bei einer Ende-zu-Ende-Kommunikation entscheidet das schwächste Glied über deren Qualität.

9.2 Qualitätskriterien

Beim Thema Quality of Service ist es sehr wichtig zu wissen, welche Parameter eigentlich die Güte eines Dienstes bestimmen. Die wichtigsten Kriterien zur Bestimmung der Dienstgütequalität sind:

- Verfügbarkeit

- Verzögerung

- Verzögerungsvarianz

- Fehlerrate

- Bandbreite

9.2.1 Verfügbarkeit

Die Verfügbarkeit kennzeichnet, meist prozentual angegeben, wie lange eine Netzwerkressource, ein Dienst oder eine Leistung zur Verfügung steht. Aber Vorsicht, beim Thema Verfügbarkeit scheiden sich gelegentlich die Geister daran, was alles von diesem Begriff umfasst wird. Provider schmücken sich gerne mit Netzverfügbarkeiten (gemeint ist damit meist der Backbone) von 99,999%, jedoch interessiert den Kunden vielmehr die Ende-zu-Ende-Verfügbarkeit – und da wird dann teilweise drastisch nach unten korrigiert. Letztere umfasst nämlich die Verfügbarkeit von allem (Backbone, Zugangsleitung, Modems, Router usw.), was jenseits der Verantwortung und des Einflussbereichs des Kunden liegt – und das ist der für den Endkunden wirklich interessante Wert.

Der sarkastisch gemeinte Satz, wenn ein Anbieter eine Verfügbarkeit von fünf mal neun verspreche, müsse man schon nachfragen, wo genau das Komma stehe, ist leider gar nicht so unrealistisch. Mir sind zwar keine so schwarzen Schafe (9,9999% Verfügbarkeit) bekannt, aber bereits der Unterschied zwischen 99,999% zu 99,0% kann schon unangenehm spürbare Folgen für den Endanwender haben, je nach Struktur des Unternehmens.

Nehmen wir ein Beispiel mit dem schlimmsten Fall, dass die möglichen Ausfälle bei 99,9% und 99,0% Verfügbarkeit tatsächlich eintreten und sich innerhalb der Arbeitszeiten bewegen:

Ein Unternehmen, das nur national tätig ist, mit fünf Arbeitstagen pro Woche und einer Tagesarbeitszeit, in der Netzverfügbarkeit benötigt wird, von 07:00 bis 19:00 (12 Stunden), würde mit einer maximalen Ausfallzeit von 87,6 Stunden (99,0%) gegenüber 8,76 Stunden (99,9%) belastet. Da sich die möglichen Ausfälle statistisch gesehen jedoch auch über Sonn- und Feiertage sowie Zeiten, in denen nicht gearbeitet wird, erstrecken, sieht eine realistische Abschätzung so aus, dass das WAN bei 99,0% maximal 26 Stunden und bei 99,9% maximal 2,6 Stunden nicht verfügbar ist.

Beim Aushandeln von Verfügbarkeiten sollte man sich unbedingt im Klaren sein, was genau damit gemeint ist, und somit sollten keine Missverständnisse auftreten, die später zu einigem Ärger führen können.

		(220 Arb.-Tage mit jeweils 12 Std.)

Tabelle 9.2: Eine Übersicht über die maximale und durchschnittliche Anzahl möglicher Ausfallstunden bei verschiedenen Mindestverfügbarkeiten

Jedoch sollte man beachten, dass eine bestimmte Verfügbarkeit, die man in einem Service Level Agreement vereinbart, nicht bedeutet, dass das Netz tatsächlich so lange oder so häufig ausfällt, sondern dies ist eine vom Anbieter garantierte Mindestverfügbarkeit, die nicht selten deutlich besser ist. Die vereinbarten Werte sind schlicht und einfach der schlimmste anzunehmende Fall.

Netzverfügbarkeit kann mit einigen Zwischenstufen zweierlei bedeuten: Die Verfügbarkeit Backbone eines Providernetzes oder das WAN als (aus Sicht des Kunden) Ganzes, vor allem inklusive der Zugangsleitungen von den Kundenstandorten. Bei Kunde und Provider kann da die Sicht der Dinge leider etwas divergieren. Falls zum Beispiel in einem hierarchisch aufgebauten WAN, mit einem zentralen Standort, auf dem zentral alle geschäftskritischen Anwendungen betrieben werden, gerade die Verbindung zu diesem Standortes ausfällt, jedoch alle 20 anderen Standorte noch eine Verbindung zum Provider haben und dessen Backbone ebenfalls noch funktioniert, kann sich die Sache je nach unterschiedlicher, zugegebenermaßen etwas überspitzt gezeichneter, Sicht so darstellen:

- Kunde: Das Netzwerk in seiner Gesamtheit kann nicht benutzt werden, ist also aus seiner Sicht nicht verfügbar, obwohl 20 von 21 Verbindungen nicht ausgefallen sind.
- Provider: Es liegt überhaupt kein Netzausfall als solcher vor, da der Backbone und alle Zugangsleitungen außer einer einzigen verfügbar sind.

Damit dies, wenn dieser Fall überhaupt eintreten sollte, zu keinem ernsten Problem wird, sollte man sich über diese Gegebenheiten im Klaren sein, dies mit dem Provider diskutieren und verhandeln oder seine Netzwerk-Infrastruktur einfach so auslegen, dass

solche Probleme entschärft oder gar ganz vermieden werden können – und zwar möglichst ohne Zutun von Provider oder Carrier durch ein geschicktes Netzwerkdesign. Denn selbst entsprechende Verträge sind problematisch, weil die Vertragsstrafen in vorgefertigten Standard-SLAs meist so niedrig sind, dass sie in keinem angemessenen Verhältnis zu dem tatsächlichen Schadenspotenzial beim Kunden stehen. Und spezielle Regelungen mit höheren Vertragsstrafen lässt sich so leicht kein Provider abringen, ohne dass mindestens ein millionenschwerer Auftrag dahinter steht.

9.2.2 Verzögerung

Die Übertragung von Datenpaketen bringt technologiebedingt eine Verzögerung (Delay) mit sich, die um so größer wird, je länger die Strecke ist und je mehr Vermittlungssysteme oder Signalregeneratoren auf der Übertragungsstrecke durchlaufen werden. Manche Applikationen, wie z.B. E-Mails oder Downloads, reagieren unkritisch auch auf längere Verzögerungen bis in den Bereich mehrerer Sekunden, andere Applikationen wie Sprach- oder Echtzeit-Videoübertragungen sind damit jedoch überhaupt nicht mehr möglich. Je geringer die Verzögerung, desto höher ist die Übertragungsqualität.

Bei einem Telefongespräch in digitalen Leitungsnetzen (ISDN) wird beim Verbindungsaufbau diesem Gespräch ein fester virtueller Kanal (z.B. 64 KBit/s bei ISDN) für die Dauer des Gesprächs zugewiesen. Die Daten werden mit einer minimalen Verzögerung und vor allem ohne Verzögerungsvarianz zwischen beiden Endgeräten ausgetauscht. Ist die benötigte Bandbreite im Telefonnetz nicht verfügbar, dann findet das Gespräch eben nicht statt.

In Datennetzen sieht der Fall etwas anders aus. Bis auf ganz wenige Ausnahmen, auf die wir im Weiteren noch eingehen werden, findet keine Reservierung eines Datenkanals in der Art statt, dass auch dem gesamten Weg zwischen Sender und Empfänger in den Vermittlungssystemen Ressourcen, zugewiesen werden. Die Daten werden zum Empfänger geschickt und je nach Protokoll von diesem bestätigt oder im Fehlerfall einfach noch einmal verschickt. Nehmen wir als Beispiel eine Standortverbindung. Hier senden alle Applikationen in den beiden Standorten unabhängig voneinander Pakete hin und her – ob die benötigte Bandbreite nun vorhanden ist oder nicht. Falls sie nicht vorhanden wird, dann werden Datenpakete zwischengespeichert oder verworfen und nach einiger Zeit nochmals angefordert und wieder zu übertragen versucht.

Man kann zwischen verschiedene Arten von Verzögerung unterscheiden:

- Verzögerung durch Schnittstellenwandlung (Serialization Delay)
- Verzögerung durch Signallaufzeiten (Propagation Delay)
- Verzögerung durch Routing/Switching (Processing Delay)
- Verzögerung durch Überlast (Congestion Delay)

So summieren sich die meist geringen Verzögerungen durch die Signallaufzeiten mit den Verzögerungen durch Bandbreitenengpässe zu teilweise erheblichen Gesamtverzögerungen im Hundert- oder Tausend-Millisekunden-Bereich. Und das ist für bestimmte interaktive Anwendungen schon viel zu viel!

Serialization Delay

Diese Verzögerung hängt ausschließlich von der Paketgröße und der Geschwindigkeit der seriellen Schnittstelle ab. Da kann weder etwas mit Arbeitsspeicher, Prozessorleistung oder Warteschlangen verbessert werden.

	64 KBit/s	128 KBit/s	256 KBit/s	512 KBit/s	1,536 MBit/s	2,048 MBit/s

Tabelle 9.3: Das Serialization Delay in Abhängigkeit von Paketgröße und Schnittstellengeschwindigkeit.

In Tabelle 9.3 sind einige Werte für gängige Schnittstellengeschwindigkeiten und Paketgrößen angegeben. Man darf aber nicht den Fehler machen zu sagen, dass ein VoIP-Paket mit 200 Byte auf einer 128-KBit/s-Schnittstelle (z.B. auf einem ADSL Uplink) nur etwa 16 ms verzögert wird. Das ist die minimale Verzögerungszeit, die nur unter idealen Bedingungen erreicht wird. Falls zum Beispiel nur einen Sekundenbruchteil vor dem VoIP-Paket ein FTP-Paket mit 1500 Byte ausgegeben wird, dann wird dieses zuerst fertig übertragen, und dann kommt das VoIP-Paket an die Reihe. Die Verzögerung in diesem Fall wäre also schon 110 ms! Derartige Verhältnisse erzeugen übrigens auch Jitter.

Die einzige Möglichkeit, dagegen etwas zu unternehmen, ist es, alle Pakete ab einer gewissen Größe auf ein kleineres Maß, zum Beispiel auf die Größe von VoIP-Paketen, zu verkleinern (Fragmentierung), um diesen schneller die Schnittstelle zur Verfügung stellen zu können.

Propagation Delay

Propagation Delay ist eine Größe, die überhaupt nicht zu verändern ist. Sie wird bestimmt durch physikalische Konstanten, wie z.B. die Lichtgeschwindigkeit. So werden Daten über einen interkontinentalen Lichtwellenleiter mit 5000 km Länge um 16 ms verzögert. Dieser Wert ist natürlich für sich alleine keiner, aber eben nicht null und damit eine von vielen kleinen Verzögerungen, die sich in Summe zu einer ansehnlichen Gesamtverzögerung summieren können.

Processing Delay

Speziell in VPN-Gateways, Routern, Switchen und Firewalls entstehen Verzögerungen durch die teils exzessive Verarbeitung von Paketen oder Datenframes. Faktoren, die Verzögerungen erzeugen können, sind unter anderem:

- Paketfilterung
- Modifikation von Paketen in Gateways
- Ver- und Entschlüsselung

Forwarding-Prozesse

Warteschlangen (Queues)

In solchen Systemen gibt es glücklicherweise eine Reihe von Maßnahmen, um die Gesamtverzögerung zu verringern. So sollten die entsprechenden Geräte ausreichend leistungsfähig sein, also neben entsprechend schnellen Prozessoren und Chipsätzen soweit möglich ASICs (Application Specific Integrated Circuit, speziell für bestimmte Anwendungen entwickelte Chips) für bestimmte Funktionen einsetzen. Auch sollte die Konfiguration der Systeme entsprechend ausgelegt sein und insbesondere unnötige Funktionen vermeiden.

Congestion Delay

Diese Verzögerungen entstehen, wenn in einem Übertragungssystem eine Überlastsituation existiert. Dies kann ein Erreichen der maximalen Systemkapazität sein oder, der häufigere Fall, wenn auf einem Interface mehr Pakete oder Frames ausgegeben werden sollen, als die maximale Schnittstellengeschwindigkeit zulässt. Letztes ist typisch für Systeme mit mehr als zwei Schnittstellen gleicher Geschwindigkeit oder Systeme mit Schnittstellen unterschiedlicher Geschwindigkeit. Ein ADSL-Modem ist ein Beispiel für letzteren Fall: Auf seinem Ethernet-Interface sind maximal 10 MBit/s möglich, auf dem DSL-Uplink aber zum Beispiel nur 128 KBit/s. Wenn Datenpakete mit insgesamt mehreren MBit/s auf dem Ethernet-Interface ankommen, aber nur mit 128 KBit/s auf der DSL-Leitung gesendet werden kann, dann entsteht eine Überlast, die zu erheblichen Verzögerungen durch massive Paketverluste und zu wiederholtem Senden führt.

In der Praxis ist der typische Datenverkehr aber kein konstanter Paketstrom, sondern eher ein stark unregelmäßiges Muster (Bursty Traffic). Die Überlastsituationen sind dann ebenfalls auch statistisch verteilt und von sehr kurzer Dauer. Die meisten Systeme behelfen sich durch Warteschlangen (Queues), in denen Pakete oder Frames, die gerade nicht übertragen werden können, zwischengespeichert werden, bis diese zu Zeiten geringerer Last auf der Schnittstelle ausgegeben werden können. Damit werden zwar Paketverluste und deren Folgen vermieden, aber es kommt durch die Zwischenspeicherung trotzdem zu Verzögerungen und Jitter.

9.2.3 Verzögerungsvarianz (Jitter)

Was für manche Applikationen noch unverträglicher als eine große Verzögerung ist, sind Variationen der Verzögerungszeiten, in der Fachterminologie auch als Jitter bezeichnet. Gerade für echtzeitähnliche Anwendungen wie VoIP bedeutet Jitter eine starke Reduktion der Sprachqualität.

Jitter kann verschiedene Ursachen haben, eine haben wir bereits besprochen, nämlich Serialization Delay. In besagtem Beispiel hatten wir den Fall, dass entweder ein Sprachpaket sofort übertragen werden kann (16 ms Verzögerung) oder dass man im schlimmsten Fall die Übertragung eines 1500-Byte-Pakets (110 ms Delay) abwarten muss. Die Verzögerungsvarianz beträgt in diesem Fall 94 ms, ein Wert, der hart an der Gesamtgrenze für gute Sprachqualität (100 ms laut ITU-T Y.1541) liegt.

Andere Ursachen sind zum Beispiel unterschiedliche Wege von IP-Paketen durch IP-Netze. So etwas kann z.B. bei parallelen Links oder dem Einsatz von falsch konfigurier-tem Equal Cost Multipath (ECMP) Routing leicht vorkommen.

9.2.4 Fehllerate, Bitfehllerate

Die Fehllerate, oft auch als Bitfehllerate spezifiziert, gibt das Verhltnis von gesendeten zu fehlerhaft bertragenen oder verlorenen Informationen an. Fehlerhafte Datenpakete sind bei der Datenbertragung ebenfalls als verloren anzusehen, da sie sptestens bei der Fehlerkorrektur hoherer Protokolle erkannt und eliminiert werden. Es gibt aber nicht wenige Applikationen, wie z.B. Sprache oder Daten, in denen eine gewisse Fehllerate tolerabel ist. Sie wird entweder vom Anwender subjektiv gar nicht bemerkt oder als nicht strend empfunden. Ein leichtes Knacken im Telefonhrer oder ein schwarzes Bild-punktchen fur 0,02 Sekunden sind durchaus tolerabel, andererseits kann aber ein einzi-ges fehlerhaftes Bit in einer mehreren Megabyte groen Excel-Datei fatale Folgen haben.

Anwendung	Verzgerung	Jitter	Fehllerate

Tabelle 9.4: Anforderung an Verzgerung, Jitter und Fehllerate durch verschiedene Applikationen

9.2.5 Bandbreite

Die Bandbreite ist ein Kriterium, das sich eigentlich nur dann auf die Qualitt von Netz-werkdiensten auswirkt, wenn sie nicht in ausreichendem Mae da ist. Mit anderen Wor-ten, wenn die Ende-zu-Ende-Bandbreite fur alle Verbindungen zwischen den betreffen-den Endpunkten hoher ist als deren summierte, maximal mogliche Bandbreiten, dann hat eine weitere Erhohung der Bandbreite absolut keinen Effekt auf die Dienstgute.

Wenn nicht, dann werden aufgrund von berlasten die restlichen Qualittsparameter aktiv. Es gibt Verzgerungen, Verzgerungsvarianz und Paketverluste. Dies ist aber lei-der der Normalfall, denn sptestens beim bergang vom LAN zum WAN hat man das Problem, dass die WAN-Verbindung in der Regel deutlich geringere Datenraten auf-weist.

64 Byte	80 Byte	160 Byte	256 Byte	512 Byte	1 KByte	1,5 KByte

Tabelle 9.5: Das Verhltnis von Bandbreite auf IP-Ebene zur Bandbreite von Ethernet

Die Kalkulation von Bandbreiten für bestimmte Dienste ist nicht trivial, ja genau genommen unmöglich, da man in der Regel den Netzwerkverkehr, insbesondere bei Datendiensten, nicht vorhersagen kann. Bei Sprachverbindungen kann man zumindest die Bandbreite und die Paketgrößen vorherbestimmen. Bei der Berechnung der theoretisch möglichen Bandbreiten muss man jeweils den Overhead der Dateneinkapselung durch Ethernet, PPP, IP, IPSec, TCP, UDP, RTP usw. berücksichtigen.

9.3 QoS-Technologien

9.3.1 TCP-Flusssteuerung

Dass es, vor allem in Weitverkehrsnetzen mit ihren im Vergleich zu lokalen Netzen niedrigen Übertragungsraten, zu Überlastsituationen (Congestion) kommen kann, ist seit langem bekannt. Aus diesem Grunde wurden schon in den Anfangszeiten von IP Session-Protokolle wie TCP (RFC 793) entwickelt, um in Verbindung mit dem paketorientierten IP-Protokoll eine Ende-zu-Ende-Flusssteuerung zu realisieren. Das Verhalten dieser Flusssteuerung ist auch noch heute oder vielleicht gerade heute ein wichtiges Element moderner QoS-Verfahren zur Übertragung von Daten, die keinen Echtzeitbedingungen genügen müssen.

In Vermittlungssystemen wie z.B. IP-Routern kommt es zu Überlastsituationen, wenn die Summe der vom Gerät empfangenen Daten eine bestimmte Zeit lang größer als die vom System gesendeten Daten ist. Dann sind irgendwann die Zwischenspeicher für die Datenpakete voll, und es können keine weiteren mehr empfangen werden. Weitere ankommende Pakete werden ignoriert – also verworfen; egal welcher Art die sendende Applikation ist.

Eine Möglichkeit der Abhilfe ist es, die Bandbreite der Verbindungsleitungen zu erhöhen. Allerdings wird hier das prinzipielle Problem nicht gelöst, sondern nur verschoben. Vor allem erhöht sich nicht nur die Bandbreite in lokalen Netzen, sondern auch die Geschwindigkeit der Endsysteme und deren Netzwerkadapter, wodurch das Problem exakt das gleiche ist, nur jetzt mit zehn- oder hundertfach höherer Geschwindigkeit. Auch das Vergrößern der Zwischenspeicher (Puffer) macht nur bis zu einer gewissen Größe Sinn, ab dann werden unter Umständen Pakete so lange zwischengespeichert, bis deren Lebenszeit abgelaufen ist, und letztendlich wird, wenn diese dann endlich übertragen werden, eine nutzlose Zusatzlast durch ungültige Pakete erzeugt. Außerdem sind diese beiden Ansätze mit nicht unerheblichen Kosten verbunden.

TCP hingegen realisiert einfach den Ansatz, mit der gerade vorhandenen Lastsituation am effizientesten umzugehen. Als Session-Protokoll auf der TCP/IP-Schicht 4 steuert es die Verbindung zwischen zwei Applikationen. Es garantiert die zuverlässige Zustellung der zu übertragenden Daten in der richtigen Reihenfolge. Zu diesem Zweck werden eingehende Pakete bestätigt, und der Empfänger informiert den Sender dabei auch über seine Möglichkeiten, wie viele Daten er gerade empfangen kann. Diesen Wert bezeichnet man als TCP-Fenstergröße (TCP Window Size). Der Sender, der die Flusssteuerung durchführt, benutzt hierzu die folgenden Variablen und Funktionen:

Einen Timer, der dann ein Signal auslöst, wenn ein gesendetes Paket nach einer bestimmten Zeit nicht bestätigt wurde.

Das Empfangsfenster (Window Size)

Das Überlastfenster (Congestion Window)

Der Schwellenwert (Threshold)

Abbildung 9.1 illustriert die Funktionsweise der Flusssteuerung im TCP-Protokoll. Die Variablen werden auf bestimmte Anfangswerte voreingestellt, in unserem Beispiel ist das Empfangsfenster 32 KByte, das Überlastfenster 1 KByte und der Schwellenwert 32 KByte groß. Nachdem eine TCP-Verbindung aufgebaut wurde, beginnt die Übertragung der Pakete erst einmal mit Maßen. Denn der TCP-Slow-Start-Algorithmus fängt zuerst mit einer kleinen Paketgröße an, die, falls die Pakete bestätigt werden, exponentiell bis zum Wert des Empfangsfensters anwächst.

Ohne Überlast kann nun mit diesen Werten weiter übertragen werden, es sei denn, der Empfänger verkleinert die Fenstergröße. Bei Überlast wird nun z.B. Paket 9 nicht bestätigt, und der Timer löst aus. Wenn dies geschieht, wird der Threshold-Wert auf die Hälfte seines vorherigen Wertes reduziert und der Slow-Start-Algorithmus von neuem gestartet – allerdings nur bis zum Erreichen des neuen Thresholds (16 KByte). Ab hier erfolgt nun keine exponentielle Vergrößerung der übertragenen Pakete mehr, sondern es wird sich linear bis zur Fenstergröße vorgetastet. Somit passt sich die Datenrate moderat an die aktuelle Lastsituation an.

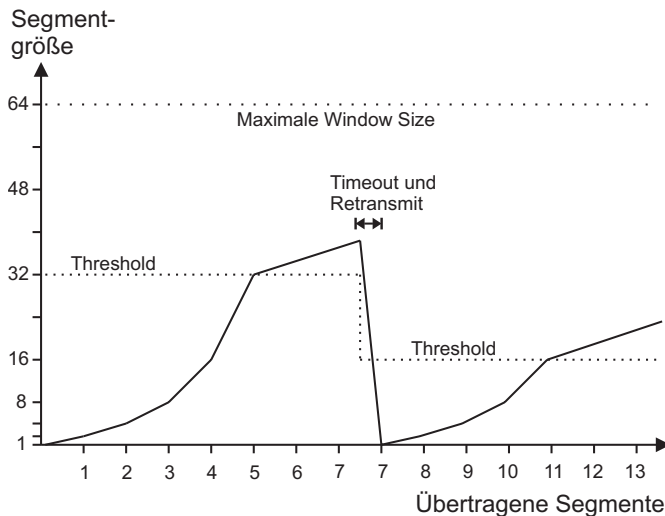


Abbildung 9.1: Die TCP-Flusssteuerung ermöglicht eine Anpassung der Datenrate an Überlastsituationen.

In diese Ende-zu-Ende-Flusssteuerung können nun intelligente VPN-Router oder Gateways eingreifen, die entlang der Übertragungsstrecke zwischen Sender und Empfänger arbeiten. Deren wichtigstes Ziel (neben Routing und Sicherheit) ist es, eine Überlastsituation möglichst im Vorfeld zu vermeiden. Falls solche Geräte erkennen, dass sie auf eine Überlast zusteuern, werden einfach bestimmte Pakete verworfen (s. u., RED). TCP nimmt nun fälschlicherweise (aber so gewollt) an, dass ein Paket aufgrund einer Über-

last nicht beim Empfänger angekommen ist, und beginnt mit den oben beschriebenen Maßnahmen, die zuerst einmal in eine niedrigere Datenrate münden. Damit kann eine Überlast vermieden werden, zumindest meistens. Denn nicht alle Datenpakete sind TCP-Pakete, es gibt noch eine Reihe anderer IP-Protokolle (z.B. UDP), die solche Mechanismen nicht implementiert haben.

Ein weiteres Verfahren, der so genannte Nagle-Algorithmus, hat in fast jede TCP-Implementierung Einzug gehalten, um vor allen kleine Datenmengen effizient zu übertragen. Vereinfacht gesagt sammelt der Algorithmus Applikationsdaten einer Session so lange, bis entweder die maximale Anzahl Bytes für ein IP-Paket erreicht ist oder bis innerhalb der existierenden Session ein Paket von der Gegenseite (meist ein ausstehendes TCP-Acknowledge) eingetroffen ist. Einzig wenn der Zustand der Session „Idle“ ist, werden die Daten, die eine Applikation auf ein TCP-Socket schreibt, sofort gesendet.

An dieser Stelle vielleicht zwei wichtige Anmerkungen:

1. TCP geht davon aus, dass eine hohe Übertragungsqualität gegeben ist und ein Paketverlust daher nur in Überlastsituationen auftritt. Das ist auch bei den meisten kabelgebundenen und optischen Übertragungstechnologien der Fall; jedoch nicht bei drahtloser Übertragung (z.B. GSM, GPRS, UMTS), denn hier gehen gelegentlich Pakete verloren, ohne dass eine Überlastsituation vorliegt. TCP reagiert darauf aus einer Sicht korrekt mit einer Herunterregelung der Datenrate, was in Summe zu einem Durchsatz führen kann, der sehr deutlich unter dem technisch Möglichen liegt! Man mag jetzt einwenden, dass für bestimmte Applikationen aus diesen und anderen Gründen UDP eingesetzt wird. Allerdings wird zum Beispiel bei einem SSL-VPN alles, egal ob die Anwendungen TCP oder UDP benutzen, über TCP übertragen.
2. Es mag vielleicht der Eindruck entstanden sein, TCP würde immer dann eingesetzt, wenn kritische Qualitätskriterien eingehalten werden müssen. Dies trifft aber nur in bestimmten Fällen zu. Insbesondere echtzeitähnlicher Datenverkehr wird hingegen meist über UDP übertragen. Applikationen wie interaktive Sprach- oder Videoübertragung benutzen fast immer die Kombination von RTP (Real Time Protocol) und UDP. Hier müssen kleine Pakete möglichst verzögerungsfrei, ohne Fragmentierung oder Pufferung (größere Datenblöcke) und vor allem ohne wiederholtes Senden verloren gegangener Pakete übertragen werden. TCP tut von all dem genau das Gegenteil!

9.3.2 Weighted Fair Queueing (WFQ)

Weighted Fair Queueing (WFQ) und Random Early Discard (RED) sind zwei Verfahren, die in vielen modernen Übertragungssystemen als QoS-Algorithmen ihren Dienst verrichten. Während WFQ versucht, mit einer bestehenden Überlastsituation fertig zu werden, benutzt man RED dazu, diese möglichst bereits im Vorfeld zu vermeiden.

Von einer Überlastsituation spricht man, wenn einem Vermittlungssystem (Router, Switch, VPN-System usw.) mehr Informationseinheiten zugeführt werden, als es wieder abgeben kann. Typische Fälle sind zum Beispiel Router mit einem 2-MBit-WAN-Interface und einem 10/100-MBit-LAN-Interface: Hier können über das LAN-Interface weit aus mehr Daten in den Router gelangen, als dieser über die WAN-Schnittstelle weiterleiten kann. Für eine gewisse Zeit, allerdings auf Kosten einer stark steigenden Paketverzögerung, können Paketverluste durch die Verwendung von Warteschlangen

vermieden werden. Wenn die Warteschlangen jedoch durch fortgesetzte Überlast irgendwann keine Pakete mehr aufnehmen können, werden alle weiteren eingehenden Pakete verworfen – unabhängig von ihrer Wichtigkeit. Oft reicht bei dem typischerweise sehr sprunghaften Datenverkehr (Bursty Traffic) eine normale Warteschlangenfunktion (FIFO, First In First Out) aus, jedoch nicht immer. Denn mittlerweile ist die Diskrepanz im Verhältnis zwischen LAN- und WAN-Bandbreiten immer größer geworden, so dass intelligentere Verfahren benötigt werden.

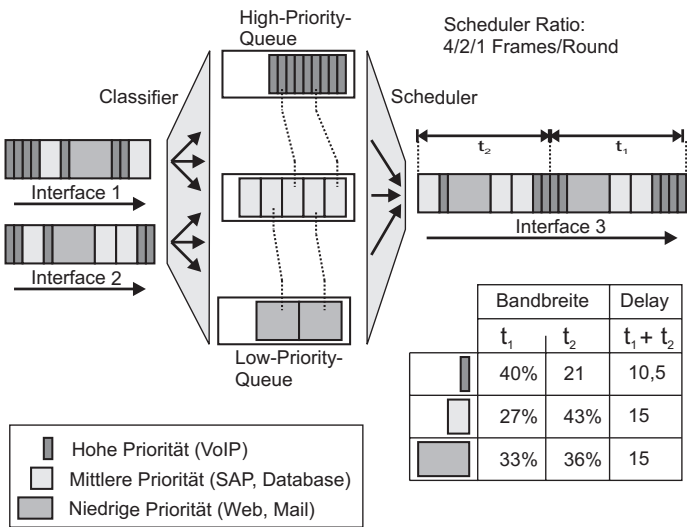


Abbildung 9.2: Fair Queueing kann unter Umständen zu unerwünschten Verzögerungen führen.

Als Queueing bezeichnet man im Netzwerk-Fachjargon die Verwendung von Warteschlangen, in der sich Netzwerkpakete befinden, die auf einer bestimmten Schnittstelle ausgegeben werden sollen. Normalerweise werden diese Warteschlangen, im einfachsten Falle eine, meist jedoch vier bis acht, fest einer bestimmten Netzwerkschnittstelle zugeordnet. Falls mehr als eine Warteschlange pro Interface verwendet wird, haben unterschiedliche diese Prioritäten und dürfen somit pro Zeitintervall unterschiedliche Datenmengen auf dem zugeordneten Interface ausgeben. Ein so genannter Scheduler legt mittels eines speziellen Algorithmus fest, wann welche Warteschlange wie viele Pakete auf einem Interface ausgeben kann. Dabei wird trotz allem garantiert, dass auch die Warteschlange mit der niedrigsten Priorität unter jeder Lastbedingung noch Pakete abgeben darf. Dies bezeichnet man in diesem Zusammenhang auch als Non-Blocking.

„Weighted“, also gewichtet, bedeutet, dass nicht die Anzahl von Datenpaketen in einer Queue, sondern ihre Größe berücksichtigt wird. In der Praxis bedeutet dies, dass die Leerung der Warteschlangen aufgrund ihrer tatsächlichen Größe (in Byte) und nicht aufgrund der Anzahl von Paketen erfolgt. Auf diese Weise werden vor allem Protokolle mit kleinen Paketgrößen, z.B. Voice-over-IP-Pakete, nicht benachteiligt – also fair behandelt. Ein weiterer angenehmer Nebeneffekt von WFQ liegt darin, dass sich bestimmte Flüsse auf eine sehr gut vorhersagbare Paketverzögerung einpegeln.

Welches Paket in welche Queue gelangt, wird aufgrund bestimmter Merkmale im Paket-Header, also aufgrund von Adressen, Adresspaaren, DSCP, Protokoll- oder Portnummern usw., entschieden. Man spricht im Zusammenhang mit QoS sehr oft von Flüssen. Ein Fluss in diesem Zusammenhang ist das logische Zusammenfassen von Paketen in eine Gruppe aufgrund gemeinsamer Identifikationsmerkmale. Ein Fluss kann sich beispielsweise durch gleiche IP-Adresspaare oder gleiche Protokoll- oder Portnummern der zugehörigen IP-Pakete definieren.

In den Abbildung 9.2 und Abbildung 9.3 sehen wir normales Fair Queueing (FQ) und Weighted Fair Queueing (WFQ) im direkten Vergleich. In dem Beispiel handelt es sich um ein System, das zu den betrachteten Zeitpunkten t_1 und t_1 auf den beiden Eingangs-Interfaces Datenströme zugeführt bekommt, die auf dem dritten Interface ausgegeben werden sollen. Alle Schnittstellen haben die gleiche Bandbreite, so dass in dem Beispiel doppelt so viele Pakete pro Zeiteinheit eingeht, wie das System wieder verlassen können. Es werden drei Warteschlangen für hohe, mittlere und niedrige Priorität eingesetzt.

Des Weiteren wird ein klassenbasierender QoS-Ansatz verwendet, mit den drei wie folgt definierten Klassen:

Niedrige Priorität für unkritischen Best-Effort-IP-Transport wie E-Mails, Web- und sonstigen Verkehr. Hier sind kleine Verzögerungen und hohe Bandbreiten nicht erforderlich.

Mittlere Priorität für geschäftskritische Nicht-Echtzeit-Applikationen, zum Beispiel SAP oder Datenbankanwendungen, die jedoch bereits bestimmte Mindestanforderungen an Durchsatz und Verzögerung stellen.

Hohe Priorität für echtzeitähnlichen Verkehr, hier: Voice over IP (VoIP). Hierfür werden minimale Verzögerungen und ein bestimmter garantierter Durchsatz gefordert, also ähnliche Verhältnisse wie im analogen oder digitalen Telefonnetz.

Im Beispiel in Abbildung 9.2 (Fair Queueing) werden die eingehenden Pakete in der Reihenfolge ihrer Ankunft klassifiziert und je nach Zugehörigkeit in die entsprechenden Warteschlangen geschoben. Gleichzeitig fängt der Scheduler an, Pakete aus den Warteschlangen auf Interface 3 auszugeben. Der eingesetzte Algorithmus ist so ausgelegt, dass er in jeder so genannten Runde vier Pakete aus der High-Priority-Queue, zwei Pakete aus der Queue mit mittlerer Priorität und ein Paket aus der Low-Priority-Queue ausgibt. Somit ist garantiert, dass auch Pakete mit niedrigster Priorität nicht blockiert werden und das Verhältnis der ausgegebenen Pakete (4/2/1) ihrer Priorität entspricht.

Wenn wir uns das Datenmuster auf Interface 3 genauer anschauen, sehen wir allerdings, dass dieses keineswegs optimal ist und teilweise krasse Verstöße gegen unsere Qualitätsforderungen, insbesondere bei den VoIP-Paketen, aufweist. Denn bei diesen sind relative große Verzögerungen zu beobachten, das letzte Paket wird sogar erst gegen Ende der Periode t_2 ausgegeben! Auch die tatsächliche Bandbreite entspricht keineswegs dem, was angesichts der vom Scheduler erfolgten Zuteilung 4/2/1 vielleicht erwartet wurde, nämlich 58%, 28% und 14%. Tatsächlich gemessen sind (in Periode t_1) 40%, 27% und 33%! Dieses Verhalten liegt darin begründet, dass die Pakete unterschiedlich groß sind, ein Paket der niedrigsten Priorität verschlingt viermal so viel Bandbreite wie eines der höchsten Priorität.

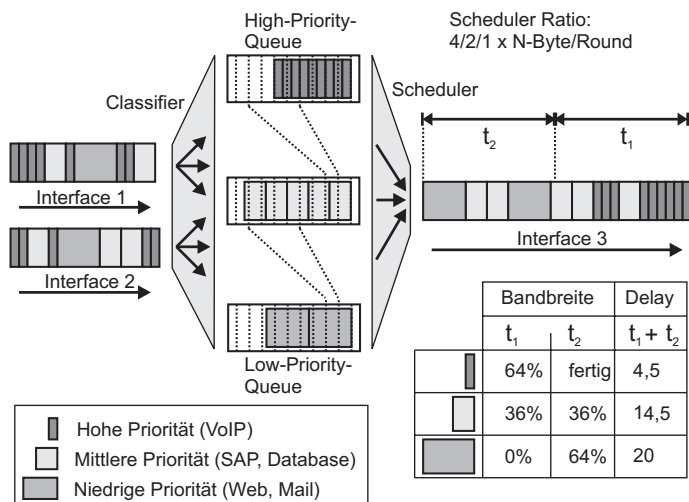


Abbildung 9.3: Weighted Fair Queueing wird nicht von unterschiedlichen Paketgrößen beeinflusst.

Diesen negativen Effekt hinsichtlich Bandbreite und Verzögerung kann man weitgehend durch den Einsatz von Verfahren wie Weighted Fair Queueing oder auf bestimmte Eigenschaften optimierten Ablegern dieser Technologie eliminieren. Der wesentliche Unterschied zu dem vorherigen Verfahren liegt darin, dass der Scheduler die Berechnung nicht mehr aufgrund der Anzahl der Pakete in den Warteschlangen, sondern aufgrund ihrer tatsächlichen Füllung, gemessen z.B. in Byte, vornimmt. Das Resultat sieht man in Abbildung 9.3. Hier werden alle VoIP-Pakete im Zeitraum t_1 ausgegeben (niedrige Verzögerung), und die Bandbreite liegt auch im gewünschten Bereich von 64% (Soll: 58%).

Die hier näher beleuchtete Variante von WFQ wird in der Literatur auch gelegentlich als Class-based WFQ bezeichnet, da der Scheduler die Pakete nach ihrer Zugehörigkeit zu bestimmten Dienstgüteklassen verteilt.

Neben dieser Art von Warteschlangen findet man in vielen QoS-Implementierungen zusätzlich auch ein so genanntes Strict Priority Queueing. Diese Warteschlangen werden geleert, bevor die anderen, gewichteten Warteschlangen bearbeitet werden. Diese Queues werden für Echtzeit-Applikationen benutzt, die keine großen Verzögerungen dulden. Manchmal findet man auch mehrere Strict Priority Queues mit unterschiedlicher Priorität. In diesem Fall werden alle diese Warteschlangen geleert, bevor die gewichteten Warteschlangen bearbeitet werden.

Allerdings sind die Warteschlangen unter ungünstigen Umständen auch irgendwann vollständig gefüllt, und dann werden in dem betreffenden Gerät alle für die entsprechende Schnittstelle bestimmten Pakete verworfen, egal welche Priorität sie besitzen. Dieser Negativeffekt wird noch zusätzlich dadurch verstärkt, dass sich bereits in den Warteschlangen befindende Pakete, die jedoch durch das Verwerfen von logisch damit verbundenen, nachfolgenden Daten sinnlos geworden sind, trotzdem weitergeleitet werden, um dann später vom Endgerät wieder verworfen zu werden.

9.3.3 Random Early Discard (RED), Weighted Random Early Discard (WRED)

Es wäre daher sinnvoll, dieser Situation bereits im Vorfeld entgegenwirken zu können, und genau dies versucht man mit RED. Dieses Verfahren (Abbildung 9.4) tut dabei prinzipiell nichts anderes, als bereits damit anzufangen, Pakete zu verwerfen, bevor eine sich androhende Überlast, beispielsweise erkennbar an einer sehr hohen und tendenziell weiter steigenden Füllung der Warteschlangen, tatsächlich eingetreten ist. Manche Session-Protokolle, meist handelt es sich dabei natürlich um TCP, verfügen über eine intelligente Flusssteuerung, die es ermöglicht, sich dynamisch an Netzwerkengpässe und dann beispielsweise die Übertragungsrate an gegebene Situationen anzupassen, falls Pakete verworfen werden.

Erkennt RED eine sich anbahnende Überlastsituation, so werden eingehende Pakete auf zufälliger Basis verworfen. Die Session-Protokolle in den Endgeräten erkennen daraufhin, dass die Pakete von der Gegenseite noch einmal angefordert bzw. nicht bestätigt werden, und senken daraufhin die Übertragungsrate, wodurch die drohende Überlast möglicherweise vermieden werden kann. Durch die zufällige Auswahl (Random) der zu verwerfenden Pakete, werden rein statistisch gesehen alle Datenflüsse auf geringere Übertragungsraten justiert – auch solche, bei denen das vielleicht gar nicht erwünscht ist.

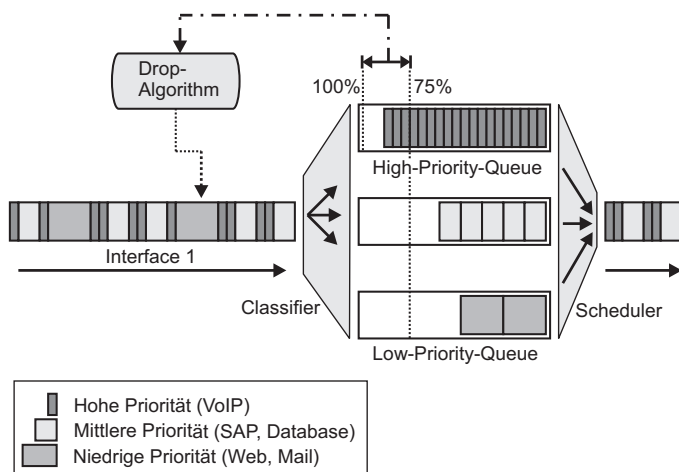


Abbildung 9.4: Random Early Discard wirkt einer drohenden Überlast entgegen.

Eine mögliche Lösung dieses Problems ist WRED, das sich von RED dadurch unterscheidet, dass es nicht auf Zufallsbasis beliebige TCP-Pakete verwirft, sondern seine Auswahl aufgrund gewisser Kriterien wie Paketpriorität oder Klassenzugehörigkeit trifft. Somit können Pakete mit niedriger Priorität eher verworfen werden als solche mit hoher Priorität, was dazu führt, dass nicht so wichtige Applikationen ihre Datenflüsse auf einen niedrigeren Durchsatz einregeln können, während höher priorisierte Anwendungen davon gar nichts mitbekommen.

Protokolle, die nicht verbindungsorientiert arbeiten, z.B. UDP, nehmen an diesem Prozess natürlich nicht teil, und RED hat hier keinerlei Auswirkungen. Allerdings wird UDP auch oft dann verwendet, wenn entweder eine Flusssteuerung auf höheren Schich-

ten des OSI-Modells erfolgt, hier kann der Ansatz von RED je nach Applikation wieder innerhalb gewisser Grenzen funktionieren, oder wenn man schlicht und einfach keinen QoS benötigt.

Es gibt mittlerweile eine Fülle verschiedenster Varianten und Implementierungen dieser Verfahren, z.B. Flow-Based WFQ (FBWFQ), Class-Based WFQ (CBWFQ), um nur einige zu nennen, die alle unterschiedliche Schwerpunkte haben und sich oft nur in kleinen Details oder gar nur dem Namen nach unterscheiden. Aber Vorsicht, keines dieser Protokolle, auch nicht in Kombination miteinander (üblich sind Kombinationen aus aufeinander abgestimmten WFQ- und RED-Implementierungen), ist in der Lage, bestimmte Performance-Garantien zu bieten!

9.3.4 Strict Priority Queueing

Neben den gewichteten Queueing-Verfahren gibt es noch das so genannte Strict Priority Queueing. Warteschlangen dieses Typs werden praktisch immer im Zusammenspiel mit WFQ eingesetzt und zeichnen sich dadurch aus, dass sie zuerst geleert werden und dann mit dem Leeren der gewichteten Warteschlangen begonnen wird. Der Zweck ist offensichtlich: Hier werden besonders zeitkritische Pakete, meist von Applikationen mit Echtzeitcharakteristik, transportiert. Solche Applikationen sind meist interaktive Sprach- oder Videoanwendungen.

Die Verzögerung dieser Warteschlangen ist sehr klein und hängt ausschließlich von der Datenmenge der entsprechend markierten Pakete ab. Man findet oft mehrere, meist zwei, Strict Priority Queues mit unterschiedlicher Priorität. Entsprechend ihrer Wertigkeit bekommen sie unterschiedlich große Zeitscheiben zugeteilt.

9.3.5 Fazit

Das Zusammenspiel von WFQ, RED und Strict Priority Queueing ist daher die Grundlage sehr vieler Quality-of-Service-Implementierungen in Routern, Switches oder VPN-Gateways. Da sowohl auf Layer 2 (z.B. Ethernet) als auch auf der Netzwerkschicht, hier interessiert natürlich vor allem IP, eine Reihe von Standards zur Klassifizierung und Markierung von Paketen oder Frames existiert, eröffnet dies natürlich einige interessante Aspekte. Denn durch geeignete Abbildung (Mapping) der verschiedenen Dienstklassen dieser Standards aufeinander kann man einen weitgehend durchgängigen QoS erreichen. Denn es nutzt nichts, in den LANs seiner Standorte mit modernen Switches mit acht Hardware-Queues pro Interfaces, speziellen ASICs usw. im lokalen Netz fast einen perfekten QoS zu garantieren, und die Kommunikation dieser Standorte untereinander erfolgt durch eine unterdimensionierte WAN-Verbindung mit Best-Effort-Forwarding.

9.4 Flussbasierende Dienstgüte

Flussbasierende QoS ist ein Ansatz, in dem ein verbindungsorientierter QoS realisiert werden soll. Das bedeutet, dass für eine bestimmte Verbindung während deren Lebensdauer entlang einer Übertragungsstrecke bestimmte Qualitätskriterien von jeder beteiligten Ressource garantiert werden müssen. Im Klartext: Sobald eine Verbindung, nehmen wir eine TCP-Verbindung, zwischen zwei Rechnern aufgebaut wird, müssen sich

alle in der Übertragungsstrecke befindlichen Systeme untereinander darüber abstimmen, ob die angeforderten Ressourcen verfügbar sind, und müssen sie im Erfolgsfall für diese Verbindung auch reservieren und bereitstellen. Das Ganze, im Fachjargon auch als Integrated Services (IntServ) bezeichnet, wurde im so genannten Resource Reservation Protocol (RSVP) spezifiziert. RSVP arbeitet immer unidirektional, das heißt für eine bidirektionale Datenkommunikation muss die Signalisierung und Reservierung zweimal durchgeführt werden.

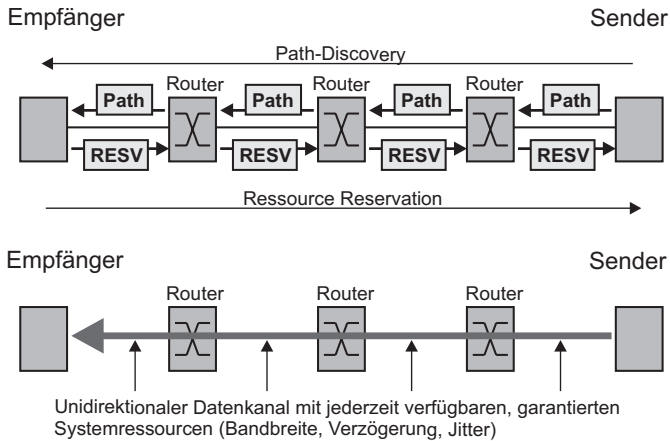


Abbildung 9.5: Der Aufbau einer Verbindung in RSVP beansprucht einiges an Ressourcen.

Der Sender schickt zum Aufbau eines Pfades eine PATH-MESSAGE zum Empfänger. Jedes RSVP-fähige System entlang des Weges erzeugt einen PSB (Path State Block) und registriert das RSVP-System, von dem die Message geschickt wurde, als seinen PHOP (Previous Hop). Dies läuft auf allen beteiligten Systemen ab, bis die Message beim Empfänger angekommen ist. Der Empfänger erzeugt eine FLOWSPEC (Flow Specification), in der die Charakteristika der Reservierung beschrieben sind, und schickt diese in einer RESV-MESSAGE zu seinem PHOP zurück. Erhält ein System eine RESV-MESSAGE, zu der bereits ein PSB angelegt wurde, wird überprüft, ob die Reservierung für den betreffenden Datenstrom durchführbar ist. Wenn ausreichend Ressourcen verfügbar sind, wird ein RSB (Reservation State Block) erzeugt und die RESV-MESSAGE an den jeweiligen PHOP weitergeleitet. So nimmt die Signalisierung den Weg zurück zum Sender.

Am Ende dieses Prozesses ist ein Pfad für den Datenfluss mit den entsprechenden Systemressourcen reserviert. Kann ein System keine Reservierung durchführen (z.B. zu wenig Bandbreite oder zu wenig Speicher), werden entsprechende Fehlermeldungen verschickt.

Am Ende einer RSVP-Session verschickt entweder der Empfänger der Nutzdaten eine RESVTEAR(Reservation-Teardown)-Nachricht, um die Reservierungen freizugeben, oder falls der Sender eine Beendigung wünscht, verschickt dieser eine PATHTEAR(Path-Teardown)-Nachricht.

Leider ist dieses Protokoll wegen des immensen Aufwands für die Signalisierung und der teilweise sinnlosen Reservierung von Ressourcen, die gar nicht mehr benötigt werden, in großen Netzen nicht besonders gut skalierbar und deshalb fast nirgendwo als QoS-Protokoll im Einsatz. Jedoch dort, wo echtzeitähnliche, garantierte Grenzwerte für Verzögerungszeiten oder Mindestbandbreiten benötigt werden und eine überschaubare Infrastruktur vorliegt, kann RSVP durchaus eine Alternative darstellen.

9.5 Klassenbasierende Dienstgüte, DiffServ

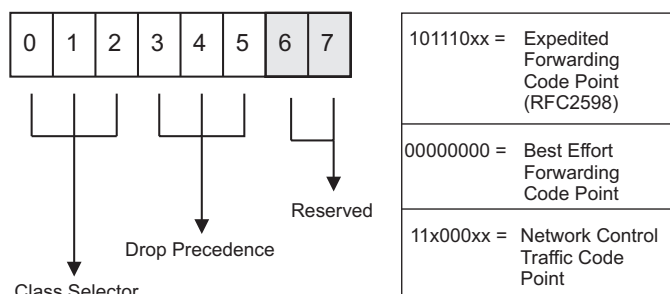
Im Gegensatz zu flussbasierender QoS setzt klassenbasierende QoS, auch DiffServ (Differentiated Services, RFC2475) genannt, auf ein anderes Prinzip: Hier werden verschiedene, wohl definierte Service-Klassen spezifiziert, die jedem Vermittlungssystem bekannt sind und dort auch umgesetzt werden können. Die IP-Pakete werden dafür von Endsystemen oder intelligenten Netzwerksystemen aufgrund der Applikation, die sie erzeugt hat, klassifiziert und entsprechend markiert. Die Vermittlungssysteme (Switches, Router, VPN-Systeme) erkennen anhand dieser Markierung, dem Differentiated Services Code Point (DSCP), welcher Klasse das Paket angehört, und behandeln es entsprechend.

Die Systeme kennen also nur Service-Klassen und keine expliziten Datenflüsse oder Verbindungen. Sie sind für jede Service-Klasse auf ein bestimmtes Verhalten (PHB, Per Hop Behaviour) konfiguriert, das heißt, sie leiten eingehende Pakete in entsprechende Warteschlangen verschiedener Priorität oder verwerfen sie unter bestimmten Umständen (z.B. Überlast) auch.

9.5.1 Die DiffServ-Architektur

DiffServ verlegt die Komplexität der Klassifizierung des Datenverkehrs auf die Grenzen des Netzwerks. Diese Systeme (DiffServ-Edge-Router) müssen die zu übertragenden Datenpakete beim Eintritt in das Netz klassifizieren, markieren und bereits ihrer Service-Klasse entsprechend behandeln. Die Router im Netzwerk müssen die Entscheidung, wie ein Paket zu behandeln ist, lediglich aufgrund des DSCP-Felds im IP-Header treffen. Es erfolgt weder eine Signalisierung zwischen den Routern noch müssen die Zustände einer Vielzahl verschiedener Datenflüsse verwaltet werden. Viele dieser Router haben bereits Mechanismen und Ressourcen zur Paket-Priorisierung implementiert, so dass eine Erweiterung auf die DiffServ-Architektur nur ein logischer Schritt ist, der meist keine teuren Hardware-Erweiterungen der Systeme nach sich zieht.

Die Architektur von DiffServ legt fest, dass die Klassifizierung und Markierung der Pakete in einem so genannten Edge-Router erfolgt, also einem Router an der Grenze eines Netzwerks. Endsysteme oder Applikationen müssen diese Architektur nicht unterstützen, können dies aber, und tun es auch zunehmend. Wo die Klassifizierung und Markierung letztendlich stattfindet, ist daher unterschiedlich und hängt neben der Qualitätsstrategie des Netzwerks auch von den verschiedenen beteiligten Komponenten, also Applikationen, Betriebssystemen und Netzwerksystemen, ab.



Assured Forwarding Code Points (RFC2597)

	Class 1	Class 2	Class 3	Class 4
Low Drop Precedence	001010	010010	011010	100010
Medium Drop Precedence	001100	010100	011100	100100
High Drop Precedence	001110	010110	011110	100110

Class Selector Code Points (RFC2474)

Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
000000	001000	010000	011000	100000	101000	110000	111000

Abbildung 9.6: Die neue Funktion des IP-TOS-Byte als Differentiated Services Code Point (DSCP)

Es kann sowohl eine Klassifizierung in den Endsystemen stattfinden, z.B. durch Applikationen, oder dies kann auch auf dedizierten Netzwerkkomponenten nach entsprechenden Regeln (Policies) erfolgen. Diese Regeln können zentral festgelegt und verwaltet und dann auf die beteiligten Systemkomponenten verteilt werden. Es kann auch nötig sein, durch geeignete Policies bestimmte Klassifizierungen an Netzwerkgrenzen zu ändern, da unter Umständen in Endgeräten Markierungen durch Applikationen oder Benutzer vorgenommen wurden, die nicht mit der netzwerkweiten QoS-Strategie konform sind.

DiffServ ist ein Protokoll in der Netzwerkschicht, das somit unabhängig von Layer-2-Protokollen (wie Frame Relay oder ATM) ist. Daher kann es auf einer Vielzahl von Infrastrukturen genutzt werden. Die VPN-Router müssen natürlich entsprechend konfiguriert werden, um die entsprechende Service-Klasse korrekt zu behandeln, also muss beispielsweise festgelegt werden, welche ATM- oder Ethernet-Parameter wie zu konfigurieren sind. Aber dies erfolgt nur in eine Richtung, aus der Sicht des Schichtenmodells von oben nach unten, und es findet, bis auf wenige optionale Ausnahmen, keine Wechselwirkung in der Art statt, dass Layer-2-Spezifika die Service-Klasse beeinflussen. Dies ist ein weiterer wichtiger Pluspunkt zugunsten des DiffServ-Modells, denn sehr viele größere Netze benutzen zum Transport von IP-Paketen eine Vielzahl verschiedener Infrastrukturen (ATM, IEEE802.1q, Frame Relay usw.), die unterschiedliche Quality-of-Service- und Priorisierungsmechanismen bieten. Diese können den DSCP auswerten, ihn aber nicht ändern, so dass die Klasseninformation während der vollständigen Übertragung erhalten bleibt.

9.5.2 Die DiffServ-Service-Klassen

Zurzeit sind in DiffServ drei verschiedene Service-Klassen festgelegt:

- Premium
- Tiered
- Best Effort

Zu jeder dieser Klassen gibt es ein festgelegtes Per Hop Behaviour (PHB), das in den Routern oder Switches eines DiffServ-Netzwerks unterstützt werden muss. Das PHB unterscheidet zwischen unmittelbarem Weiterleiten (EF, Expedited Forward), garantier-tem Weiterleiten (AF, Assured Forward) und normalem Weiterleiten (DF, Default Forward) der IP-Pakete. Wie das die verschiedenen Systeme technisch realisieren, ist nicht im Standard festgelegt.

Für den so genannten Premium-Service muss jeder Router permanent einen gewissen Teil seiner Ressourcen reservieren, unabhängig von seiner tatsächlichen Auslastung. Für den Tiered-Service gilt das Gleiche, jedoch werden diese Ressourcen in verschiedene Prioritätsstufen untergliedert. Was an Ressourcen übrig bleibt, wird vom Best-Effort-Service benutzt.

In Abbildung 9.6 sehen Sie den Aufbau des DSCP-Felds im IP-Header. DiffServ benutzt derzeit, wie im RFC2474 festgelegt, die ersten sechs Bits davon. Davon legen die ersten drei Bits die Klasse fest, und die folgenden drei Bits geben die Vorrangigkeit innerhalb der jeweiligen Klasse an, mit der Pakete in Überlastsituationen verworfen werden können. Der Codepunkt 101110 markiert den Premium-Service und der Codepunkt 000000 den Best-Effort-Service.

9.5.3 Der DiffServ-Edge-Router

Der Klassifizierer ist die Funktion, die in einem so genannten Edge-Router ein eingehendes Paket daraufhin prüft, welche Bits im DSCP-Feld zur Klassifizierung des Pakets gesetzt werden müssen. Er teilt die IP-Pakete somit in verschiedene Klassen ein, die der Betreiber des Transportnetzwerks unterstützt. Nach welchen Kriterien eine solche Klassifizierung erfolgt, richtet sich entweder nach dem Service-Level-Agreement (SLA), das mit dem Provider abgeschlossen wurde, und/oder nach der Qualitätsstrategie des Netzwerks. Die Entscheidung des Klassifizierers erfolgt zum Beispiel aufgrund von Quell- oder Zieladressen, Port- oder Protokollnummern oder aufgrund des DSCP-Felds. Denn es kann durchaus sein, dass ein Paket verschiedene Netze unterschiedlicher Provider durchläuft oder dass in einem Kundennetz bereits eine DiffServ-Klassifizierung erfolgt ist.

Zurzeit erfolgt die Markierung statisch nach fest im Edge-Router vorgegebenen Regeln. Anschließend wird das markierte Paket seiner Markierung entsprechend in den passenden Ausgangspuffer geschrieben und von dort in das Transportnetz weitergeleitet. Zusätzlich findet man im Edge-Router neben dem Markierer und dem Klassifizierer noch einen so genannten Shaper. Dieses Modul sorgt dafür, dass der Datenverkehr in das Transportnetz einen bestimmten vereinbarten Wert oder die mögliche Bandbreite einer Verbindung nicht überschreitet.

9.5.4 Der DiffServ-PHB-Router

Der PHB-Router braucht sich keine Gedanken mehr über die Klassifizierung oder Markierung eines Pakets zu machen, er muss es nur aufgrund der Einträge im DSCP-Feld verarbeiten und in die korrekte Warteschlange stellen. Der PHB-Router weist eine Teilmenge der Funktionen des Edge-Routers auf und wird somit viel weniger belastet. Die Verarbeitung ist demzufolge relativ schnell, denn es muss nur das DSCP-Feld ausgewertet werden. Da das PHB für die EF-Klasse bedingt, dass ein jitterfreies, schnelles Weiterleiten der Pakete garantiert ist, muss der Provider umfangreiche Maßnahmen treffen, um dies zu erreichen – und dies verursacht natürlich entsprechende Kosten. Zukünftige Modelle, die auf dem Einsatz so genannter dynamischer Policy-Server basieren, ermöglichen eine dynamische Zuteilung von Ressourcen für die EF-Klasse aufgrund der aktuellen Netzlast und befreien damit von der statischen Reservierung der notwendigen Betriebsmittel.

Netzwerkweiter Quality of Service erfordert auch, dass diese verschiedenen Technologien und Verfahren miteinander kombiniert werden. Ein vielversprechender Ansatz ist die Idee, DiffServ zu benutzen und in den entsprechenden Systemen ein entsprechendes Mapping, also eine Abbildung auf die jeweilige Technologie in diesen Geräten, vorzunehmen.

Nortel Networks Service Class	DiffServ Code Point (DSCP)	ATM Service Category	PPP Class Numbers	IEEE 802.1p User Priority
Gold	AF4x, CS4	rt-VBR	5	5
	AF3x, CS3		4	4
Silver	AF2x, CS2	nrt-VBR	3	3
Bronze	AF1x, CS1		2	2
Standard	DE, CS0	UBR	1	0

Tabelle 9.6: Das Abbilden verschiedener Dienstgüteklassen auf die in verschiedenen Abschnitten eines Netzwerks eingesetzten Technologien

Verschiedene Hersteller, z.B. Microsoft oder Nortel Networks, haben verschiedene, eigene Dienstklassen definiert, die mit bestimmten Klassen der DiffServ-Architektur korrespondieren. Diese Klassen definieren sich innerhalb unterschiedlicher Verkehrskategorien, wie in Tabelle 9.7 für Microsofts und Nortel Networks Dienstklassen zu sehen ist. Diese Dienstklassen korrespondieren mit entsprechenden DiffServ-Klassen, die im DSCP von IP-Headern markiert werden können. Netzwerksysteme wie Layer-2/3-Switches, Router, Gateways, ATM- oder Frame-Relay-Systeme können diese Information auswerten und auf ihre jeweilige Layer-2-Technologie abbilden (mappen), um die gewünschte Dienstqualität so gut wie mit der gegebenen Technologie möglich zu gewährleisten.

Network Control	Critical		CS7
Guaranteed	Network	0x30 (CS6)	CS6
	Premium	0x28 (CS5)	EF, CS5
	Platinum		AF4x, CS4
Controlled Load	Gold	0x18 (CS3)	AF3x, CS3
	Silver		AF2x, CS2
Qualitative	Bronze	0x00 (DE)	AF1x, CS1
Best Effort	Standard	0x00 (DE)	DE, CS0
Non-Conforming Traffic	Custom	0x00 (DE)	Custom

Tabelle 9.7: Ein Vergleich der Dienstklassen verschiedener Hersteller (z.B. Microsoft und Nortel) und ihrer Abbildung auf den DiffServ Code Point (DSCP)

Um keine Änderung des IP-Header-Formats notwendig machen zu müssen, benutzt DiffServ das TOS-Byte (TOS, Type of Service), das zweite Byte des IP-Headers, als so genannten Differentiated Services Code Point (DSCP). Dieses Byte wurde im RFC791 wie folgt spezifiziert:

- Bits 0-2: Precedence
- Bit 3: 0 = Normal Delay, 1 = Low Delay
- Bit 4: 0 = Normal Throughput, 1 = High Throughput
- Bit 5: 0 = Normal Reliability, 1 = High Reliability
- Bits 6-7: Reserved for Future Use.

Die Bedeutung der Bits wurde im RFC2474 (Definition of the Differentiated Services Field in the IPv4 and IPv6 Headers) neu festgelegt, um den heutigen Anforderungen gerecht zu werden. Das ist kein großes Problem, denn das TOS-Feld wurde fast niemals benutzt. TOS und DSCP sind nicht miteinander kompatibel, auch wenn die drei Precedence Bits im TOS-Byte und die Class Selector Code Points (s.u.) in etwa miteinander vergleichbar sind.

PHB sind Aggregate, in denen IP-Pakete gleicher QoS-Klassen zusammengefasst werden. Innerhalb eines solchen Aggregats werden die Pakete hinsichtlich QoS nicht mehr voneinander unterschieden. Die Klassen sind in verschiedenen RFCs festgelegt und auf verschiedene Verkehrskategorien zugeschnitten. Folgende PHB sind definiert:

EF (Expedited Forwarding), RFC2598

AF (Assured Forwarding), RFC2597

CS (Class Selector), RFC2474

BE (Best Effort), RFC2474

Man kann innerhalb von Unternehmensnetzen auch eigene DSCP-Werte und eigene PHB verwenden, ist dann jedoch inkompatibel zu standardkonformen Implementierungen. Das kann dann unangenehm werden, wenn es zu einem Mix solcher Implementierungen kommt, denn das Standardverhalten ist es, bei Paketen mit einem unbekannten bzw. inkompatiblen DSCP den Best Effort PHB zu benutzen.

Die EF-Klasse wurde implementiert, um Anwendungen, die in diese Kategorie fallen, eine entsprechende Dienstgüte bereitzustellen. EF wird mit der höchsten Emission Priority und der niedrigsten Discard Priority implementiert. In der Praxis müssen die Netzknoten für Pakete der EF-Klasse niedrigste Verzögerung, Jitter und Paketverlust garantieren. EF wird hauptsächlich zur Simulation der Qualität von leitungsvermittelnden Netzen (interaktive Sprache, Video) in IP-Umgebungen eingesetzt.

Insgesamt besteht die AF-Klasse aus zwölf verschiedenen Selektoren, aus denen sich Router und Switches die notwendigen Informationen für das Queueing und das während oder vor einer drohenden Überlast notwendige Verwerfen von Paketen entnehmen. Die AF-Klasse besteht aus vier verschiedenen Service-Klassen (AF1x bis AF4x), innerhalb derer jeweils drei unterschiedliche Discard-Prioritäten (Drop Precedence Level) möglich sind (AFx1 bis AFx3).

AF4x hat die höchste Emission Priority und AF1x die niedrigste. AFx3 hat die höchste Discard-Priorität und AFx1 die niedrigste.

In der Praxis erfordert die Discard Priority je nach Last/Überlast und den konfigurierten Verkehrsprofilen ein Remarking des DSCP.

Die Class-Selector-Klasse ist etwas einfacher als die AF-Klasse ausgelegt. Hier gibt es acht mögliche Prioritätsklassen. Es werden dafür die gleichen Bits benutzt, die auch die alte TOS-Spezifikation für das IP-Precedence-Field vorgesehen hat. CS7 hat die höchste Emission Priority und die CS0 die niedrigste, die damit praktisch der Best-Effort-Klasse entspricht.

Discard Priority wird von der CS-Klasse nicht unterstützt! Pakete mit einer CS5-, CS6- oder CS7-Markierung erfordern üblicherweise Strict Priority Queuing und sollten normalerweise niemals verworfen werden.

DE entspricht praktisch auch der Klasse CS0, beide haben den DSCP-Wert 0x0000. BE bedeutet niedrigste Emission Priority und höchste Drop Priority. Man sollte sich aber davon nicht zu sehr abschrecken lassen. Wenn bestimmte Applikationen nicht unbedingt bestimmte QoS-Parameter erfordern, kann man ruhig weiterhin deren Pakete über Best Effort PHB transportieren lassen und sollte kritische Ressourcen auch tatsächlich nur für kritische Applikationen reservieren.

DiffServ-Implementierungen können in Routern, Switchen oder Gateways unterschiedlich angelegt sein, je nach Funktion des Geräts. Normalerweise ist immer die PHB-Router-Funktion implementiert, im Fall eines DiffServ-Edge-Routers kommen noch Funktionen wie Klassifizierung und Markierung von Paketen hinzu. Im Weiteren wird eine vollständige DiffServ-Implementierung (DiffServ Edge Router) besprochen.

In Abbildung 9.7 sehen Sie die verschiedenen Funktionen eines DiffServ-Routers auf einer seiner Schnittstellen. Die physikalischen Schnittstellen werden dabei in ein Ingress Interface für eingehenden Verkehr und ein Egress Interface für abgehenden Verkehr unterteilt.

Eingehende Pakete werden anhand ihrer Layer-3- und Layer-4-Header klassifiziert und von der Marker-Funktion im DSCP markiert. Insbesondere das IP-Protokoll und die TCP/UDP-Portnummern geben Aufschluss darüber, welche Anwendung das zu klassifizierende Paket erzeugt hat.

Das DiffServ-Meter misst die eingehenden Datenströme der verschiedenen DiffServ-Klassen. Der Zweck ist die Prüfung, ob ein konfiguriertes Verkehrsprofil überschritten wird. In diesem Fall werden weitere Funktionen (Policer) in die weitere Bearbeitung der betreffenden Pakete involviert. Der Policer steuert die Menge des zu bestimmten Verkehrsprofilen gehörenden Datenverkehrs. Bei so genanntem Out-of-Profile Traffic, also falls eine bestimmte Datenrate überschritten wird, stehen dem Policer zwei Möglichkeiten zur Verfügung. Er kann entweder die Pakete sofort verwerfen (Dropping), oder er kann die Pakete neu markieren (Re-Marking), indem er eine höhere Drop-Precendence, also um eine höhere Priorität dieses Paket aus der Warteschlange im Egress Interface zu entfernen, im DSCP einträgt.

Der Shaper versucht, Verkehr mit vielen Bursts (Spitzen) etwas zu glätten und in ein konfiguriertes Verkehrsprofil „hineinzuzwängen“. Auf dem Egress Interface erfolgt dies üblicherweise durch geeignetes Queuing – und damit auch Verzögern von Paketen, falls die Queues nicht entsprechend oft bedient werden.

Eingegangene Pakete werden, nachdem dies alles durchlaufen wurde, in eine entsprechende Queue (Warteschlange) geschoben und warten dort auf ihre Übertragung. Die Warteschlangen haben unterschiedliche Prioritäten. Falls eine Warteschlange eine bestimmte Größe überschritten hat, werden auf dem Ingress Interface wahlfrei Pakete, die in diese Warteschlange kommen würden, verworfen. Dies dient dazu, eine Flusssteuerung auf Session- oder Applikationsebene dazu zu bewegen, ihre Datenrate nach unten zu justieren.

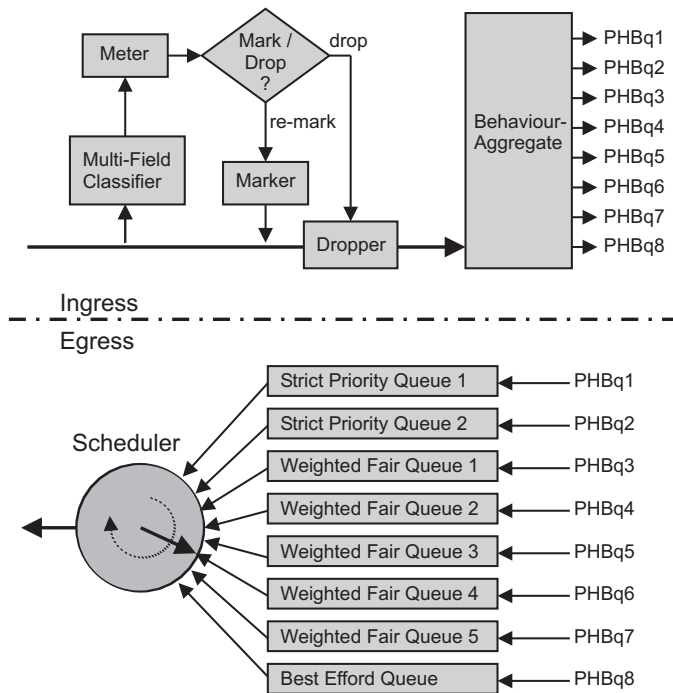


Abbildung 9.7: Ein Interface eines DiffServ-Routers

Warum wird dem Thema QoS in VPN ein eigenes Kapitel gewidmet? Es sind doch für alle Netzwerkarten (Ethernet, Frame Relay, PPP, ATM, MPLS usw.) und vor allem auch für IP (DiffServ) entsprechende QoS-Verfahren spezifiziert. Und moderne Übertragungssysteme wie Switches oder neuere Router haben meist zum Zweck höchster Performance bis zu acht Hardware-Queues auf jeder Schnittstelle. Und genau diese Protokolle, Netze und Systeme werden doch auch von VPN mitbenutzt. Und genau da liegt der Denkfehler! Denn in VPN gibt es keine realen Schnittstellen, sondern nur virtuelle, die teilweise auch nur bei Bedarf aktiviert werden.

Nehmen wir einmal folgenden Fall aus der Praxis: Ein VPN soll 20 kleinere Außenstellen mit einem zentralen Standort über das Internet verbinden. Die Außenstellen werden mit ADSL-Modems und VPN-Routern (192 KBit upstream/ 2 MBit downstream) angebunden, und die Zentrale verfügt über eine 34-MBit/s-Anbindung über Ethernet an den Router eines Service Providers. Hinter diesem Router wird lokal der zentrale VPN-Router des Kunden angebunden. Da kritische Applikationen (VoIP) eingesetzt werden, müssen bestimmte Bedingungen erfüllt sein:

1. Es dürfen gleichzeitig nicht mehr als zwei Gespräche (G.711, ca. 80 KBit/s, 20 ms Voice/ Packet) zu einer Außenstelle stattfinden. Das ist kein Problem, das kann jeder vernünftige Call Manager steuern.

2. Da neben Sprache auch andere Daten übertragen werden, müssen die VoIP-Pakete mit höchster Priorität (Strict Priority Queueing) behandelt werden. Das ist mit DiffServ ebenfalls kein Problem. Oder?

In den Außenstellen gibt es keine Probleme, hier kann man für ausgehende Pakete DiffServ verwenden und per Interface Rate Limiting (so etwas sollte jeder vernünftige VPN-Router besitzen) die Interface-Geschwindigkeit auf 192 KBit/s anstelle der für sein Ethernet-Interface voreingestellten 10 MBit/s konfigurieren. Bei korrekter Konfiguration steuert der VPN-Router die Kommunikation und kann bei zwei gleichzeitigen Gesprächen und hoher Datenlast per WRED die lokalen Datensender zu niedrigerer Übertragungsrate bewegen. Da pro physischem Interface nur eine VPN-Verbindung betrieben wird, sind die Verhältnisse für reale und virtuelle Verbindung praktisch identisch.

In diesem Szenario verfügt die Zentrale über ein reales Interface (34 MBit/s) zum Internet und 20 virtuelle Interfaces zu den Außenstellen. Im realen Interface (Ethernet) ist QoS kein Problem, hier arbeiten unter Umständen vielleicht sogar Hardware-Queues. Hier kommen DiffServ und Ethernet Priority zur Anwendung. Die virtuellen Interfaces verfügen natürlich nicht über Hardware-Queues, sondern hier muss jede Art von QoS per Software emuliert werden.

In unserem Beispiel müssten also pro virtuellem Interface acht Queues mit Strict Priority und Weighted Fair Queueing, Weighted Random Early Discard und Interface Rate Limiting emuliert werden. Das wären alleine zusätzliche 160 Queues per Software in einem System, das auch noch ein paar andere rechenintensive Aufgaben wie Ver- und Entschlüsselung durchzuführen hat. Und 20 VPN-Verbindungen zentral zu terminieren sind eigentlich Charakteristika von sehr kleinen Netzen. In größeren Systemen, die Hunderte oder Tausende von Gegenstellen bedienen, hätte man zig Tausende von Warteschlangen zu emulieren – etwas was definitiv auch ein 100.000-Euro-VPN-System in die Knie zwingt.

Also wird man in der Praxis in bezahlbaren VPN-Routern Verfahren wie DiffServ nur auf realen Interfaces ermöglichen und nicht auf den virtuellen Interfaces. Und genau das ist das Problem; denn DiffServ nur auf dem physikalischen Interface wirft sofort eine Frage auf: Welche Interface-Rate muss man für die ausgehende Richtung einstellen? 20 Außenstellen mal 2 MBit/s macht rein rechnerisch 40 MBit/s, das wäre schon mehr, als das Interface kann, also gelten hier 34 MBit/s. Allerdings gilt dieser Wert nur für das reale Interface. Daher können unter bestimmten Umständen über eine VPN-Verbindung zu einer Außenstelle bis zu 34 MBit/s Datenverkehr einbrechen, wenn über die anderen Verbindungen gerade nichts läuft. Das Resultat wäre extreme Überlastsituation, denn die Außenstellen sind nur mit 2 MBit/s (downstream) angebunden.

Der zentrale VPN-Router kann auf diese Situation nicht reagieren, da der Verkehr ja noch voll innerhalb seines Profils ist. Nur wenn die Geschwindigkeit über 34 MBit/s zu gehen droht, würde er aktiv werden. Je nach Anwendung kann so etwas bis zum Abbruch von Verbindungen führen. Einziger Trost: Die Sprachverbindungen bleiben von den Problemen innerhalb gewisser Grenzen unberührt, denn hier erfolgt die Steuerung der Anzahl der Verbindungen zu einer Außenstelle auf einer anderen Ebene durch den Call Manager. Wenn die VoIP-Pakete richtig markiert sind, werden sie über Strict Priority Queueing sofort weiter transportiert, während sich die restlichen Datenpakete in den anderen Warteschlangen gedulden müssen.

Die Lösung, einfach eine kleinere Geschwindigkeit auf dem Interface des zentralen Routers zu konfigurieren, hätte fatale Folgen: Erstens würde das die Überlastsituation nur dann beenden, wenn die Geschwindigkeit gleich der maximalen Geschwindigkeit der Außenstelle, also 2 MBit/s, wäre. Und zweitens stünden dann für alle 20 virtuellen Verbindungen insgesamt auch nur 2 MBit/s zur Verfügung, und in der Praxis wäre jegliche Kommunikation, vor allem auch die Sprachkommunikation, fast unmöglich. Aber auch Kompromisse mit Geschwindigkeiten irgendwo zwischen 2 und 34 MBit/s sind faule Kompromisse, denn auch hier würde den Sprachverbindungen Bandbreite fehlen.

Aber auch andere Faktoren moderner VPNs wirken sich unmittelbar auf den QoS aus. So entstehen durch Aktivitäten wie Tunneling, Ver- oder Entschlüsselung oder Kompression in VPN-Komponenten Verzögerungen, die sich zusätzlich zur Gesamtverzögerung addieren. Alle VPN-Verfahren vergrößern zudem die IP-Pakete, was im Extremfall dazu führen kann, dass die Pakete fragmentiert werden müssen, was zu weiteren Verzögerungen führen kann.

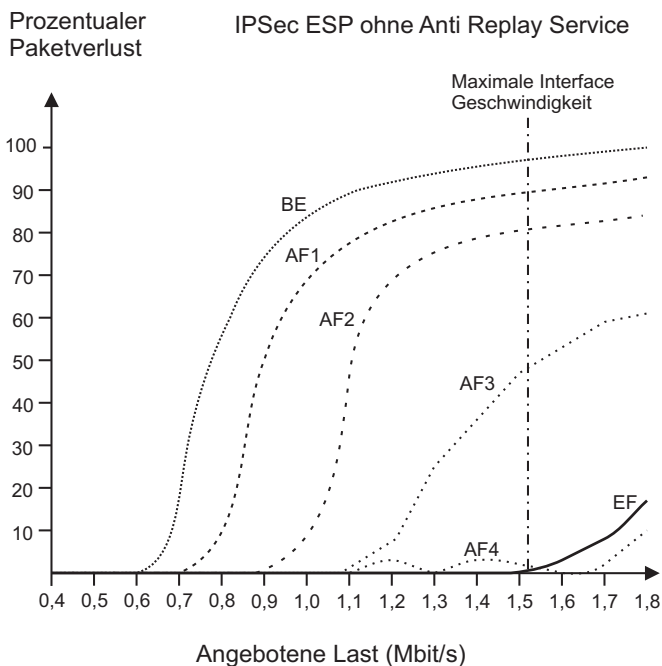


Abbildung 9.8: IPSec ohne Anti Replay Service (ARS) entspricht dem normalen Verhalten von DiffServ-IP-Verkehr.

Viele Sicherheitsverfahren schützen sich gegen Angriffe durch wiederholtes Senden von Paketen (Replay Attack), indem sie mit Sequenznummern arbeiten. Diese Sequenznummern müssen beim Empfänger innerhalb eines gewissen Bereichs (Empfangsfenster, Sliding Window) liegen, ansonsten werden die Pakete verworfen (vgl. IPSec, Kapitel 5). Leider sorgen Verfahren wie DiffServ dafür, dass IP-Pakete unterschiedlich lange in den Warteschlangen zwischengespeichert werden, je nachdem, zu welcher Service-Klasse sie gehören. Damit wird die Reihenfolge der Pakete teilweise stark verändert, und es kann vorkommen, dass die Pakete außerhalb des Empfangsfensters liegen und verworfen werden.

Je schneller ein Interface arbeitet, desto länger können die Warteschlangen für definierte Verzögerungszeiten sein und desto stärker tritt dieser Effekt zutage, weil die Fenstergröße in den meisten Fällen statisch ist. In Abbildung 9.8 ist das Verhalten für verschiedenen markierte Pakete ohne Anti Replay Service (ARS) aufgrund einer Reihe von Messungen dargestellt. Es handelt sich dabei um einen IPSec-Router, der auf der lokalen Seite mit einem 100 MBit Ethernet-Interface und auf der WAN-Seite mit einem T1-Interface (1,5 MBit/s) ausgerüstet ist und eine IPSec-Verbindung zur Gegenseite aufgebaut hat. Als Verschlüsselungsverfahren wurde der Worst Case mit Triple-DES ohne Hardware-Beschleunigung gewählt. Der Verkehr auf dem lokalen Interface, das mehr als 66-mal schneller als das WAN-Interface arbeitet, wurde langsam von 400 KBit/s bis auf knapp 1,8 MBit/s erhöht, also weit über die Datenrate hinaus, die auf der T1-Schnittstelle ausgegeben werden kann.

Gemessen wurde die Datenrate auf der lokalen Schnittstelle der Gegenseite. Das Verhalten der Übertragung ist wie zu erwarten: Am schlechtesten werden die Best-Effort-Pakete (BE) vom VPN-Router behandelt, sie werden lange vor der eigentlichen Überlast auf dem WAN-Interface als erste verworfen. Als nächste Maßnahme bei weiter steigender Geschwindigkeit wird die Datenrate der AF-Pakete abgestuft reduziert. Am längsten werden die EF-Pakete (Expedited Forwarding) ohne Paketverlust übertragen, erst ab dem Punkt der WAN-Interface-Überlastung (1,5 MBit/s) wird es holprig.

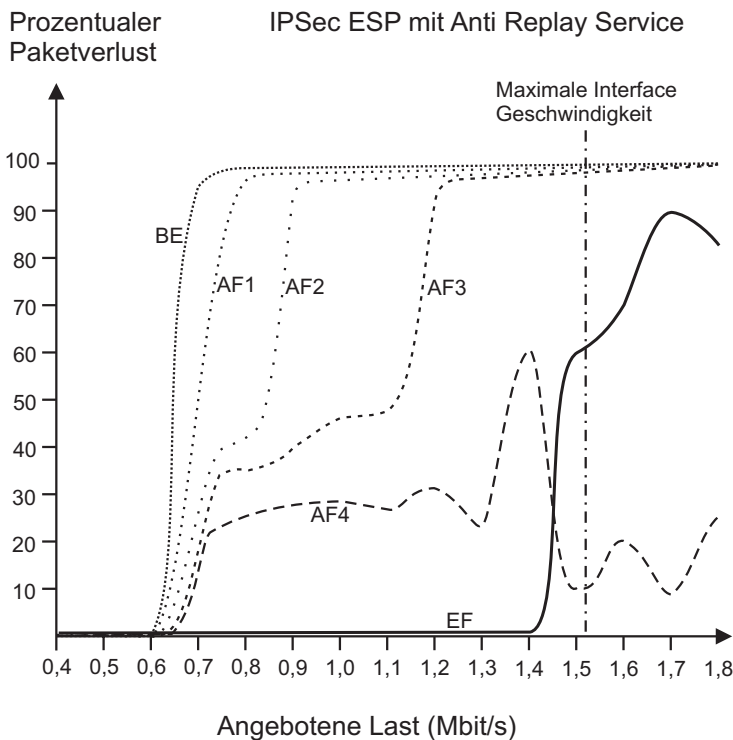


Abbildung 9.9: Mit ARS werden die IPSec-Pakete so lange verzögert, bis sie nicht mehr im ARS-Zeitfenster eintreffen und verworfen werden.

Völlig anders sieht es hingegen aus, wenn in IPSec der Anti Replay Service aktiviert ist. Hier erwartet die Gegenseite, dass die IPSec-Pakete in der Reihenfolge ankommen, in der sie vom Sender verschickt wurden. Genau gesagt, sie müssen innerhalb eines dynamischen Fensters von meist 32 oder 64 aufeinander folgenden Sequenznummern eintreffen. Andernfalls werden die Pakete sang- und klanglos verworfen. Diese Bedingung trifft aber nur für Pakete zu, die nicht allzu lange in Warteschlangen aufgehalten werden. Das sind zuerst einmal Pakete, die per Strict Priority Queueing bearbeitet werden, also vor allem solche mit EF-Markierung im DSCP. Diese Pakete werden, wie man in Abbildung 9.9 sieht, bis kurz vor dem Punkt der tatsächlichen WAN-Interface-Überlastung auch mit aktiviertem ARS ohne nennenswerte Paketverluste übertragen.

Für den Rest der Paket-Klassen (BE und AF1 – AF4) sieht es allerdings verheerend aus: Hier werden teilweise schon bei nur 0,8 MBit/s angebotener Last bereits alle BE- und AF1-Pakete verworfen, also man hat hier 100% Paketverlust. Am wackersten hält sich noch der Durchsatz für AF4-markierte Pakete. Das Verwerfen der Pakete erfolgt wohlge-merkt auf der Empfängerseite, was der Gesamtperformance noch zusätzlich abträglich ist, denn hier werden Pakete übertragen, nur um vom Empfänger sofort verworfen zu werden. Was dies alles für reale Applikationen bedeutet, überlasse ich der Fantasie und Erfahrung des Lesers

Die einzigen praktikablen und gleichzeitig standardkonformen Lösungen sind, die ARS-Fenstergröße stark zu erweitern, wobei das Verfahren dann irgendwann sinnlos zu werden beginnt, oder am besten den Anti-Replay-Service gleich ganz abzuschalten. Hier muss man, falls man Quality of Service in seinem Netz einsetzen will, leider einen Kompromiss zwischen sehr hoher Sicherheit und tatsächlicher Benutzbarkeit machen. Natürlich hängt das Verhalten entscheidend vom Netzdesign ab. Hätte man beispielsweise auf dem lokalen Interface 1,5 MBit/s und auf der WAN-Seite 100 MBit/s zur Verfügung, würde sich das Problem so gar nicht stellen. Aber in der Praxis sind die WAN-Geschwindigkeiten meist der Flaschenhals, den man durch DiffServ optimal nutzen will.

Problematisch kann es auch werden, wenn VPN-Protokolle selbst bestimmte Verfahren eingebaut haben, die sich auf ihre QoS-Charakteristik auswirken. SSL ist zum Beispiel so ein Fall, da es generell zur Übertragung TCP benutzt und damit allen anderen Anwendungen die Merkmale dieses Protokolls aufzwingt – auch solchen, die TCP aus gutem Grund gar nicht verwenden!

Das Thema DiffServ ist in IPSec recht pragmatisch gelöst worden. Auf IP-Ebene gibt es zu diesem ganzen Thema ein einziges Byte: der DiffServ Code Point (DSCP). Der DSCP dient dazu, den Layer-2-Protokollen zu signalisieren, um welche DiffServ-Klasse es sich bei dem betreffenden Paket handelt. IPSec im Transportmode verändert das TOS-Byte im Header nicht, wodurch sich für DiffServ auch nichts ändert.

Im IPSec-Tunnel-Mode hingegen ist das vollständige originale IP-Paket verschlüsselt, und es wird ein neuer IP-Header vorangestellt. Um DiffServ auch hier nicht zu blockieren, kopiert IPSec bei ausgehenden Paketen den DSCP des inneren, verschlüsselten IP-Headers in den äußeren IP-Header – natürlich unverschlüsselt. Bei eingehenden Paketen finden keine Operationen auf dem DSCP statt.

Wie im vorherigen Kapitel erläutert, muss man sich auch, falls das Netzwerkszenario dies erfordert, Gedanken über den Anti Replay Service machen und diesen gegebenenfalls außer Kraft setzen. Übrigens: Der ARS kann nur auf der Empfängerseite ein- oder ausgeschaltet werden. Der Sender trägt immer Sequenznummern ein, und es liegt alleine am Empfänger, diese auch auszuwerten.

Natürlich muss man sich auch fragen, wie die IPSec-Transformationen die Qualitätskriterien von Übertragungen beeinflussen. Natürlich hängt das zum großen Teil von der Leistungsfähigkeit der VPN-Router oder Gateways ab, allerdings können an dieser Stelle ein paar Faustregeln aufgezeigt werden. IPSec (ESP oder AH) ist ein reines IP-Protokoll und damit ein reiner paketvermittelnder Dienst, der ausschließlich auf Layer 3 arbeitet und den Datenflüssen keine bestimmten Verkehrscharakteristiken aufzwingt.

Zuerst einmal werden die Pakete durch die Verschlüsselung, Hashing usw. zusätzlich verzögert. Diese Verzögerung ist umso geringer, je schneller die Systeme arbeiten. Bei der Auswahl von VPN-Komponenten sollte man darauf achten, dass die Systeme mit Echtzeitbetriebssystemen arbeiten. Reine VPN-Gateways neueren Datums, also spezielle Hard- und Software (Appliance) und keine Router oder Server, verzögern Pakete unter normaler Last im Mittel um deutlich weniger als 1 Millisekunde. Dies ist ein Wert, der auch für Sprach- oder interaktive Videoanwendungen voll ausreicht. Gerade bei diesen zukunftssträchtigen Anwendungen, die recht kleine IP-Pakete erzeugen, tritt auch ein anderer störender Effekt nicht auf: Paketfragmentierung. Denn bei großen IP-Paketen kann IPSec unter Umständen so viel Paket-Overhead erzeugen, dass das resultierende IPSec-Paket zu groß für das physikalische Übertragungsmedium ist und daher fragmentiert werden muss.

Das führt uns nun zum Thema Bandbreite. Bei IPSec gibt es einen Faktor, der die Bandbreite negativ beeinflusst und der nicht von der Leistungsfähigkeit der VPN-Systeme abhängt. Es handelt sich dabei einfach um das Größenverhältnis von IPSec-Paket und originalem IP-Paket. Dieser Faktor wird umso ungünstiger je kleiner das originale IP-Paket wird, da der Overhead durch IPSec relativ konstant ist, in der Regel 52 Byte bei IPSec im Tunnel Mode. Das heißt, dass zu einer Paketgröße von 64 Byte (originales IP-Paket) noch 52 Byte durch IPSec hinzugefügt werden und somit nur noch 55% Nutzbandbreite übrig bleiben – nur durch den Overhead von IPSec. Bei größeren IP-Paketen sinkt der Faktor entsprechend schnell, bei einer Paketgröße von 256 Byte hat man bereits 83% und bei 1024 Byte über 95% der theoretisch möglichen Nutzbandbreite. Relevant wird dieser Faktor natürlich bei steigendem Anteil von IP-Sprachverkehr, denn die typischen VoIP-Paketgrößen (RTP) liegen je nach eingesetztem Codec im Bereich zwischen 80 und 200 Byte.

Theoretisch mögliche
Nutzbandbreite

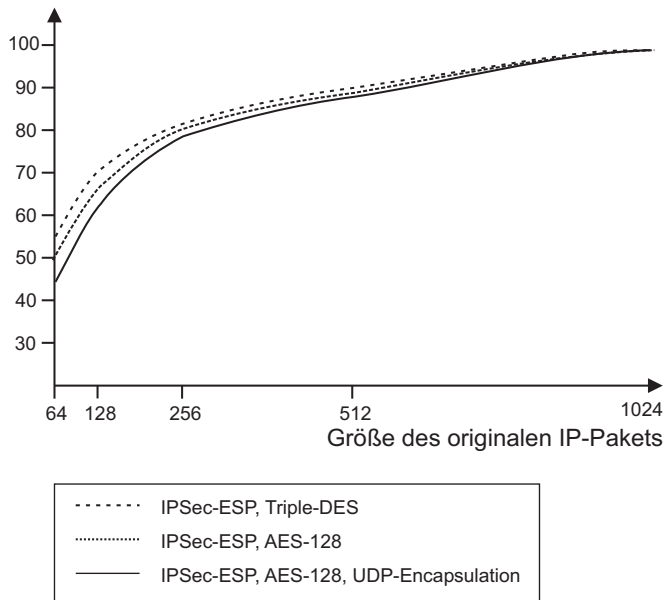


Abbildung 9.10: Durch den Overhead von Tunneling-Protokollen (hier: IPsec) wird die Nutzbandbreite in Abhängigkeit der Paketgröße teilweise sehr stark reduziert.

In SSL-VPN tritt ein Problem zutage, das auf der hohen Implementierungsebene von SSL beruht: Verfahren wie DiffServ können in der Regel nicht erkennen, von welcher Applikation die Pakete erzeugt wurden, und können sie auch nicht entsprechend behandeln.

Nehmen wir einmal den Fall, der, wie es aussieht, im größten Teil aller SSL-VPN auftritt, nämlich einen SSL-VPN-Client. Dort werden UDP- oder TCP-Pakete (oder im Fall von Thick-Clients auch IP-Pakete) auf ihrem Weg durch den TCP/IP-Stack abgefangen und dem SSL-Protokoll zugeführt. SSL übergibt seine Records anschließend wieder an das TCP-Protokoll. Nun kann aber DiffServ, das auf der IP-Ebene arbeitet, nicht mehr erkennen, welche Applikation das ursprüngliche Paket erzeugt hatte. Wie in Kapitel 7 beschrieben, interpretiert SSL keine Applikationsdaten und kann daher auch keine diesbezüglichen Informationen weitergeben. Die einzige für DiffServ relevante Information, die auf Ebene von SSL erzeugt werden könnte, wäre ohnehin nur die Portnummer – und die ist bereits für SSL selbst reserviert und vorgegeben.

Hier ist also vom Standard her absolut kein Mechanismus zur Signalisierung von QoS-Informationen an tiefere Protokollschichten vorgesehen. Eine entsprechende, mit privaten Portnummern realisierbare Implementierung würde also bedeuten, nicht mehr standardkonform und, das ist noch schlimmer, nicht mehr interoperabel zu sein!

Allerdings ist das in Praxis of kein Thema von großer Bedeutung, denn SSL-VPN sind nur für Remote Access einsetzbar, und dort spielt QoS oft eine untergeordnete Rolle, weil die Verbindungen zum Unternehmen in der Regel auf dem Best-Effort-Transport des Internets basieren und eine Markierung von IP-Paketen ziemlich sinnlos ist.

Beim Thema QoS in SSL-VPN gilt es, auch noch etwas anderes zu bedenken: Bisher ist in der Öffentlichkeit ein Thema gar nicht oder nur sehr zurückhaltend diskutiert worden, nämlich dass SSL das TCP-Protokoll zum Transport seiner Records benutzen muss, um überhaupt funktionieren zu können. Das erscheint auf den ersten Blick nicht weiter schlimm, aber in SSL-VPN geht die Tendenz ganz klar in Richtung SSL-VPN-Client, entweder als Thick Client (Transport von IP-Paketen über SSL) oder SOCKS-basierend, indem TCP- oder UDP-Pakete über SSL transportiert werden. Und diese Funktion sorgt unter anderem dafür, dass auch Datenströme von Applikationen über SSL und damit auch TCP übertragen werden, die Letzteres unbedingt vermeiden wollen.

Insbesondere Echtzeitanwendungen wie Sprache oder Video über IP kommen weder mit dem wiederholten Senden verlorener Pakete noch mit der TCP-Flusssteuerung klar und verwenden aus diesem Grund UDP. Eines dieser Protokolle ist zum Beispiel RTP (Real Time Protocol), das aufgrund seiner Charakteristik und Funktionalität speziell in interaktiven Applikationen wie zum Beispiel VoIP (Voice over IP) kein TCP verwenden kann.

Um dies zu verstehen, muss man sich kurz Sinn und Funktion von RTP ins Gedächtnis zurückrufen. RTP soll Daten in Echtzeit übertragen, was mit IP eigentlich unmöglich ist. Allerdings ist es erlaubt, dass auch das eine oder andere Paket verloren gehen darf und man eine gewisse, künstliche Grundverzögerung einbauen darf. RTP sequenziert die Übertragung und fügt Zeitstempel in die Pakete ein. Die Datenpakete von RTP bei VoIP-Anwendungen enthalten ein oder mehrere Sprach-Samples, die über UDP-Datagramme zum Empfänger gesendet werden. Falls ein Paket verloren geht, wird in der Regel das vorhergehende Sample einfach noch einmal verwendet.

Bei TCP sieht die Sache anders aus, hier wird so lange gewartet, bis der Sender das verlorene Paket noch einmal gesendet hat, egal ob noch weitere Pakete ankommen oder nicht. Dies kann etliche zehn oder hundert Millisekunden dauern, während der TCP keine Daten mehr an die Applikation gibt. Nach der erneuten Übertragung erhält die Anwendung nun die komplette Sequenz aus etlichen Samples. Üblicherweise werden aber nur die Samples wieder in Audiosignale umgewandelt, die zum entsprechenden Zeitpunkt auch erwartet wurden und innerhalb eines tolerablen Zeitfensters liegen – der Rest wird verworfen. Somit wurden aus einem verlorenen Sample gleich mehrere erzeugt. Das Ganze hört sich akustisch nach einer entsprechend langen Unterbrechung an.

Diese Effekte treten natürlich nur dann verstärkt auf, wenn relativ viele Pakete verloren gehen oder zu lange verzögert werden. Das ist momentan in vielen Netzen nicht zu erwarten, da zurzeit immer noch ein ziemliches Bandbreitenüberangebot existiert und die Breitbandzugangsdienste ebenfalls relativ zuverlässig arbeiten. Aber garantiert ist dieser Zustand, insbesondere im Internet, eben nicht, und bei der Planung eines Netzwerks sollte man eigentlich auf Nummer sicher gehen.

Ein anderes Phänomen kann auftreten, wenn es sich nicht um Echtzeitanwendungen, sondern um ganz normale Datenanwendungen handelt, die TCP verwenden. Ein SSL Client überträgt damit praktisch TCP über TCP. Beide Protokolle laufen dabei völlig unabhängig voneinander. Geht nun ein Paket verloren, dann erkennen das beide TCP-Stacks und fordern unabhängig voneinander Retransmits an. Das kann im Extremfall (mehrere verlorene Pakete) dazu führen, dass die Verbindung sich praktisch selbst abschnürt. Auch hier gilt, dass dieser Effekt nur dann störend wahrnehmbar wird, wenn relativ viele Pakete verloren gehen.

Auch der Nagle-Algorithmus, der in fast allen heutigen TCP-Implementationen zur Steigerung der Performance dafür sorgen soll, dass IP-Pakete möglichst auf ihre maximale Größe gefüllt werden, ist bei Real-Time-Anwendungen eher unerwünscht. Denn dort sollen die Pakete umgehend und in der Regel auch mit höchster Priorität versendet werden. Der Nagle-Algorithmus hingegen sammelt und verzögert damit so viele RTP-Pakete, bis ein IP-Paket maximal gefüllt werden kann. Damit wird praktisch eine feste, unzulässige Grundverzögerung erzeugt. In Windows sind die Applikationen dafür verantwortlich, ob deren TCP-Verbindung den Nagle-Algorithmus benutzt oder nicht. Globale Einstellungen gibt es, zumindest in Windows 2000 und Windows XP, nicht. Und die SSL- oder TLS-Standards erklären sich ausdrücklich für nicht zuständig, die Nutzdaten zu interpretieren, und können somit auch nicht abhängig von der Applikation gezielt ein TCP-Socket mit der Option NO_DELAY (kein Nagle-Algorithmus) oder DELAY (mit Nagle-Algorithmus) öffnen. Hier gilt es in der Praxis zu ermitteln, was die jeweiligen Betriebssysteme zu gegebener Zeit hergeben, um bei interaktiver Sprach- oder Videokommunikation über SSL keine unnötigen Verzögerungen ertragen zu müssen.

Qos in L2TP und L2TP/IPSec VPN

Obwohl der RFC3193 im Vergleich zu anderen RFCs noch relativ jung ist, ist es enttäuschend, dass wieder ein VPN-Protokoll das Licht der Welt erblickt hat, das aus Sicht seiner Entwickler anscheinend nicht mehr als E-Mail und unkritischen Datenverkehr transportieren soll.

9.9.1 DiffServ

Wer einmal die L2TP-Standards mit einem Editor öffnet und nach Begriffen wie „DiffServ“, „Quality of Service“ oder „DSCP“ durchsucht, der wird eine herbe Enttäuschung erleben, denn es findet sich nichts darin. Im originalen L2TP-Standard, dem RFC2661, ist das auch verständlich, denn L2TP wurde ursprünglich entwickelt, um in Dial-In-POPs von Carriern Anrufe zu aggregieren und diese in einem L2TP-Tunnel zum Endkunden zu transportieren. Und für diesen Tunnel konnte man relativ gut Traffic Engineering betreiben. Ein Kopieren des DSCP eines getunnelten IP-Pakets in den äußeren Tunnel-Header oder ein Mapping der PPP Class Number wäre ohnehin problematisch, da sehr viele IP-Pakete unterschiedlichster Verbindungen innerhalb des Tunnels übertragen werden.

Was ich persönlich aber nicht verstehe, dass auch in einem RFC3193 (aus Feder von Microsoft, Intel und Cisco!), der vor allem Microsofts L2TP/IPSec-Technologie offiziell machen sollte, ebenfalls nicht der kleinste Gedanke an QoS verschwendet wurde.

9.9.2 Einfluss von L2TP/IPSec auf die Dienstgüte

Der Einfluss von L2TP/IPSec auf die Dienstgüte hängt sowohl von L2TP und PPP als auch von IPSec ab. Die Implikationen von IPSec auf die Dienstgüte wurden in diesem Kapitel bereits diskutiert, so dass wir uns hier auf L2TP beschränken können.

Insgesamt finden im Vergleich zum IPSec-Tunnel-Mode (2) bei L2TP/IPSec wesentlich mehr (5) Protokolleinkapselungen statt. Dieser Prozess führt zu weiteren Verzögerungen, die sich eher negativ auf bestimmte Anwendungen auswirken.

Auch der Verbindungsaufbau einer L2TP/IPSec-Verbindung ist ein größerer Akt. Denn zusätzlich zu IPSec, das in der Microsoft-Lösung auch noch mit digitalen Signaturen arbeiten muss, müssen noch ein L2TP-Tunnel und eine PPP-Session ausgehandelt werden.

Es ist zwar kein protokollimmanentes Problem von L2TP, aber L2TP/IPSec benötigt auf den zentralen Komponenten (VPN-Router, Gateaways, Server) ziemlich viel an Ressourcen, was diese Systeme in ihren Grenzbereich treiben und weitere, in diesem Fall dann drastische Verzögerungen nach sich ziehen kann. Weder L2TP noch PPP sind zustandslose Dienste, beide benötigen entsprechenden Speicher und entsprechende Rechenzeit.

10 Access-Technologien

Nicht alles ist in einem VPN virtuell. Die IP-Pakete müssen über reale Netze, mit realem Übertragungsverhalten und realen Preisen übertragen werden.

In diesem Kapitel sollen kurz die verbreitetsten Access-Technologien mit ihren Möglichkeiten, Vor- und Nachteilen dargestellt werden. Wie die Paket innerhalb des Internets übertragen werden, kann der Anwender nicht beeinflussen, und somit ist das auch nicht Gegenstand dieses Kapitels.

Unter Access ist hier generell der Zugriff zum Internet gemeint. Technologien wie Frame Relay, ATM, SDH, POS usw. werden hierfür normalerweise nicht eingesetzt und hier auch nicht behandelt.

10.1 Mobile Technologien

10.1.1 GPRS

Global System for Mobile Communication (GSM)

Obwohl ursprünglich ein europäischer Standard, werden Mobilfunknetze auf Basis des Global System for Mobile Communication (GSM) mittlerweile in weit über 100 Ländern eingesetzt. Anfangs wurde der Frequenzbereich von 900 MHz zugelassen, später kam das 1,8 GHz-Band (USA 1,9 GHz) hinzu. Über ein Teilband können bei GSM per statischem Zeitmultiplexverfahren entweder 8 (Full Rate) oder 16 (Half Rate) Sprachkanäle und zusätzliche Informationen zur Fehlererkennung übertragen werden.

Da die Netzbetreiber nur eine begrenzte Menge von Funkfrequenzen zur Verfügung haben und die Sendeleistung der Mobilgeräte aus Gewichtsgründen (Akku-Betrieb) möglichst gering zu halten ist, hat man sehr viele kleine Funkzellen geschaffen. Damit kann man Frequenzen mehrfach benutzen, sofern sie nicht in benachbarten Funkzellen liegen, und die Sendeleistung kann ebenfalls relativ niedrig bleiben. Die Größe dieser Funkzellen liegt je nach System und Anzahl zu versorgender Teilnehmer im Bereich von wenigen hundert Metern bis über 30 km, typisch sind jedoch Zellengrößen von durchschnittlich 5 bis 10 km. Solch einen Netzverbund bezeichnet man als zellulares Netz. Wichtig dabei ist, dass die Funkzellen über ein terrestrisches Netzwerk (SDH, ATM) miteinander verbunden sind und sich bewegende Teilnehmer von einer zur nächsten Funkzelle übergeben werden können (Handover), ohne dass die Verbindung dabei unterbrochen wird.

Die Komponenten, aus denen typische GSM-Netze heute bestehen, und deren Zusammenwirken sind vereinfacht in Abbildung 10.1 dargestellt. Man trennt hier zwischen dem Funkbereich (BSS, Base Station Subsystem) und dem Festnetzbereich (NSS, Network Subsystem). Man sieht auch sofort, dass hier ein deutlich höherer Aufwand als im normalen Telefonfestnetz betrieben werden muss.

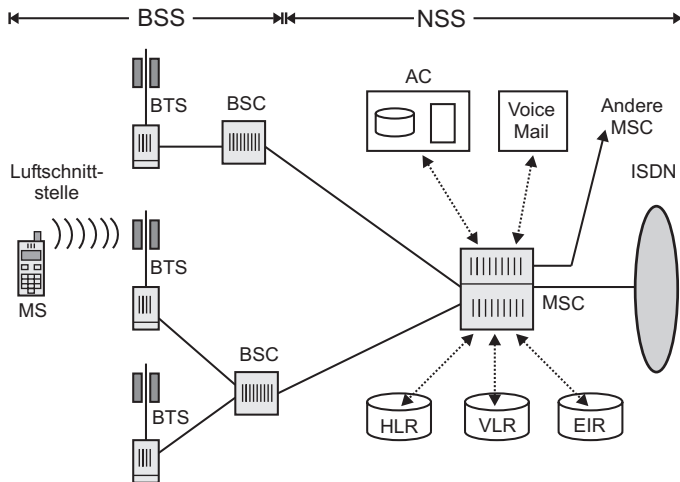


Abbildung 10.1: Der prinzipielle Aufbau eines GSM-Netzwerks

Im drahtlosen Zugangsbereich findet man, gewissermaßen als einzige sichtbare GSM-Komponente, zahlreiche BTS (Base Transceiver Station), die für die digitale Funkübertragung zu den mobilen Stationen, also den Handys, zuständig sind. Aufgrund der relativ geringen Größe von GSM-Funkzellen findet man mittlerweile sehr viele solcher Stationen, entweder weithin sichtbar an Sendemasten oder gut getarnt in Kirchtürmen versteckt. Eine oder mehrere Base Transceiver Stations werden von einem Base Station Controller (BSC) gesteuert. Mehrere BSC wiederum werden von einem Mobile Switching Center (MSC) gesteuert, das als Vermittlungsstelle für Mobilteilnehmer untereinander oder den Übergang in andere Netze verantwortlich ist. Eine weitere wichtige Funktion, die BTS, BSC und MSC übernehmen, ist das oben erwähnte Handover, das beim Wechsel eines aktiven Teilnehmers in eine andere Funkzelle notwendig wird. Je nach Wechsel kann dies sehr komplexe Aktivitäten aller beteiligten Funktionseinheiten auslösen, insbesondere wenn ein Wechsel in den Bereich einer anderen MSC stattfindet. Im einfachsten Fall wird der Teilnehmer von einer BTS an die benachbarte „weitergereicht“.

Weiterhin sind Sicherheitsfunktionen wie Benutzerauthentifizierung und Schutzmaßnahmen gegen unberechtigtes Abhören der Funkverbindungen über das Authentication Center (AC) oder das Pflegen „schwarzer Listen“ gestohlener Mobilteile durch das EIR (Equipment Information Register) hinzugekommen. Zur Verwaltung der Benutzerinformationen werden HLR (Home Location Register) eingesetzt, in denen alle relevanten Daten wie Rufnummer, erlaubte Dienste und Zugriffsberechtigungen der Teilnehmer verwaltet werden. Dieses Register wird in einem Mobile Switching Center gepflegt. Das VLR (Visitor Location Register), es ist meistens in der MSC integriert, enthält gewissermaßen ein temporäres HLR von Teilnehmern, die sich vorübergehend im Bereich eines MSC befinden, aber nicht dessen HLR zugeordnet sind. Diese Daten werden bei Bedarf vom jeweiligen HLR des Teilnehmer angefordert.

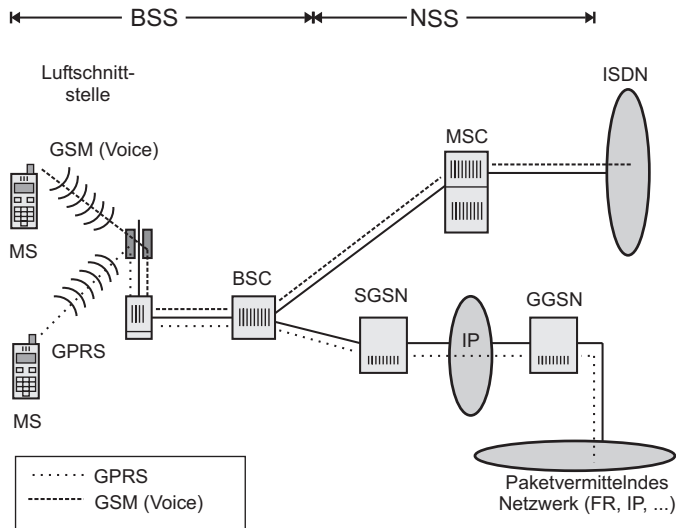


Abbildung 10.2: GPRS benutzt einen Teil der GSM-Infrastruktur.

General Packet Radio Service (GPRS)

Mit diesem Verfahren sind in momentan existierenden GSM-Netzen Geschwindigkeiten von theoretisch bis zu 53,6 KBit/s möglich, jedoch werden diese in der Praxis meist nicht erreicht. Von unabhängigen Fachmagazinen durchgeführte Tests und Messungen berichten jedoch von einem Mittelwert von über 32 KBit/s bei einigen Carriern, was einen durchaus brauchbaren Wert darstellt.

GPRS sollte auch keineswegs als Zwischenlösung bis zur Verfügbarkeit von UMTS betrachtet werden, denn außerhalb der Gebiete, die zukünftig von UMTS versorgt werden, stellt es auch in Zukunft die einzige brauchbare drahtlose Datenübertragungslösung für den mobilen Einsatz dar!

GPRS basiert im Funkbereich auf der vorhandenen GSM-Technologie und den damit aufgebauten Netzen. Die diesen gegenüber jedoch deutlich höhere Geschwindigkeit erzielt GPRS durch die Bündelung von mehreren Funkkanälen für eine Datenverbindung. Falls in einer Zelle durch hohe Lasten nicht genügend Kanäle zur Verfügung stehen oder die Qualität stark eingeschränkt ist, sinkt in Folge auch die Bandbreite einer GPRS-Verbindung.

Der Unterschied zum (Sprach-)GSM liegt in der weiteren Verarbeitung der Signale hinter dem so genannten Base Station System. Normale GSM-Sprachverbindungen werden vom Base Station Subsystem (BSS) über das Mobile Switching Center (MSC) entweder zu anderen Mobilteilnehmern oder in das leitungsvermittelnde Festnetz (Circuit Domain) des Carriers geleitet. Im Fall des GPRS jedoch werden die Informationen vom BSS (Base Station Subsystem) statt zum MSC zu einem Serving GPRS Support Node (SGSN) geführt und dort mittels eines speziellen Tunneling-Protokolls, dem GTP (GPRS Tunneling Protocol), zum einem Gateway (GGSN, Gateway GPRS Support Node) zu paketvermittelnden Netzen geführt. Der Verlauf dieses Informationsflusses ist in Abbildung 10.3 anhand eines Schichtenmodells veranschaulicht.

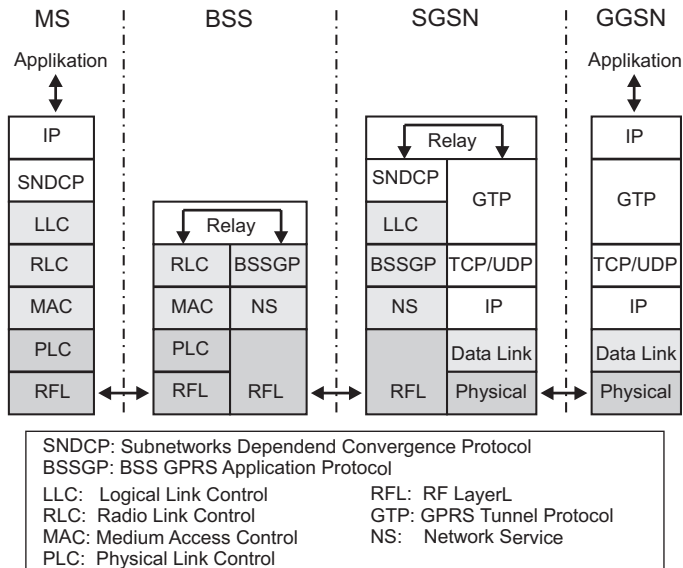


Abbildung 10.3: Der Transport von IP-Paketen aus einem GPRS-Device in ein IP-Netz erfordert einigen Aufwand.

GPRS ist ein verbindungsloser Dienst, der mit der so genannten Context Reservation arbeitet. Hierbei werden im Gegensatz zur Resource Reservation nur dann Übertragungsressourcen verwendet, wenn auch Daten übertragen werden. So kann auf der einen Seite ein mengenbezogene Abrechnung erfolgen, und andererseits können die Sendeeinrichtungen aller Mobilstationen benutzt und so die Effizienz des Gesamtsystems signifikant erhöht werden.

10.1.2 UMTS

Die Entwicklung von UMTS wurde schon vor über 14 Jahren, 1992, gestartet um, so das Ziel, eine weltweit einheitliche Mobilkommunikation, zu ermöglichen. Anders als GSM, das dieses Ziel leider verfehlt hat, wie man weiß, seit man mit seinem europäischen Handy in den USA oder Japan telefonieren wollte, sollte dieses System der dritten Generation (3G) anders sein: international und gleichermaßen für Sprache, bewegte Bilder und Datenkommunikation geeignet. Als Resultat wurden vom 1998 gegründeten 3rd Generation Partnership Project (3GPP), bestehend aus den Standardisierungsgremien ANSI-T1 (USA), ARIB (Japan), TTC (Japan), TTA (Korea) und ETSI (Europa), Spezifikationen zu UMTS veröffentlicht – allerdings ohne jeglichen bindenden Charakter. Bindenden Charakter haben nach wie vor nur lokale Spezifikationen, und deren Hauptinteresse sind leider lokale Gegebenheiten und der Schutz lokaler Unternehmen und deren Technologien, von politischen Einflussnahmen ganz zu schweigen.

Um es kurz zu machen: Es gibt keinen weltweit einheitlichen UMTS-Standard in der Art, dass man hier ein UMTS-Handy kauft und in den USA damit telefonieren kann. Es gibt unverbindliche Empfehlungen der 3GPP, und es gibt verschiedene lokale Standards, nach denen Systeme implementiert werden. So wird zum Beispiel in Japan UMTS, im

restlichen fernen Osten UWC-136 und in den USA CDMA2000 eingesetzt. Die Unterschiede sind oft nur klein, aber meist ausreichend genug, damit die Systeme mehr untereinander interoperabel sind.

Ein wesentlicher Unterschied zwischen GSM und UMTS ist die so genannte Luftschnittstelle zwischen Mobilteil und Basisstation. Während GSM mit dem TDMA-Verfahren (TDMA, Time Division Multiple Access) arbeitet, bei dem mehrere Mobilteile zeitversetzt auf der gleichen Frequenz senden und empfangen, benutzt man bei UMTS das WCDMA-Verfahren (WCDMA, Wideband Code Division Multiple Access), bei dem die Daten für alle Teilnehmer innerhalb einer Funkzelle gleichzeitig übertragen werden. Damit die Teilnehmer aus diesem Mischmasch die für sie bestimmten Informationen herausfiltern können, werden sie vor dem Senden mit einem speziell codierten Signal überlagert, das für jedem Empfänger unterschiedlich ist. Dieses Signal, die so genannte Chipsequenz, muss vor der Übertragung zwischen Sender und Empfänger ausgehandelt worden sein.

Da die UMTS-Nutzbandbreite 5 MHz, gegenüber nur 0,2 MHz bei GSM beträgt, kann man sehr viel höhere Übertragungsraten bis hinein in den MBit/s-Bereich erzielen. Alle Teilnehmer in einer Zelle teilen sich dynamisch die zur Verfügung stehende Bandbreite, die allerdings mit steigender Entfernung zur Basisstation absinkt.

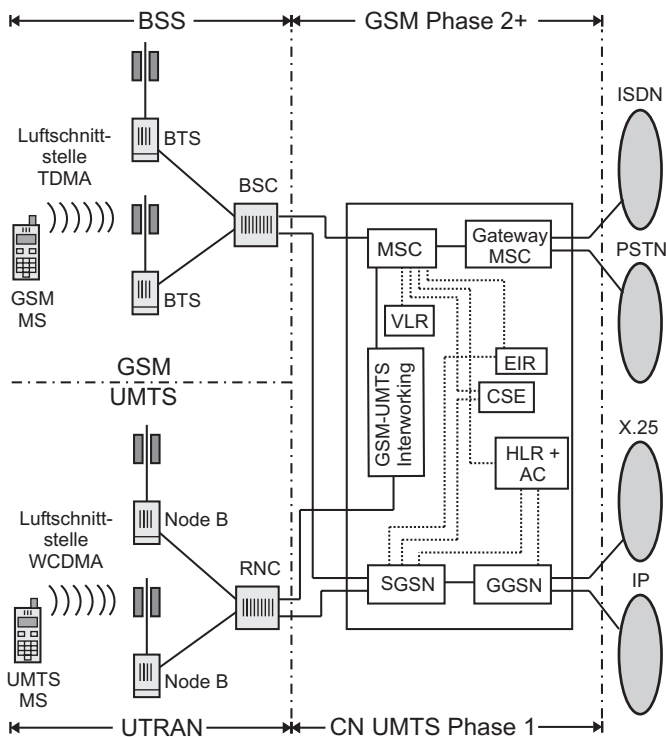


Abbildung 10.4: In der ersten Phase wird in UMTS nur die Luftschnittstelle geändert, der Rest der GSM-Infrastruktur kann noch mitbenutzt werden.

Allerdings ist die Anordnung der Basisstationen eine andere als in GSM-Netzen, denn UMTS unterscheidet vier verschiedene Zonen:

Zone 1: Diese kleinste Zone, in der so genannte Pico-Zellen mit einem Radius von wenigen zehn Metern eingesetzt werden. Sie stellen gewissermaßen die „Hot Spots“ von UMTS dar. In dieser kleinen Zelle wird die maximale Übertragungsgeschwindigkeit von 1,92 MBit/s erreicht. Die Geschwindigkeit, mit der sich ein Teilnehmer bewegen darf, beträgt 10 km/h, was auch der Situation in Bürogebäuden, Wartehallen oder auf einem Campus angemessen ist.

Zone 2: Hier arbeiten Mikrozellen mit Radien von mehreren hundert Metern bis wenigen Kilometern, wobei erhöhte Anforderungen durch die Mobilität der Teilnehmer entstehen. Die erlaubten Geschwindigkeiten gehen bis zu 120 km/h. Wer allerdings innerorts so schnell fährt, kann im günstigsten Fall nur mit einer Übertragungsrate von 384 KBit/s rechnen, mit 10 km/h werden maximal 1,92 MBit/s erreicht. Ortschaften und Teilbereiche von großen Städten fallen in den Bereich, den diese Zone abdecken soll.

Zone 3: Zur Abdeckung von kleineren Ortschaften, dünn besiedelten Gebieten und vor allem Verkehrswegen werden die Makrozellen der Zone 3 eingesetzt, die vom Radius (bis etwa 20 km) her am ehesten mit einer GSM-Funkzelle vergleichbar sind. Aufgrund der Abdeckung von Verkehrswegen, auf denen sich Straßen- und Schienenfahrzeuge sehr schnell fortbewegen können, werden Geschwindigkeiten bis 500 km/h unterstützt, in diesem Fall allerdings nur noch mit maximal 144 MBit/s Übertragungsgeschwindigkeit.

Zone 4: Für ganz abgelegene und fast unbesiedelte Gegenden gibt es die Zone 4 mit globalen Zellen, auch als Hyper- oder Umbrella-Zellen bezeichnet. Der Zellradius liegt im Bereich von einigen zehn bis zu einigen 1000 Kilometern, wenn, für letzteren Fall, die Basisstation zuvor in eine geostationäre Umlaufbahn geschossen wurde. Aufgrund der Tatsache, dass in unbesiedelten Gegenden natürlich auch nur wenige Kunden existieren und eine geostationäre Basisstation gleichzeitig sündhaft teuer ist, ist anzunehmen, dass Zonen mit 1000-km-Zellen vorerst Fiktion bleiben.

Die verschiedenen Zonen können sich auch überlappen, denn es macht durchaus Sinn, einen gewissen Bereich mit einer Makrozelle abzudecken, jedoch kleine Teilbereiche, in denen ein sehr hohes Kommunikationsaufkommen existiert, mit einer oder mehreren Pico-Zellen zu versorgen.

Aufgrund der Tatsache, dass UMTS sich vor allem im Bereich der Luftschnittstelle von GSM unterscheidet, ist es möglich, die vorhandene Festnetz-Infrastruktur von GSM und GPRS weiter bzw. mitzubnutzen. Aus diesem Grund sieht die Architektur von UMTS zuerst einmal gar nicht so viel anders aus als die von GSM. Man unterscheidet zwischen dem User Equipment (UE), dem UMTS Terrestrial Radio Access Network (UTRAN) und dem Core Network (CN). Diese Bereiche sind durch hinreichend definierte Schnittstellen miteinander verbunden und ermöglichen es, ein UMTS-Netzwerk schrittweise aufzubauen und vor allem die vorhandene GSM-Infrastruktur zu benutzen. Vor allem im Core Network sind praktisch alle Funktionen für beide Netze nutzbar.

So wird den UMTS auch phasenweise realisiert, indem in der ersten Stufe (UMTS Phase 1) zuerst einmal die neue Luftschnittstelle eingeführt wird. Dafür werden neues Benutzer-equipment und eine neue Technologie (UTRAN) im Bereich der Basisstationen notwendig.

Das bestehende Core Network wird weiter benutzt (Abbildung 10.4), ergänzt um Systeme zum so genannten Interworking zwischen UMTS- und GSM-Schnittstellen zum Mobile Switching Center (MSC). Auch die vorhandenen Systeme zur Authentifizierung (AC), Benutzerverwaltung (HLR, VLR) und Equipmentregistrierung (EIR) können von GSM und UMTS gemeinsam benutzt werden.

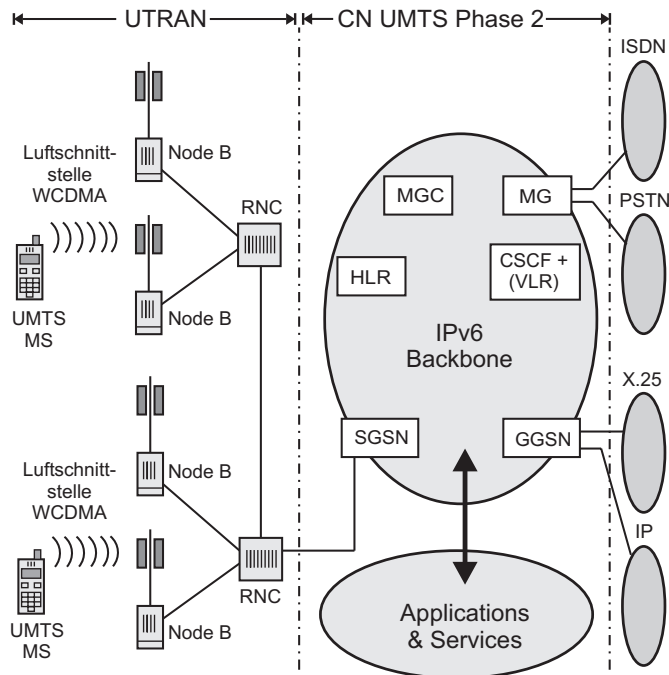


Abbildung 10.5: Der Endausbau von UMTS (Phase 2) baut auf IPv6 und Protokolle wie SIP.

Das Core Network wird erst in der UMTS Phase 2, der so genannten Release 4/5, verändert und an seine neue Funktion als Netzwerk für UMTS-Multimediasdienste angepasst. Innerhalb des Core Networks wird IP als universelles Protokoll für Sprache, Daten und alle weiteren Dienste eingesetzt. Das spiegelt sich auch in der Vermittlungstechnik wider, hier werden schnelle Layer-3-Switches und -Router eingesetzt. Auch der Rufaufbau erfolgt in UMTS Phase 2 mit dem SIP (Session Initiation Protocol), nicht mehr mit herkömmlicher Signalisierung. SIP wird im CSCF (Call State Control Function) bearbeitet, das auch die Funktion des Visitor Location Registers (VLR) aus der GSM-Welt mit übernimmt.

Die „alte Welt“, also das leitungsvermittelnde Festnetz, wird über Media Gateways (MG) gesteuert von Media Gateway Controllern (MGC) mit dem IP-Backbone verbunden. Natürlich versteht es sich von selbst, dass als IP-Protokoll im Core Network nur die IP-Version 6 in Betracht kommt!

Auch im Bereich neuer Dienste ist einiges geplant, nicht zuletzt auch so genannte Location Based Services, also Dienste, die den augenblicklichen Standort eines Benutzers mit speziell darauf abgestimmten Funktionen verknüpfen können.

10.1.3 WLAN

Mittlerweile gibt es kaum noch größere Bahnhöfe, Hotels oder andere Orte – mittlerweile sogar Flugzeuge –, an denen Geschäftsreisende mit ihren Mobilsystemen per Wireless LAN über einen so genannten Hot Spot nicht mit dem Internet verbunden werden.

Wireless LANs arbeiten je nach Standard im 2,4- und 5-GHz-Band. Leider sind diese Frequenzbänder nicht ganz unproblematisch. Im 2,4-GHz-Band darf sich praktisch jeder tummeln – und tut dies auch. Und im 5-GHz-Band arbeiten viele Radaranlagen, so dass jedes System beim Booten erst einmal eine Zeit lang den Frequenzbereich auf Radarsender abhören muss, bevor es selbst in Betrieb geht. Die Kanäle, auf denen innerhalb der Bänder gearbeitet werden darf, sind je nach Land unterschiedlich freigegeben.

	Frequenzband	Datenrate (brutto)	Kompatibel zu

Tabelle 10.1: Momentan sind diese drei oberen WLAN-Standards aktuell, IEEE 802.11n ist für 2006 geplant.

In WLAN Hot Spots wird der Infrastrukturmodus, der ausschließlich Kommunikation zu den Access Points zulässt, eingesetzt. Meist handelt es sich um Multiband-Infrastrukturen, welche die Standards IEEE 802.11 a, b und g unterstützen.

In der Regel erfolgt in den Hot Spots weder eine Authentifizierung noch eine Datenverschlüsselung auf der Luftschnittstelle. Normalerweise muss man sich auf einer Website des Betreibers anmelden und entsprechende Zeitkontingente kaufen. In Zukunft wird auch eine Authentifizierung und darüber auch eine Abrechnung per SIM-Karte geben, hier ist ein Standard (EAP-SIM) in Arbeit.

10.1.4 Satellitenverbindungen

Vorab gleich ein Wehrmutstropfen: Fast alle Satellitenverbindungen sind Einbahnstraßen sind nicht rückkanalfähig, und es können somit nur Daten vom Provider zum Kunden übertragen werden. Die andere Richtung, also gewissermaßen das Anfordern dieser Daten, erfolgt über andere Wege wie ISDN, Modems usw. Mittlerweile bieten die ersten Provider auch rückkanalfähige Satellitenverbindungen an, jedoch erfordern diese spezielle Antennen und Transceiver (Sender und Empfänger), haben jedoch den Vorteil, völlig unabhängig von der Versorgung mit Telefon- und Datenleitungen zu sein.

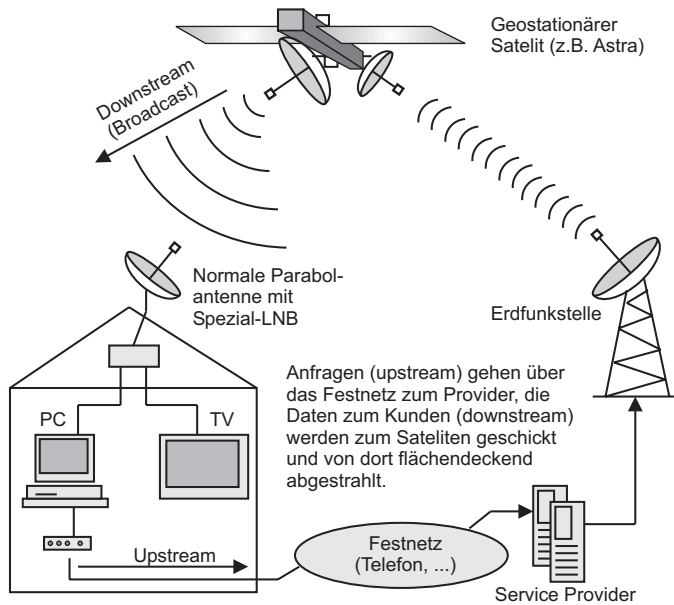


Abbildung 10.6: Unidirektionale Satellitenverbindungen bedingen immer noch eine terrestrische Verbindung zum ISP.

Die eingesetzte Technik ist recht einfach beschrieben: Der Provider sendet die Daten, die zu den Endkunden übermittelt werden sollen, zum Satelliten, dieser setzt sie um und strahlt sie zurück zur Erde. Dies geschieht natürlich flächendeckend, man benutzt hierfür normale Fernmeldesatelliten wie „Eutelsat“ oder „Astra“. Bei Einsatz eines zweiten Digital-LNBs kann man eine bereits vorhandene Satellitenantenne mit benutzen, sofern sie auf den richtigen Satelliten ausgerichtet ist. Die Bandbreiten bewegen sich im Bereich von bis zu 4 MBit/s.

Allerdings handelt es sich um eine Broadcast-Technologie, das heißt alles, was die verschiedenen Nutzer an Daten anfordern, wird sequenziell über dieses eine Medium abgestrahlt. Der PC auf der Empfängerseite muss sich dann über entsprechende Treiber das herausuchen, was auch für ihn bestimmt ist. Leider kann dieses Verfahren zu relativ hohen Verzögerungen führen. Bedingt durch die Übertragungstechnik hat man bereits mindestens 200 Millisekunden, insgesamt muss man mit 300 bis 500 ms Delay rechnen. Damit verbieten sich manche Applikationen wie Sprach- oder Videoanwendungen von selbst.

Zu all den Qualitätsproblemen kommen noch erhebliche Sicherheitsprobleme hinzu, denn die Informationen, die für einen bestimmten Kunden bestimmt sind, werden natürlich von den künstlichen Erdtrabanten wie zum Beispiel dem „Astra“-Satelliten flächendeckend über Europa verteilt – ein Albtraum für viele. Also muss, meist vom Kunden selbst, zusätzlich noch aufwändige Sicherheitstechnik, insbesondere eine zuverlässige Datenverschlüsselung, eingesetzt werden.

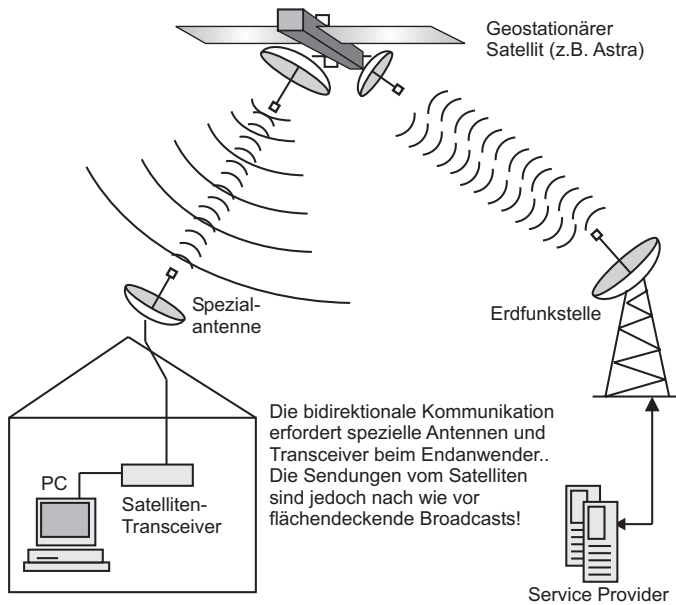


Abbildung 10.7: Erst eine bidirektionale Satellitenverbindung macht wirklich mobil.

Eine Satellitenverbindung setzt man am besten nur dann ein, wenn es anders überhaupt nicht geht. Technische Limitierungen hinsichtlich der Verzögerungen von mehreren hundert Millisekunden, mangelnde Sicherheit und die Tatsache, dass die meisten Dienste keinen Rückkanal anbieten, machen das Ganze zu einer Lösung, zu der man nur notgedrungen greift.

10.2 Ortsgebundene Technologien

10.2.1 ADSL

Digital Subscriber Line (DSL) ist eine High-Speed-Technologie, die auf der so genannten Last Mile, also dem Zugang zum Endteilnehmer, mit herkömmlicher zweiadriger Kupferdraht-Technologie, wie sie bei ISDN oder analoger Telefonie eingesetzt wird, auskommt. Lediglich das Anschlussorgan beim Endkunden und die Gegenstelle in der Ortsvermittlungsstelle (OVSt) werden ausgetauscht bzw. ergänzt, größere kostenaufwändige Kabelarbeiten sind nicht nötig. Mittlerweile gibt es verschiedene DSL-Varianten, von denen wir aber aufgrund ihrer Bedeutung nur die folgenden zwei, nämlich ADSL und SDSL, besprechen werden.

ADSL steht für Asymmetric DSL. Asymmetrisch bedeutet hier, dass mit unterschiedlichen Geschwindigkeiten vom Carrier zum Endkunden (downstream) und umgekehrt (upstream) gearbeitet wird. Die maximal möglichen Geschwindigkeiten sind dabei 6,144 MBit/s downstream und 640 KBit/s upstream, allerdings hängen diese Geschwindigkeiten von verschiedenen Parametern, insbesondere aber von dem Durchschnitt und der Länge des Kupferkabels zwischen dem ADSL-Modem und dem DSLAM (DSL Access Multiplexer) in der Ortsvermittlungsstelle, ab.

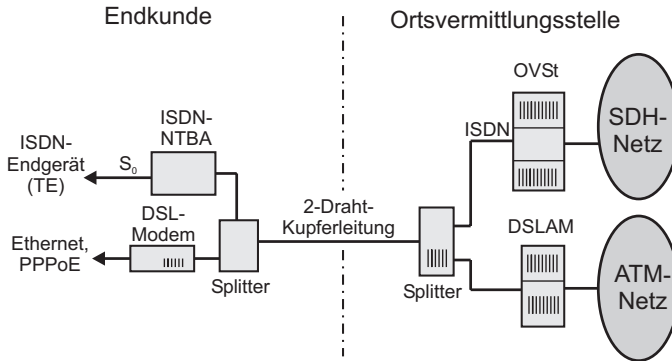


Abbildung 10.8: ADSL im Überblick. Telefonie und ADSL werden gemeinsam über ein zweiadriges Kupferkabel übertragen.

Die Minimalkonfiguration (lt. Standard) mit 1,536 MBit/s downstream und 16 KBit/s upstream ist üblicherweise bis zu einer Entfernung von 4,6 km garantierbar.

Downstream-Geschwindigkeit	Downstream-Multiplex-Hierarchie

Tabelle 10.2: Die verschiedenen Downstream-Geschwindigkeiten von ADSL

ADSL ist auf der einen Seite eine digitale Modemtechnologie, die auf relativ neuen Algorithmen und entsprechender Hardware in Form hochspezialisierter Chipsätze basiert und dadurch eine so hohe Geschwindigkeit auf einem Medium wie zum Beispiel einem nicht abgeschirmten, zweiadrigen Kupferkabel ermöglicht. Auf der anderen Seite ist ADSL geradezu ein Paradebeispiel für Technologie-Recycling, denn ansonsten greift man auf Altbewährtes zurück:

PPP mit seinen ganzen Funktionen eignet sich hervorragend zum Einkapseln von Netzwerkprotokollen, der Unterhaltung von Punkt-zu-Punkt-Verbindungen, zur Authentifizierung und zum Accounting.

Ethernet ist eine billige, bestens eingeführte Netzwerktechnologie, die man auf Endbenutzerseite geschickt einsetzen kann.

ATM ist die Technologie der Wahl im Backbone des Carriers, also sollte die Verbindung zwischen Modem und dem Zugangskonzentrator (BSN, Broadband Service Node) ATM sein.

Das Ergebnis ist PPPoE (PPP over Ethernet) über ATM/AAL5 zwischen Endgerät, Modem und BSN. Man sieht: Das einzige Neue an ADSL ist die Modemtechnologie zur Breitbandübertragung über die vorhandene Zweidraht-Telefonleitung.

Eine Standardisierung ist mittlerweile durch das ANSI (T1.413) erfolgt, ebenso wurde internationalen und europäischen Gegebenheiten in Form einer Ergänzung durch ITU und ETSI Rechnung getragen. Mittlerweile hat sich auch als Modulationsverfahren DMT (Discrete Multi Tone) weitgehend durchgesetzt. Dieses Verfahren unterteilt das Gesamtübertragungsband in viele kleine Teilbereiche (Subcarrier). Dieser so genannte Full Rate ADSL Standard ist in der G.992.1-Empfehlung der ITU beschrieben.

ADSL stellt neben den beiden Kanälen in beide Richtungen noch einen analogen (POTS, Plain Old Telephone Service) oder einen digitalen (ISDN) Telefondienst zur Verfügung. Der Telefonkanal wird durch einen einfachen, passiven Frequenzfilter, den so genannten Splitter, ein- und ausgekoppelt, wodurch auch beim Ausfall des ADSL-Modems oder des DSLAMS in der OVSt zumindest noch Telefonie möglich ist.

Dem Endanwender präsentiert sich ADSL entweder in Form eines ADSL-Modems, das ihm sein Anbieter überlässt, oder nur in Form eines DSL-Splitters, an die er selbst geeignete Geräte anschließen muss. Im ersten Fall hat der Kunde üblicherweise ein einziges Gerät, an dem er auf der einen Seite eine Ethernet-Schnittstelle sowie optional einen Telefonanschluss, je nach Vertrag analog oder digital (ISDN), und auf der anderen Seite einen Anschluss für das DSL-Kabel zur Verfügung hat. Im zweiten Fall sollte der DSL-Provider dem Kunden eine klare Spezifikation zur Verfügung stellen, anhand der er sich passende DSL-Modems beschaffen kann.

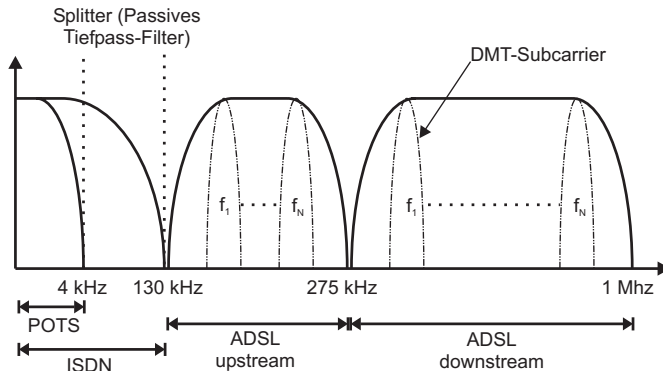


Abbildung 10.9: Das von ADSL und der Telefonie benutzte Frequenzspektrum

Die DSL-Technologie ermöglicht auch das „Entbündeln“ der Kupferleitung, die somit von zwei verschiedenen Carriern gleichzeitig ohne zusätzliche Verkabelung zum Kunden hin benutzt werden kann. Der Splitter beim Kunden und in der OVSt ermöglicht, dass ISDN oder analoge Telefonie und DSL von unterschiedlichen Unternehmen ein- und ausgekoppelt werden können. Die Deutsche Telekom ist verpflichtet, anderen Anbietern Platz in ihren Räumen für deren DSLAMs zur Verfügung zu stellen (Co-Locations-Prinzip) und auch die Ein- und Auskopplungen an den Splitttern (Line Sharing) vorzunehmen.

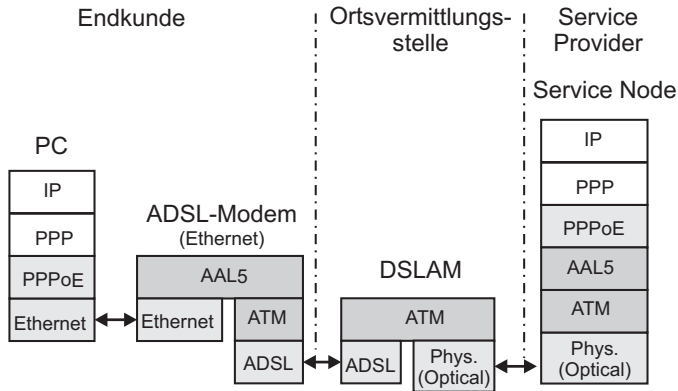


Abbildung 10.10: ADSL benutzt viele bekannte Technologien wie PPP, ATM und Ethernet.

ADSL hat aber auch einige Nachteile, insbesondere im Geschäftsbereich. Man bekommt z.B. dynamische IP-Adressen zugewiesen, die der Provider zu allem Überfluss auch noch laufend ändern darf (und dies auch tut). Dadurch sind die DSL-Endsysteme nicht über eine fest zugeordnete IP-Zieladresse erreichbar. Somit ist ein direkter Ersatz für die teuren Festverbindungen nicht möglich, und es bleibt leider nur noch der Internet-Zugriff als brauchbares DSL-Einsatzszenario übrig. Einen Ausweg bieten die VPN-Technologien, mit denen über das Internet virtuelle Punkt-zu-Punkt-Verbindungen aufgebaut werden. Allerdings haben auch VPNs mit dem Problem zu kämpfen, dass die Provider meist einmal pro Tag die Verbindung zwangsweise beenden und beim Wiederaufbau andere IP-Adressen per PPPoE zuweisen.

Eine weitere Einschränkung des Einsatzgebietes von ADSL kann auch in dessen Asymmetrie liegen, denn seine Geschwindigkeitsverhältnisse sind typisch für Websurfer, denn die Seitenanforderungen an Webserver sind von der Datenmenge sehr viel geringer als das, was der Webserver daraufhin zurückliefert. Bei anderen Verkehrsprofilen, in denen auch die Upstream-Bandbreite sehr groß sein muss, zum Beispiel bei konvergenten Netzen und VoIP, ist ADSL keine gute Wahl.

10.2.2 SDSL

SDSL (Symmetric DSL) ist eine DSL-Variante, in der die Übertragungsgeschwindigkeit in beide Richtungen gleich groß ist, daher auch der Name. Technologisch bedingt kann hier kein Telefondienst mehr integriert werden, SDSL ist also ein reiner Datendienst. Allerdings können Geschwindigkeiten bis zu 2,048 MBit/s in beiden Richtungen angeboten werden, natürlich nur bei guten Verhältnissen hinsichtlich Länge und Qualität der Zuleitung.

Viele Anbieter bieten die Option, dann natürlich auch zu geringeren Preisen, niedrigere Geschwindigkeiten einzusetzen, die nach Bedarf des Kunden jedoch später bis zu den vollen 2,048 MBit/s erhöht werden können.

Allerdings hat SDSL seinen Preis, denn im Paket mit festen IP-Adressen und entsprechendem Service kommt man schnell zu Preisen, die man auch für Festverbindungen oder Frame-Relay-Dienste berappen muss. Allerdings ist das nicht technologisch bedingt, sondern auch wieder, wie bei ADSL, eine kaufmännische Entscheidung der Anbieter. Ein Provider allerdings, der überhaupt keine Festverbindungen oder Frame-Relay-Dienste in seinem Portfolio hat, könnte SDSL durchaus günstiger anbieten. SDSL wird inzwischen von etlichen Carriern angeboten, so auch außer der DTAG von einer Vielzahl von Providern, die jedoch oft nur lokal oder in einer Reihe von Städten oder Ballungszentren tätig sind.

Die meisten SDSL-Angebote umfassen auch Equipment zur Terminierung von SDSL, so dass die Verbindung an den Endkunden meist per Ethernet erfolgen kann.

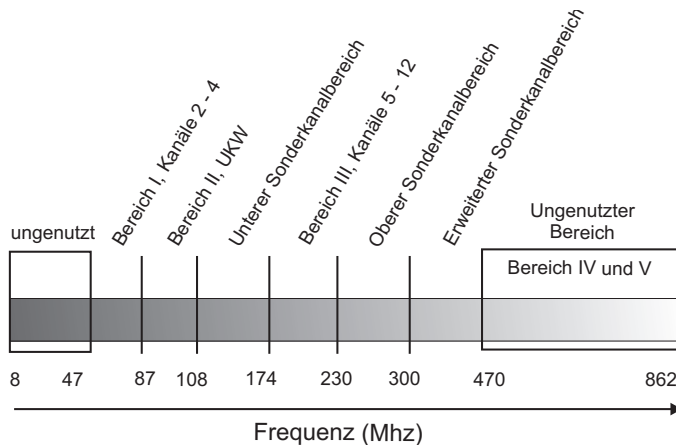


Abbildung 10.11: Die Breitbandkabel bieten noch ausreichend Kapazität für Datenkanäle.

10.2.3 Breitbandkabel

Es stellt sich jetzt vielleicht dem einen oder anderen die Frage, warum ein simples, nicht abgeschirmtes Kupferkabel zwischen der Telefondose zu Hause oder im kleinen Büro und der am nächsten gelegenen Ortsvermittlungsstelle durch DSL zu einem Hochgeschwindigkeitsmedium gemacht werden soll, wenn in Deutschland doch im Jahr 2002 bereits 56% aller Haushalte über einen Breitbandanschluss verfügen. Denn in all diesen Haushalten existiert ein Zugang zum TV-Kabelnetz.

Von den elektrischen Eigenschaften her ist solch ein Kabelnetz für Frequenzen bis zu über 820 MHz geeignet, bei Einsatz entsprechender Modulations- und Übertragungsverfahren kommt man sehr schnell in den Gigabit-Bereich, falls man dieses Medium auch zur Datenübertragung nutzen möchte. Allerdings teilen sich mehrere Anschlüsse ein Kabelsegment, so dass pro Teilnehmer maximal Geschwindigkeiten bis zu 2 MBit/s realistisch sind – denn fernsehen will man ja auch noch.

Für den Kunden sieht die Sache recht einfach aus: Er bekommt von dem Provider ein so genanntes Kabelmodem, schließt es an eine Antennensteckdose an und kann dann loslegen. Meist bieten die Modems eine Ethernet-Schnittstelle, an die man sein Daten-Equipment anschließen kann.

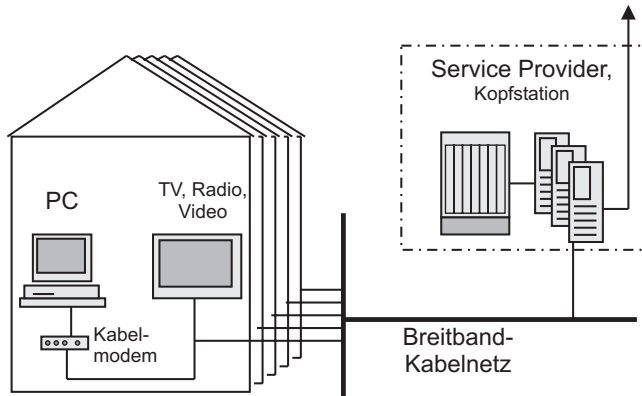


Abbildung 10.12: Die Kabelmodem-Technologie bringt Ethernet vom Provider zum Endteilnehmer.

Auf den meisten Breitbandkabeln sind zurzeit nicht ganz 50% der verfügbaren Bandbreite durch Radio- und TV-Kanäle belegt, der Rest ist ungenutzt. Allerdings beschert ein Entwurfskriterium des Kabelnetzes, das zum damaligen Zeitpunkt durchaus sinnvoll war, den Providern heute einiges an technischen Herausforderungen. Das Kabelnetz ist nämlich von Haus aus nicht rückkanalfähig, es kann also Signale nur in eine Richtung transportieren, und zwar in Richtung der zu versorgenden Haushalte. Der Rückweg, also die Einspeisung von Signalen von Seiten der Endteilnehmer in das Netz, wurde durch technische Maßnahmen blockiert. Das ist natürlich für bidirektionale Daten- oder auch Sprachkommunikation das absolute Aus, hier muss die Technik an die neuen Bedürfnisse angepasst werden, das Netz muss also rückkanalfähig gemacht werden.

Die technischen Gegebenheiten des Breitbandkabelnetzes treffen den Endkunden nur insoweit, als dass er sich den Anschluss unter Umständen mit einigen hundert anderen teilen muss, was sich negativ auf Bandbreite, Verzögerung und Jitter auswirken kann. Um diesen möglichen negativen Effekt etwas einzudämmen, wird manchmal (nicht immer!) die maximale Geschwindigkeit pro Teilnehmer künstlich so weit verringert, dass in Summe die Gesamtkapazität eines Kabelsegments nicht überschritten werden kann. So kann man im Breitbandkabelnetz etwa mit vergleichbaren Durchsatzwerten wie mit ADSL rechnen.

Aufgrund der technischen Gegebenheiten bietet sich die Technologie vor allem für kleine Außenstellen oder Heimbüros an. Wenn hier keine strikten QoS-Anforderungen gestellt werden, ist die Technologie eine preisgünstige Alternative.

10.2.4 Digitale Standardfestverbindungen

Digitale Standardfestverbindungen sind gewissermaßen ein Abfallprodukt der digitalen Telefonnetze. Je nach Land und Carrier werden verschiedene Geschwindigkeiten angeboten, üblich sind 64 KBit/s, 128 KBit/s und 2 MBit/s, teilweise auch 34 MBit/s. Manche Anbieter bieten auch maßgeschneiderte Geschwindigkeiten, meist in Abstufungen von 64 KBit/s an ($n \times 64$ KBit/s) oder einigen Zwischenwerten (256, 512 usw.) bis hin zu 34-MBit/s-Zugängen, die in Schritten von 2×2 MBit/s angepasst werden können. Auf-

grund der immer noch monopolähnlichen Marktposition der Deutschen Telekom AG (DTAG) im Last-Mile-Bereich, der auch von fast allen anderen Mitbewerbern für deren Datendienste mitbenutzt wird, wird im folgenden technischen Teil ausnahmsweise detailliert auf die Produkte eines bestimmten Carriers, eben der DTAG eingegangen.

Standardfestverbindung 64 KKit/s, Typ: Digital 64S

Die Technik orientiert sich stark am ISDN, genau genommen ist eine digitale Standardfestverbindung Digital 64S ein ISDN-B-Kanal (B1-Kanal), der am Vermittlungssystem vorbei fest zur Gegenstelle geschaltet ist. Der D-Kanal und der zweite B-Kanal bleiben einfach unbenutzt. Somit vereinfacht sich für den Carrier der Aufwand ungemein, denn er setzt eine einzige Technologie sowohl für sein Sprachnetz (Leitungsvermittlung) als auch für seine Datenfestverbindungen ein. Auf der Schicht 1 der Verbindung wird mit dem I.430-Protokoll, wie auch beim ISDN, gearbeitet. Früher bot die Telekom auch eine Verbindung mit ITU-T-G.703-Protokoll an, diese Verbindung wurde als Digital 64 U (U steht für unstrukturiert) bezeichnet, ist aber mittlerweile nicht mehr im offiziellen Angebot der DTAG zu finden.

Standardfestverbindung 128 KBit/s, Typ Digital 64S2

Hier werden anstelle eines B-Kanals wie bei der Digital 64S beide Kanäle (B1- und B2-Kanal) verwendet, und man kann Daten mit 128 KKit/s übertragen.

Standardfestverbindung 2 MBit/s

Auch hier gibt es eine strukturierte (Digital 2MS) und unstrukturierte (Digital 2MU) Variante. Im ersten Fall läuft auf der Schicht 1 das G.703-Protokoll, dem eine Rahmenstruktur nach der IUT-T-G.704-Empfehlung übergeordnet ist, wodurch ein 64 KBit/s-Zeitschlitz verloren geht und die nutzbare Bandbreite nur noch 1,984 MBit/s beträgt. Allerdings können die verbleibenden 31 Kanäle auch dazu benutzt werden, Verbindungen anstatt zu einer Gegenstelle zu 31 unterschiedlichen Gegenstellen mit jeweils 64 MBit/s zu betreiben!

Typ	Nutzbare Geschwindigkeit	Fehler-rate	Mittlere Verfüg.	Schnittstellen (elektrisch und Rahmenstruktur)
Digital 64S2	128 KBit/s	10^{-7}	98,5%	ITU-T empf. I.430 (S0-FVM)
Digital 2MU	2,048 MBit/s	10^{-7}	98,5%	ITU-T G.703
Digital 2MS	1,984 MBit/s	10^{-7}	98,5%	ITU-T G.703 und G.704
Digital 34M	34,368 MBit/s	10^{-7}	98,5%	ITU-T G.703
Digital 155M	149,760 MBit/s	10^{-7}	98,5%	ITU-T G.703, G.708 und G.709 (SDH STM-1 mit VC4)

Tabelle 10.3: Das aktuelle Angebot an digitalen Standardfestverbindungen der DTAG

Die unstrukturierte Digital 2MU überträgt hingegen mit der vollen Bandbreite von 2,048 MBit/s Daten zwischen zwei Gegenstellen. Hier wird auf das Framing (G.704) verzichtet.

Damit nicht genug, es sind mittlerweile auch andere, höhere Geschwindigkeiten und verschiedene Abstufungen erhältlich. So sind mittlerweile auch E3-Verbindungen (34.368 MBit/s) unstrukturiert (G.703) oder strukturiert erhältlich, ebenso so genannte N-mal-2-MBit/s-Verbindungen mit einer kundenspezifischen Bandbreite in Vielfachen von 2 MBit/s. Hierbei werden einfach nur die E1-Zeitschlitzte belegt, die der Kunde auch wirklich benötigt und bezahlt. Bandbreitenanpassungen nach unten oder nach oben bedingen in der Folge kein weiteres oder anderes Equipment und können vom Provider sehr zeitnah konfiguriert werden.

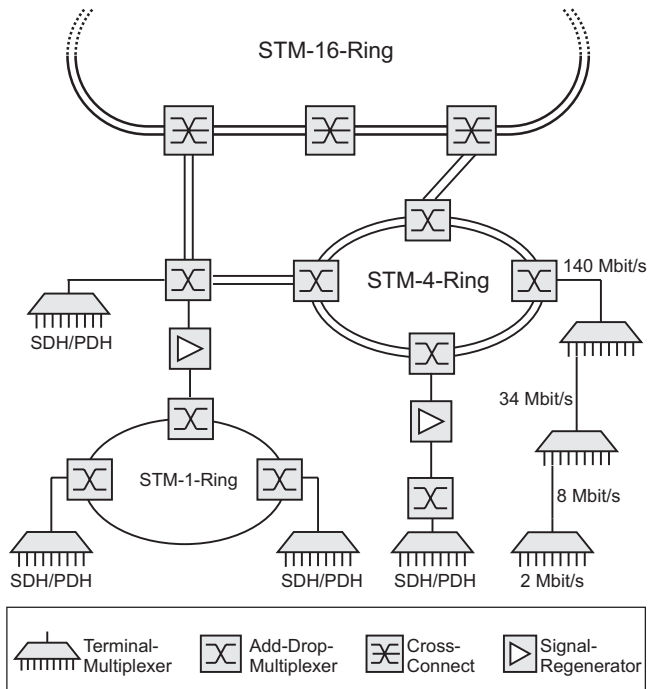


Abbildung 10.13: Die Basis aller digitalen Festverbindungen ist die Synchron Digitale Hierarchie (SDH), ein riesiges optisches Netzwerk.

Der Begriff „Mietleitung“ oder „Standleitung“, der oft suggeriert, es handle sich um eine physikalische Leitung, die dem Kunden exklusiv zur Verfügung gestellt wird, ist in diesem Zusammenhang falsch. Eine digitale Standardfestverbindung, nehmen wir der Einfachheit halber eine Digital 64/s, besteht aus zwei Zugängen vom Endkunden zu den Vermittlungssystemen des Carriers, üblicherweise sind dies zweidrahtige Kupferleitungen. Zwischen den beiden Vermittlungssystemen und möglicherweise weiteren, auf der Übertragungsstrecke zu durchlaufenden Vermittlungsknoten, liegen Glasfaserverbindungen, auf denen eine Technologie zum statischen Multiplexen, üblicherweise SDH (Synchron Digitale Hierarchie), eingesetzt wird. In diese Verbindungen wird zum Beispiel ein 64KBit/s-Kanal eingefügt (Add) und auf der Gegenstelle wieder ausgekoppelt (Drop).

Die Multiplex-Technologie bei den Carriern bezeichnet man gelegentlich als „Five Nines“ (Fünf Neunen), was bedeuten soll, dass solche Netze eine Verfügbarkeit von 99,999% aufweisen – eben das, was Sie vom Telefonnetz gewöhnt sind und erwarten. Diese Netzverfügbarkeiten, ebenso wie die minimalen Verzögerungen, garantierten Bandbreiten, und niedrige Bitfehlerraten zeichnen die digitalen Standardfestverbindungen aus, wobei im Zugangsbereich üblicherweise keine so (99,999%) hohe Verfügbarkeit mehr garantiert wird, da hier bei Standardanschlüssen keine Ersatzwege (Backup) existieren und das Equipment ebenfalls nicht in den Preisregionen von SDH-Komponenten liegt.

Das High Speed Serial Interface (HSSI) ist, analog zu X.21/V.35, ein Standard zur vollständigen Schnittstellenbeschreibung zwischen Datenend- und Datenübertragungsgeräten (DTE und DCE) – allerdings für Geschwindigkeiten bis zu 52 MBit/s. Es hat sich mittlerweile zu einem verbreiteten Standard für T3- und E3-Verbindungen gemausert, der von allen namhaften Router-Herstellern unterstützt wird. HSSI verwendet einen 50-poligen Stecker, der baugleich mit SCSI-II-Steckverbindern ist, jedoch andere elektrische Spezifikationen erfüllen muss.

Da viele Carrier und Service Provider intern üblicherweise SDH benutzen und PDH in Form von digitalen Standardfestverbindungen zu ihren Kunden führen, werden die seriellen Schnittstellen den Kunden über ein zusätzliches Modem zur Verfügung gestellt. Da mittlerweile jedoch viele Router oder Gateways auch direkt die digitalen 34-MBit/s-, 2MBit/s-, 64-KBit/s- und 128-KBit/s-Schnittstellen unterstützen und diese in Summe meist günstiger sind, sind serielle Schnittstellen und der damit verbundene Mehraufwand durch das DCE (Modem, Adapter) mehr und mehr auf dem Rückzug.

Nach wie vor spielen analoge Wählverbindungen eine große Rolle, wenn es um mobilen Remote Access geht. Denn obwohl ISDN, DSL und andere Technologien gerade boomen und sich auch, wie im Fall ISDN, schon sehr weit verbreitet haben, ist eine analoge Telefonleitung letztendlich doch das, was man im Hotelzimmer oder beim Kunden, wenn überhaupt, zur Verfügung hat.

Analog-Modems wandeln digitale Signale in analoge Signale um, um sie über leitungsvermittelnde, analoge Sprachverbindungen zu übertragen. Auf der Gegenstelle werden die Signale wieder in digitale Form gebracht. Das hört sich einfach an, wird aber komplex, sobald die Bandbreite der zu übertragenden Daten höher als die Nutzbandbreite des Sprachkanals ist – und dieser geht nur bis 3,6 KHz. Bis zu dieser Bandbreite eignen sich noch einfache Modulationsverfahren, jenseits davon setzt man relativ komplizierte Verfahren ein, die jedoch auch nur bis zu einer Übertragungsbandbreite von 56 KBit/s vom Modem (upstream) und 33,6 KBit/s zum Modem (downstream) reichen, und oft nicht einmal das, denn viele Verbindungen erreichen abhängig von der Leitungsqualität oft nur Werte zwischen 40 und 48 KBit/s. Nachdem in der Vergangenheit verschiedene

herstellereigene Standards um die Vorherrschaft gerungen haben, hat sich mittlerweile V.90 als weltweiter Modemstandard durchgesetzt.

Ein Nachteil der Modemtechnologie, neben der eingeschränkten und von der Leitungsqualität abhängigen Bandbreite, ist die im Vergleich zu ISDN lange Verbindungsaufbauzeit von manchmal bis zu 30 Sekunden. Für Dial-on-Demand-Anwendungen, also jedes Mal ein Auf- und Abbau der Verbindung, sobald Daten zur Übertragung anstehen, ist diese Technologie deshalb nicht geeignet.

Wie bereits erwähnt, ist der mobile Remote Access die Domäne der analogen Modemtechnik. Ob vom Heimbüro, vom Kunden oder vom Hotelzimmer aus mit dem Unternehmensnetzwerk kommuniziert werden soll, ein einfacher analoger Zweidraht-Telefonanschluss findet sich fast überall. Und wenn der Modemstecker und die Belegung der Pins passt, dann steht einem erfolgreichen Verbindungsaufbau nichts mehr im Wege.

ISDN, Integrated Services Digital Network, der Name sagt es schon, ist viel mehr als ein Medium für Datenübertragungen. Es integriert Sprach-, Daten- und Videoübertragung in einer Technologie, die durch die Digitalisierung der Sprachnetze der Telekommunikationsanbieter flächendeckend auf jeder noch so entlegenen Ortsvermittlungsstelle (OVSt) zur Verfügung steht. Die große ISDN-Variante, der S2M-Anschluss, der 30 Transportkanäle zur Verfügung stellt, wurde schnell zum Standardanschluss für mittlere TK-Anlagen.

Der Schritt zum digitalen Teilnehmeranschluss und zu neuen Mehrwertdiensten war die logische Konsequenz, zumindest in Europa und etlichen anderen Staaten außerhalb Nordamerikas. Dort verharnte man lange auf der Analogtechnik, das amerikanische ISDN bot ohnehin nur 56 KBit/s Geschwindigkeit pro Kanal, verwirrte zusätzlich durch unterschiedliche, nationale Standards und wurde von den dortigen Carriern auch nur sehr halbherzig angeboten.

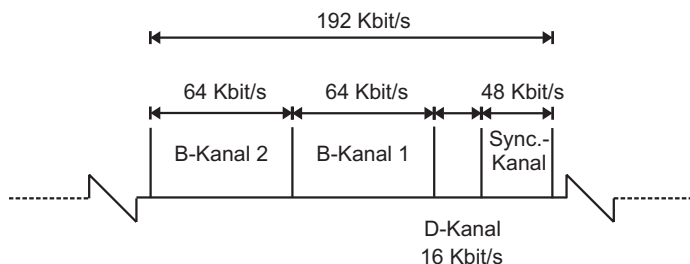


Abbildung 10.14: Die Kanalbelegung eines ISDN Basic Rate Interface

Anders in Europa. Insbesondere in England, Frankreich und Deutschland wurden bisher zig Millionen ISDN-BRI-Endanschlüsse in Haushalten und Unternehmen installiert – alleine in der Bundesrepublik gab es Anfang 2002 über 21.000.000 Leitungen!

Technisch gesehen stellt ISDN dem Endanwender den Netzwerkanschluss in Form von drei Kanälen zur Verfügung. Zwei Kanäle (B-Kanäle) erlauben die Übertragung von Daten oder Sprache mit jeweils 64 KBit/s, und ein Kanal (D-Kanal, 16 KBit/s) dient zur Leitungsvermittlung und Steuerung. Zwischen dem so genannten NTBA beim Endteilnehmer und der Ortsvermittlungsstelle (OVSt) wird eine Punkt-zu-Punkt-Verbindung in Zweidraht-Technik verwendet, wodurch die „alten“ Analogleitungen weiter benutzt werden können.

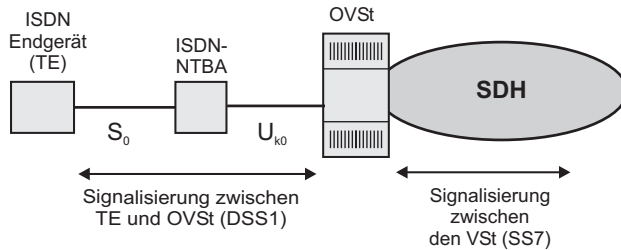


Abbildung 10.15: Euro-ISDN-Endgeräte kommunizieren über DSS1-Signalisierungsprotokoll mit dem SDH-Netz.

Zur Übertragung von Informationen über einen B-Kanal zwischen zwei ISDN-Teilnehmern wird zuerst über den D-Kanal die Verbindung ausgehandelt und aufgebaut, was bei ISDN nur ein bis zwei Sekunden dauert. Anschließend steht über das Netzwerk des Carriers ein exklusiver Datenkanal mit 64 KBit/s, minimalster Verzögerung und praktisch ohne Jitter (Änderungen der Verzögerungszeit) zur Verfügung.

Im D-Kanal können ebenfalls Daten übertragen werden, allerdings nur mit maximal 9.600 Bit/s. Dies wurde für den X.31-Dienst benutzt, um X.25-Pakete über den D-Kanal zu transportieren – aber Vorsicht, dieser Service wird nicht von jedem Carrier angeboten.

Auf der Teilnehmerseite des NTBA wird ein so genannter S_0 -Bus eingesetzt, an den bis zu acht ISDN-Endgeräte parallel angeschlossen werden können, die jeweils durch unterschiedliche Rufnummern ansprechbar sind.

In Unternehmensnetzen werden ISDN Basic Rate Interfaces hauptsächlich in kleinen Büros, Heimbüros oder als Backup für Festverbindungen eingesetzt. Üblicherweise erfolgt eine parallele Nutzung für Sprach- und Datenübertragung, da in diesen Umgebungen oft eine Datenbandbreite von 64 KBit/s ausreichend ist.

11 Design und Realisierung

Das Design eines VPN ist nicht leichter oder schwerer als das eines mit Standardtechnologie aufgebauten Netzes. Nach meiner mehrjährigen Erfahrung mit IP-VPN in mittlerweile sehr vielen, teilweise international tätigen Unternehmen waren es meist weniger praktische Probleme, mit denen die Planer und Betreiber von VPN zu kämpfen hatten. Vielmehr war es oft eine gewisse Unsicherheit gegenüber dem Thema Datensicherheit und den damit verbundenen Technologien. Hier treffen ja auch zwei Welten aufeinander: Der klassische Netzwerker will ein möglichst hohes Maß an transparenter Kommunikation und der Sicherheitsverantwortliche will genau das Gegenteil, nämlich eine nach verschiedenen Kriterien eingeschränkte oder gänzlich unterbundene Kommunikation. Bisher konnten die Gegensätze organisatorisch getrennt werden; die Netzwerker bauten ihre „privaten“ Netze, und die Verbindung zur Öffentlichkeit, meist in Form des Internets, erfolgte in aller Regel über eine Firewall. Mit dem Aufkommen der Internet-VPN wurde diese Trennung aufgehoben, und beide Seiten müssen sehr eng miteinander kooperieren.

11.1 Ein VPN ist auch nur Netzwerk

Mit diesem Leitsatz, der bis auf wenige Ausnahmen durchgehend anwendbar ist, kann man sich sein Leben als VPN-Planer sehr erleichtern. Der Trick dabei ist einfach der, dass man sich an genau die gleichen Kriterien hält, die auch beim Design eines Routernetzwerks oder einer Remote-Access-Lösung gelten. Die Sicherheitsstrategie gilt auch für traditionelle Netze, denn auch sie benutzen öffentliche Netze wie ISDN, Frame Relay oder ATM. Der Unterschied ist einfach nur, dass das öffentliche Netzwerk jetzt Internet heißt und ein Netzwerk mit virtuellen Verbindungen auf der Netzwerkschicht des OSI-Modells vorliegt und sehr viel mehr Zugangspunkte besitzt.

So viel zur Theorie. Die Wirklichkeit sieht, wie Sie wissen, leider etwas anders aus, denn in sehr vielen Organisationen wurde die Sicherheitsstrategie, die bei Internet-VPN plötzlich zum K.o.-Kriterium aufgestiegen ist, auf ISDN, Festverbindungen, Frame Relay oder ATM überhaupt nicht angewendet. Hand aufs Herz: Wer verschlüsselt denn seine Dial-In-Remote-Access-Verbindungen, wer verschlüsselt seine Frame-Relay- oder ATM-PVC? Nach vorsichtigen Schätzungen anhand dessen, was die Router-Marktführer bisher an entsprechender Technologie umgesetzt haben, sind deutlich weniger als 1% aller WAN- und Remote-Access-Verbindungen durch Verschlüsselung gegen Verletzungen der Datenvertraulichkeit geschützt! Und das, obwohl alle Telefonnetze gesetzlich zur Bereitstellung von Abhöreinrichtungen verpflichtet sind.

Natürlich darf das nur mit einer richterlichen Genehmigung erfolgen, aber die Gefängnisse sind voll von Leuten, die Gesetze gebrochen haben. Es ist auch verboten Denial-of-Service-Angriffe zu führen – schützt man sich deshalb nicht davor? Und pikanterweise ist es, technisch gesehen, wesentlich einfacher, eine Kommunikation über ein leitungs-

vermittelndes Medium wie ISDN, Frame Relay oder ATM abzuhören, als wenn diese über ein echtes, rein paketvermittelndes IP-Netz wie das Internet erfolgt.

Fakt ist also, dass im Bereich der herkömmlichen WAN-Infrastruktur nichts von dem getan wurde, was viele Sicherheitsstrategien für die Datenkommunikation über öffentliche Netze fordern. Man hat einfach Augen und Ohren verschlossen und Verbindungen wie Standardfestverbindungen oder Frame-Relay-PVC kurzum als privat und damit sicher erklärt. Dies muss aber spätestens beim Thema Internet-VPN korrigiert werden, da hier keine spitzfindigen Formulierungen mehr gegen die Zeitungs- und Nachrichtenbeiträge helfen, welche die Entscheidungsträger in den Unternehmen zum Thema Viren und andere Gefahren aus den Tiefen des Internets fast tagtäglich zu lesen und zu hören bekommen.

Allerdings kann man das Thema Sicherheit in der Planungsphase zeitlich erst einmal etwas nach hinten verschieben, und zwar bis zur Auswahl der Tunneling- und Sicherheitsprotokolle. Bis dahin kann man mit einfachen Analogien arbeiten und die virtuellen Verbindungen ganz einfach wie reale Verbindungen – oder andere virtuelle Verbindungen, wie sie in ATM oder Frame Relay verwendet werden – betrachten und sein Design entsprechend aufsetzen.

Erst wenn dies steht, dann kann man klären, welche Maßnahmen hinsichtlich Sicherheit und Quality of Service notwendig sind.

11.2 Die Ist-Aufnahme

Die Ist-Aufnahme ist ein sehr wichtiger Prozess, der neben seiner Bedeutung für eine solide Planung auch wichtige Erkenntnisse über den tatsächlichen Zustand eines Netzwerks vermittelt. Aus dem Ergebnis lassen sich nebenbei auch Informationen für andere Projekte im Bereich der LAN-Technologie und für andere Netzwerkbereiche gewinnen. Natürlich bietet die Ist-Aufnahme auch eine gute Gelegenheit, die Netzwerkdokumentation wieder einmal auf einen aktuellen Stand zu bringen, da diese nicht selten der Wirklichkeit etwas hinterherhinkt.

Ein Fehler ist dabei aber unbedingt zu vermeiden, nämlich Dinge in die Ist-Aufnahme eines Netzwerks aufzunehmen, die keinerlei Beitrag zur Funktion des Netzwerks leisten. Das erhöht zwar den Aufwand der Ist-Aufnahme, macht sie aber auch erst zu einer solchen. Ein Beispiel: Ein WAN-Router sei mit einer Software ausgestattet, die verschiedene Routing-Protokolle für IP, IPX, AppleTalk und VinesIP unterstützt; es werden aber nur IP (mit OSPF) und IPX (mit IPX-RIP) geroutet. Eine korrekte Ist-Aufnahme weist dieses Gerät als Router mit OSPF und IPX-RIP und einer Option für AppleTalk und VinesIP aus. Warum ist diese anscheinende Wortklauberei so wichtig? Ganz einfach: Aus der Ist-Aufnahme wird zusammen mit dem Sollzustand letztendlich ein Anforderungskatalog erstellt, auf dessen Basis die neu zu beschaffenden Systeme ausgewählt werden.

Wenn der Router jetzt als System, das IP, IPX, VinesIP und AppleTalk routet, aufgenommen worden wäre, müsste das neue System ebenfalls alle diese Features unterstützen, obwohl der Sollzustand dies vielleicht gar nicht fordert. Der Grund dafür ist die Migrationsphase, während der beide Welten für einen gewissen Zeitraum nebeneinander existieren.

tieren müssen. Im anderen, korrekten Fall würde man jedoch sehen, dass im Augenblick kein VinesIP und AppleTalk benötigt wird, dass es in Zukunft (Sollzustand) ebenfalls nicht benötigt wird und somit in den neuen Systemen überflüssig ist – und damit auch nicht beschafft und bezahlt werden muss. Für ein kleines Netzwerk, dessen Topologie der Netzwerkadministrator ohnehin komplett im Kopf hat, mag dies vielleicht noch etwas übertrieben scheinen. Aber sobald die Anzahl der Router zwei-, drei- oder sogar vierstellig ist, die Geräte auf weit auseinander liegende Standorte verteilt sind und die Inventarisierung von verschiedenen Personen durchgeführt wird, kommt man um eine strukturierte, gut vorbereitete Ist-Aufnahme und -Auswertung nicht mehr herum.

Was ist bei der Ist-Aufnahme nun alles zu ermitteln? Es müssen sowohl technische als auch betriebswirtschaftliche Erkenntnisse gesammelt und aufbereitet werden, nicht zuletzt deshalb, um das tatsächliche Ratiopotenzial eines VPN beurteilen zu können.

Hier kann es – das sei an dieser Stelle nicht verschwiegen, obwohl es kein Thema dieses Buches ist – zu Konflikten mit der Arbeitnehmervertretung kommen. Denn wenn man Daten sammelt, die bei missbräuchlicher Auswertung Auskunft über das Verhalten von Benutzern geben können, ist der Betriebsrat in aller Regel zustimmungspflichtig. Dies gilt insbesondere für Verkehrsanalysen. Hier sollte also rechtzeitig mit dem Betriebsrat gesprochen werden, um Konflikte von vornherein zu vermeiden.

11.2.1 Technische Aspekte

Netzwerk-Hardware

Neben einer reinen Hardware-Inventur der Router, Switches, Hubs, Remote Bridges, Remote-Access-Konzentratoren, Firewalls und Workstations für Netzwerkmanagement-Applikationen sollte auch eine Funktionsbeschreibung und Abschätzung der aktuellen Leistungsfähigkeit erfolgen. Man muss ermitteln, was wo installiert ist, wozu es benutzt wird und benutzt werden kann, was es leistet (Durchsatz, QoS) und inwieweit es erweiterbar ist. In unserem Fall ist es dabei gar nicht mal als schlecht zu bewerten, wenn es sich herausstellt, dass bei einem weiteren Wachstum des Netzwerks bestimmte Geräte das Ende ihrer Verwendbarkeit bereits erreicht haben. Auch hier ist es notwendig, nicht benutzte Module oder Schnittstellen zu kennzeichnen.

Netzwerk-Software

Hierbei gilt es, die auf den inventarisierten Netzwerksystemen installierte Software und weitere Programme zu ermitteln, die eine bestimmte Netzwerkfunktionalität ausüben, beispielsweise Netzwerkmanagement-Applikationen, Dial-In-Clients, DHCP- und DNS-Server, Konfigurationstools usw.

Neben der reinen Inventarisierung sollte auch die tatsächlich benutzte Funktionalität hervorgehoben werden – und nur diese, denn Features oder Module, die (aus welchen Gründen auch immer) nicht benutzt werden, dürfen bei einer Ist-Aufnahme keine Rolle spielen. Man kann sie zwar aufnehmen, sollte dies auch tun, muss sie aber entsprechend kennzeichnen.

Netzwerkprotokolle

Es wird nicht selten vorkommen, dass eine ganze Reihe unterschiedlicher Netzwerkprotokolle in lokalen Netzwerken eingesetzt wird. Im WAN-Umfeld werden aus verschiedenen Gründen oft nur wenige Protokolle verwendet, meist IP, sehr selten gefolgt von IPX, AppleTalk und Vines-IP oder Hostprotokollen wie SNA. Über den Sinn oder Unsinn einer solchen Protokollvielfalt muss man sich erst später, bei der Definition des Sollzustands, Gedanken machen.

Bei der Aufnahme der Protokolle ist es auch wichtig, ein Mengengerüst aufzustellen, aus dem man ablesen kann, wie viele Benutzer und Applikationen die betreffenden Protokolle überhaupt einsetzen. Und ob hier durch bestimmte Maßnahmen die Anzahl verschiedener Protokolle reduziert werden kann.

Verbindungen

Hierbei geht es um die Aufstellung aller Verbindungen und Verbindungstypen mit ihren technischen Möglichkeiten. Es sind hierbei erst einmal all die Parameter aufzunehmen, welche die jeweilige Technologie kennzeichnen, also Werte wie maximal zur Verfügung stehende Bandbreite, garantierte Bandbreite, Verzögerung, Verfügbarkeit, Serviceklassen usw. An dieser Stelle kann zu Planungszwecken auch die Mietdauer der Leitung oder Verbindung mit angegeben werden, um später daraus einen geeigneten Migrationsplan abzuleiten und gegebenenfalls fristgerecht bestimmte Leitungen kündigen zu können.

Unter diesen Punkt fallen auch Wählverbindungen – oder genauer gesagt: die Verbindungen der Remote-Access-Konzentratoren zu einem Carrier.

Verkehrsbeziehungen und Applikationen

Dieser Punkt verdient eine besondere Beachtung, denn er bietet in Verbindung mit der zu installierenden VPN-Technologie einen guten Ansatzpunkt zur Optimierung des Netzwerks – also dafür, mehr Leistung für weniger Geld zu bekommen. Denn wenn man aus der Anzahl und Art der bisher existierenden Verbindungen die Anforderungen an die virtuellen Verbindungen eines VPN ableitet, liegt man sehr oft gründlich falsch. Denn wenn Sie beispielsweise aus der Tatsache, dass zwei Niederlassungen über eine 128-KBit/s-Festverbindung verbunden sind, ableiten, eine virtuelle Verbindung über ein IP-VPN müsse mit garantiert 128 KBit/s arbeiten, dürfe nur eine unveränderliche Verzögerungszeit von wenigen Millisekunden aufweisen und habe 99,999% Verfügbarkeit zu bieten, ist das schlicht falsch!

Hier kann nur eine Analyse des tatsächlichen Datenverkehrs und der ihn erzeugenden Anwendungen helfen, und je gründlicher diese ist, desto besser können Sie später Ihre Anforderungen in einem Service Level Agreement formulieren. Ein detailliertes Verkehrsprofil hilft aber nicht nur dabei, denn wenn Sie das ganze Netzwerk mit einbeziehen, können Sie auch wichtige Erkenntnisse für den vielleicht geplanten Einsatz von Quality-of-Service, Policy-Services usw. erhalten – oder einfach nur Ihr Netzwerk etwas anders strukturieren oder segmentieren.

Solch eine Analyse ist nicht trivial. Es gilt, eine gewaltige Datenflut zu sammeln, auszuwerten und so aufzubereiten, dass man daraus entsprechende Maßnahmen ableiten kann. Es müssen sowohl zeitliche Abhängigkeiten als auch Abhängigkeiten von Protokollen und Applikationen ermittelt werden. Sie müssen die Frage beantworten können, welche Applikationen zu welchen Zeiten über ein WAN welche Art von Datenverkehr erzeugen. Sie können, als Vorgriff auf eine später vorzunehmende Klassifizierung und Priorisierung, die Datenströme hier bereits in verschiedene Klassen einteilen. Dies gilt sowohl für permanente Verbindungen als auch für den Remote Access.

11.2.2 Betriebswirtschaftliche Aspekte

Getätigte Investitionen und Restwerte

Je nachdem, welches Equipment durch die neuen VPN-Systeme ersetzt werden soll, kann es sein, dass diese Geräte noch einen gewissen Restwert haben. Dies hängt im Wesentlichen vom Beschaffungsdatum und der Abschreibungsdauer ab. Insbesondere wenn relativ neue Router oder Remote-Access-Konzentratoren ersetzt werden sollen, muss dieser Wert in eine entsprechende realistische Wirtschaftlichkeitsrechnung mit einfließen.

Verbindungskosten

Die Kosten, also die Bereitstellungsgebühren, regelmäßigen Fixkosten und die volumen- oder zeitabhängigen Tarife des Carriers sind entsprechend aufzunehmen. Es ist sinnvoll, auch eine Unterteilung nach Art der Kosten vorzunehmen, um zu sehen, wie hoch die „Leerlaufkosten“, also die Kosten ohne übertragene Daten, des Netzwerks sind. Denn genau dies ist ein Faktor, den man tunlichst nach unten drücken sollte, damit man möglichst nur für die tatsächlich übertragene Datenmenge bezahlt. Ganz weg bekommt man Grundgebühren im Augenblick leider nicht, auch dann nicht, wenn man ein VPN einsetzt, aber man kann den Anteil ruhig so weit reduzieren, wie es nur irgendwie möglich ist.

Betriebs- und Wartungskosten

Hierbei handelt sich um alle Kosten, die der Betrieb des Netzwerks verursacht, außer den Gebühren für die Datenverbindungen und die Personalkosten. Unter die Betriebskosten fallen also Kosten für Wartungsverträge mit Zulieferern, Carriern und Servicefirmen und eine Schätzung von außergewöhnlichen Kosten, die auf den bisher dafür angefallenen Kosten beruht.

Personalkosten

Neben den reinen Aufwendungen für Gehälter und Sozialleistungen müssen hier auch die Kosten für Qualifikationsmaßnahmen für die Mitarbeiter mit einbezogen werden. Auch hier gilt: nicht in die Tasche lügen und den wirklichen Personalaufwand ermitteln. Denn oftmals nehmen manche Personen verschiedene Aufgaben in verschiedenen Bereichen wahr, und man muss daher sehr genau feststellen, welchen tatsächlichen Zeitaufwand der Betrieb zurzeit benötigt. Hier sollte möglichst kein Personenbezug erfolgen, sondern der echte Zeitaufwand für den Netzbetrieb in Manntagen ermittelt werden.

11.2.3 Sicherheit

Datenvertraulichkeit

Die Aufstellung, die hier zu machen ist, soll Auskunft darüber geben, welche WAN-Verbindungen (Standardfestverbindungen, Mietleitungen, Frame Relay, ATM, Wählverbindungen über ISDN, Modem, GSM usw.) verschlüsselt sind. Weiterhin müssen die Stärke der Verschlüsselung, das eingesetzte Verfahren und das Schlüsseltauschprotokoll oder die Qualität der Schlüssel (Lebensdauer usw.) angegeben werden.

Authentifizierung

Hier müssen alle netzwerkbezogenen Verfahren und Systeme zur Authentifizierung von Benutzern beschrieben werden. Nicht mit einbezogen werden natürlich Anmeldungen an Applikationen oder Betriebssystemen. Besonderes Augenmerk ist dabei naturgemäß auf den Remote Access zu legen.

Autorisierung

Falls Anwender auf Netzwerkebene, abhängig von ihrer User-ID (oder anderen persönlich zuordenbaren Kennzeichen), der verwendeten Arbeitsstation oder des Orts des Zugriffs, unterschiedliche Benutzerrechte zugewiesen bekommen, müssen diese ebenfalls aufgenommen und beschrieben werden, um später entsprechende Anforderungen ableiten zu können.

Allerdings sollten solche Rechte wenn möglich von den Systemen vergeben werden, die diese Ressourcen auch verwalten, zum Beispiel von Netzbetriebssystemen oder Datenbankmanagementsystemen. Zugriffsrechte auf Ressourcen wie Datenbanken, Drucker oder Dateiverzeichnisse von Routern oder Remote-Access-Konzentratoren verwalten zu lassen, ist ein falscher Ansatz.

Andere Sicherheitsmaßnahmen

Im Fall, dass darüber hinaus noch weitere Sicherheitsverfahren auf Netzwerk- oder Transportebene eingesetzt werden, müssen diese so beschrieben werden, dass daraus die möglichen Berührungspunkte zu den neu einzusetzenden Systemen erkannt und berücksichtigt werden können.

11.3 Der Sollzustand

Der Sollzustand beschreibt das Netzwerk so, wie es nach der Einführung der VPN-Lösungen aussehen soll. Mit einer guten, realistischen Planung ist dieser Zustand auch grundsätzlich erreichbar. Einzig das Mengengerüst ist der typische Wackelkandidat in der Planung, denn meist wachsen die Anforderungen an die Anzahl von Netzwerk- oder Remote-Access-Ports und der Durst nach immer mehr Bandbreite doch schneller, als ursprünglich angenommen wurde. Eine zu optimistische Schätzung führt auch meist zu höheren Anfangsinvestitionen, die man in der Regel vermeiden möchte, um das Projekt nicht unnötig zu gefährden.

11.3.1 Randbedingungen

In die Planung eines VPN fließt eine Vielzahl von Kriterien ein, und es sind verschiedene Randbedingungen und Prämissen zu beachten. Neben den „natürlichen“ Anforderungen an ein VPN, auf die in Kapitel 2 ausführlich eingegangen wurde, gibt es eine Reihe anderer Faktoren, die den angestrebten Sollzustand beeinflussen können.

Andere netzwerkbezogene Projekte

Die Einführung der VPN-Technologie ist in der Regel nicht das einzige aktuelle Projekt im Netzwerkbereich eines Unternehmens. Andere Projekte können durchaus einen – natürlich auch wechselseitigen – Einfluss ausüben. Ein Beispiel, auf das aus gutem Grund in diesem Buch bereits mehrfach eingegangen wurde, ist die Einführung von Quality –of Service, um das Netzwerk effizienter ausnutzen zu können. Hier muss einerseits die VPN-Technologie auf QoS ausgelegt werden, und andererseits müssen die QoS-Planer auch berücksichtigen, dass ihr Unternehmensnetzwerk zukünftig streckenweise das Internet benutzt, mit – je nach Service Provider – möglicherweise eingeschränkter QoS-Funktionalität.

Auch die Einführung einer PKI durch die Sicherheitsabteilung ist ein gutes Beispiel, denn hier muss die VPN-Technologie entsprechende Funktionen bieten können.

Neue Applikationen

Moderne netzwerkbasierende verteilte Applikationen üben einen zunehmend stärkeren Einfluss auf das Design von Unternehmensnetzen aus. Neben einem immer größer werdenden Bedarf an Bandbreite sind auch zunehmend Qualitätskriterien wie Verzögerungszeiten und Variationen der Paketverzögerung (Jitter) im Anforderungskatalog an das Netzwerk mit aufgeführt. Insbesondere wenn teure internationale Meetings durch Videokonferenzen ersetzt werden sollen oder wenn über IP-Netze telefoniert wird, wird der Netzwerkplaner mit Anforderungen konfrontiert, die in der reinen Datenwelt bislang unbekannt waren.

Gerade das Thema Voice over IP in VPN entwickelt sich zu einem spannenden Thema, denn hier fallen unter bestimmten Umständen Anforderungen an, die mit heutiger VPN-Technologie praktisch nicht zu erfüllen sind.

Neue Standorte

Dieser Punkt fließt dergestalt in die Planung des Sollzustands ein, als dass hier genau festgelegt wird, wie zukünftige neue Standorte in das VPN zu integrieren sind. Es müssen alle Informationen bereitgestellt werden, um die notwendigen Schritte, wie Beschaffung der Systeme, Bereitstellung der Internetzugänge usw., schnell und reibungslos durchführen zu können.

Gerade in internationalen VPN ist diesem Punkt ganz besondere Beachtung zu schenken, denn in bestimmten Bereichen unseres Planeten ist der Einsatz von Internet-VPN für bestimmte Applikationen nicht möglich: 30 Hops und 2 Sekunden Verzögerung bei extremer Verzögerungsvarianz sind einfach zu viel. Hier kann man zum Beispiel gezwungen sein, IPSec-VPN und Frame Relay VPN miteinander kombinieren – etwas, was man bei der Auswahl der VPN-Router mit einkalkulieren sollte.

Neue Geschäftsbereiche

Neue Geschäftsbereiche können ebenfalls einen starken Einfluss auf den Sollzustand des Netzwerks bzw. des VPN haben – insbesondere dann, wenn die VPN-Technologie benutzt werden soll, um die geschäftliche Kommunikation mit Dritten zu schützen: der klassische Fall eines Extranet-VPN.

Ein anderer Fall wäre beispielsweise der, dass ein größeres Unternehmen Teile seines eigenen Netzes Kunden oder Partnern zur Verfügung stellt, also praktisch selbst zu einem kleinen Service Provider wird. Auf diese Weise können ansonsten brachliegende Ressourcen Gewinn bringend genutzt werden.

11.3.2 Technische Aspekte

Wie der Sollzustand aus technischer Sicht im Einzelnen auszusehen hat, richtet sich nach einer Vielzahl von Faktoren. Hier kann auch kein allgemein gültiges Beispiel gegeben werden, denn der Durchdringungsgrad der VPN-Technologie kann sehr unterschiedlich sein und von einem „totalen VPN“ bis zum Einsatz einiger weniger Remote-Access-VPN-Clients reichen. Bei der großen Dynamik heutiger Netze kann es sogar sinnvoll sein, überhaupt keinen fixen Sollzustand als solchen zu definieren, sondern nur ganz konkrete Richtlinien aufzustellen, mit welcher Art von Komponenten das Netzwerk zu erweitern ist. In einem solchen Fall stellt man einen Plan auf, wodurch und in welcher Reihenfolge die Altsysteme zu ersetzen sind und welche Systeme für neue Standorte oder neue Dienste einzusetzen sind.

Diese Festlegungen müssen eine Reihe von Punkten genau erfassen:

VPN-Systeme

Hiermit ist weniger der Gerätetyp eines bestimmten Herstellers gemeint, sondern eine allgemeine Klassifizierung der einzusetzenden VPN-Systeme, unterschieden nach Einsatzgebiet (Remote- Access, Branch-Office- oder Extranet-VPN) und benötigten Systemressourcen. Diese hängen in Branch-Office-VPN meist von der Größe der Standorte ab, während im Remote –Access VPN die Anzahl der Benutzer und die Bandbreite interessieren, die durch die Art der verwendeten Zugangstechnologie (ISDN, xDSL, E1, E3 usw.) benötigt wird.

Bei einer Strategie, in der in einem VPN-Bereich (z.B. Standortverbindungen) zwei Hersteller eingesetzt werden sollen, müssen die Systeme hinreichend genau spezifiziert werden, ansonsten ist die Interoperabilität nicht mehr gewährleistet.

VPN-Software

Hier ist von speziellen VPN-Applikationen die Rede, also beispielsweise von IPSec-Clients für die PCs oder Notebooks von Außendienstmitarbeitern. Hier ist zu beachten, dass die Hardware auch mit der zusätzlichen Verarbeitungsleistung fertig wird, welche die Verschlüsselungsprogramme benötigen. Manche IPSec-Clients lassen sich überhaupt nur auf Rechnern installieren. Das ist bei Systemen neueren Datums und vor allem neuen Algorithmen wie AES oder ECC zum Glück nicht mehr so kritisch.

Hier ein wichtiger Tipp für internationale Anwendungen: Nicht alle Staaten erlauben es Privatleuten, ihre Daten- oder Sprachkommunikation beliebig zu verschlüsseln. Bei der ganzen Diskussion um Exportbeschränkungen (vor allem aus den USA) in den letzten Jahren wurde etwas anderes aus den Augen verloren, nämlich dass bestimmte Länder den Betrieb, die Einfuhr oder sogar schon den Besitz von Verschlüsselungssystemen teilweise empfindlich, teilweise sogar mit dem Tode bestrafen! Hier sollte zum Schutz der Mitarbeiter unbedingt im Vorfeld abgeklärt werden, was in dem in Frage kommenden Land zulässig ist und was nicht!

Service Level Agreements

Eine wichtige Entscheidung ist auch die, welcher oder welche Provider eingesetzt werden darf bzw. dürfen. Man kann sich hier bereits auf bestimmte Unternehmen festlegen, falls es aus bestimmten Gründen sinnvoll erscheint. Man kann aber auch lediglich einen Rahmen vorgeben und es der jeweiligen Organisationseinheit oder Niederlassung überlassen, sich einen Provider auszusuchen, der in das technische Schema passt und im vorgegebenen Preisrahmen liegt. Hierbei sollte jedoch das SLA in den kritischen Teilen bereits vorformuliert und nicht mehr änderbar sein.

Integration in das Netzwerkmanagement

Hier werden konkrete Vorgaben gemacht, welche Funktionalitäten die Systeme bieten müssen, um in das bestehende oder geplante Netzwerkmanagement integriert werden zu können. Allerdings ist hier zu beachten, dass viele gute VPN-Systeme primär als Security-Gateways konzipiert sind und daher nur eine ganz eingeschränkte SNMP-Funktionalität bieten. Dies resultiert aus der praktisch nicht vorhandenen Sicherheit von SNMPv1 und SNMPv2.

Allerdings sollten bestimmte, nicht sicherheitskritische, aber für den Betrieb notwendigen Parameter bei SNMP-Get-Abfrage ausgelesen werden können. Auch Traps, die ein Netzwerkmanagementsystem über kritische Zustände in einem VPN-System informieren, sollten verfügbar sein.

11.3.3 Betriebswirtschaftliche Aspekte

Die betriebswirtschaftliche Sicht des Sollzustands muss sich durch eine Sache ganz besonders auszeichnen: Die Gesamtkosten über einen zu betrachtenden Zeitraum von vier oder fünf Jahren müssen bei vergleichbarer Leistung und vergleichbarem Umfang des Netzwerks niedriger sein als die Kosten, die in der Istanalyse für diese Periode ermittelt wurden. Ansonsten stimmt etwas nicht in der Berechnung, oder das VPN erfüllt seinen Hauptzweck nicht – nämlich die Betriebskosten zu reduzieren.

Falls im Sollzustand Erweiterungen des Netzwerks, neue Dienste, erweiterte Leistungen usw. hinzugekommen sind, sollten diese im Sinne einer möglichst hohen Transparenz getrennt ausgewiesen und nicht mit berechnet werden. Es muss in jedem Fall ein direkter Vergleich zwischen den Kosten der Ist-Aufnahmen und vergleichbarer Aufwände im Sollzustand möglich sein.

Die Kosten für die Einführung bzw. den Umstieg auf die VPN-Technologie sollten Sie in jedem Fall mit in die Berechnung einbeziehen.

Grob kann man zwischen folgenden Arten von Kosten unterscheiden, aus denen sich die tatsächlichen Betriebskosten eines VPN ergeben:

Einmalige Kosten

Hier fallen insbesondere die Kosten für die Investitionen in Soft- und Hardware sowie die Kosten für Planung, Consulting und die Aufwände für die Installation des Equipments an. Auch eventuell anfallende Schulungskosten fallen unter diese Kategorie. Hinzu kommen die Kosten für die Bereitstellung der notwendigen Verbindungen, also der Leitungen zu den POPs des Service Providers.

Laufende Kosten

Hier entstehen feste Kosten durch monatliche Gebühren für die bereitgestellten Verbindungen und Aufwendungen für Wartungs- und Serviceverträge für die eingesetzten Systeme. Weiterhin fallen noch variable Kosten für die übertragenen Datenvolumina und verbrauchte Verbindungszeiten an.

Die Personalkosten sind ein nicht unbeträchtlicher Faktor in dieser Rechnung. Hier muss man nicht nur die direkten Kosten für das VPN-Management mit einbeziehen, sondern auch die durch das VPN verursachten Kosten im Bereich des Benutzerservice (Help Desk). Gerade hier können, speziell im Bereich der Remote Access VPN, durch eine nicht optimale Auswahl der VPN-Client-Software schnell sehr hohe Kosten auflaufen, da sich die Probleme mit der Anzahl der Benutzer multiplizieren.

11.4 Die Übergangsphase

Die Übergangsphase ist ein ziemlich kritischer Punkt in einem VPN-Projekt, denn hier müssen über einen bestimmten Zeitraum hinweg verschiedene Systeme für den gleichen Dienst parallel betrieben werden, ohne dass sie sich gegenseitig stören dürfen.

Je nach Umfang des Unternehmensnetzwerks und des geplanten VPN-Anteils kann eine solche Übergangs- oder Migrationsphase recht lange dauern. Sie sollte in jedem Fall gut geplant werden, um später die notwendigen Prozesse im Griff zu haben. Ausführliche Tests und Probeläufe oder vielleicht abgegrenzte Pilotinstallationen, in denen man wichtige Erfahrungen für den eigentlichen Umstieg gewinnen und später umsetzen kann, können hier nicht schaden.

Wenn man clever ist, kann eine Umstellung auf die VPN-Technologie sogar derart erfolgen, dass die Anwender fast nichts davon mitbekommen und sich in ihrer täglichen Arbeitsweise auch nicht groß umstellen müssen.

Im Fall von Branch-Office-VPN ist dies tatsächlich möglich. Hier kann während eines Wartungsfensters der „alte“ Router heruntergefahren und anschließend das VPN-Gateway mit den gleichen IP-Adressen in Betrieb genommen werden. Die Benutzer bemerken davon gar nichts.

Im Remote-Access-Bereich sieht die Sache etwas anders aus, da hier die Benutzer involviert sind. Auf deren Systemen müssen die VPN-Clients installiert werden, und sie haben sich statt im Firmennetz bei einem Provider einzuwählen. Jedoch kann auch hier viel getan werden, beispielsweise mit automatischer Software-Verteilung und vorkonfigurierten Einwählskripten. Dass die Benutzer auch im VPN ihre alten Benutzer-IDs, Passwörter, PINs oder Token-Karten weiterbenutzen können, versteht sich von selbst.

11.5 Die Umsetzung der Security Policy

Die Sicherheitsstrategie des Unternehmens hat einen entscheidenden Einfluss auf das Design des VPN. Denn sie legt eindeutig fest, welche Sicherheit für die Pakete, die über ein öffentliches Netz transportiert werden, und für die VPN-Systeme selbst gefordert ist. Nach ihr richten sich die Auswahl der Client-Software, der Authentifizierungssysteme und etwaige bauliche Maßnahmen – denn ein VPN-Router, für den man eine starke Verschlüsselung vorschreibt, kann nicht in einem Büro oder Verteilerraum betrieben werden –, sofern die Security Policy konsistent umgesetzt wird. Für die netzwerkseitige Planung eines VPN muss man, je nach den Vorgaben der Sicherheitsstrategie, in der Regel folgende Kriterien berücksichtigen:

Welche Daten müssen geschützt werden? Hier kann im Netzwerkbereich nicht viel getan werden, da der Inhalt der Daten nicht Gegenstand von Protokollen der OSI-Schichten 1 bis 4 ist. Man sollte jedoch, wenn nichts anderes vorgegeben ist, den schlimmsten Fall annehmen, dass sensible Informationen im VPN übertragen werden können – auch wenn die Sicherheitsstrategie dies verbieten sollte.

Wie lange müssen die Daten sicher sein? Man muss sich dagegen schützen, dass ein Angreifer heute verschlüsselte Daten aufzeichnet, die beispielsweise zehn Jahre vertraulich bleiben müssen, und diese vielleicht erst in fünf Jahren mit der dann zur Verfügung stehenden Technologie entschlüsselt.

Vor wem müssen diese Daten geschützt werden? Die Frage ist, gegen welche Art von Angreifern man sich schützen muss: Einzelpersonen, Gruppen, Firmen oder größere Organisationen oder im schlimmsten Fall Geheimdienste. Diesen Angreifern stehen unterschiedliche Ressourcen zur Verfügung, also muss man sich entsprechend stark schützen.

Leider kann man keine allgemein gültigen Aussagen darüber machen, welche Verschlüsselungsalgorithmen und Schlüssellängen am besten einsetzbar sind. In Tabelle 11.1 sehen Sie ein paar konservativ aufgestellte Faustregeln für Schlüssellängen von Public-Key-Verfahren und die korrespondierenden Schlüssellängen von symmetrischen Verfahren. Es sollte immer ein ausgewogenes Verhältnis zwischen beiden Verfahren existieren, da die asymmetrischen Verfahren meist die Schlüssel für symmetrische Verfahren generieren oder austauschen. Es macht keinen Sinn, Daten mit 256-Bit-Schlüsseln zu verschlüsseln, die mit einem 512-Bit-Public-Key-Verfahren erzeugt wurden. Umgekehrt ist es auch ein sinnloser Aufwand, mit einer 15424-Bit-RSA-Verschlüsselung einen symmetrischen 80-Bit-Schlüssel auszutauschen.

Über 30 Jahre	256	15.424	512	15.424	512

Tabelle 11.1: Verschiedene Schlüssellängen (in Bit) und ihre voraussichtliche Schutzdauer. Die Daten stammen von ECRYPT, einer Einrichtung der EU.

Für die Public-Key-Verfahren, die Sie in Tabelle 11.1 sehen, kommen zum Beispiel für RSA praktisch keine kleineren Schlüssellängen als 1.248 Bit vor. Dies spiegelt den augenblicklichen Stand der Wissenschaft wider, versehen mit einer ordentlichen Sicherheitsreserve, um auch in der Zukunft vor bösen Überraschungen sicher zu sein. Falls es die Performance der eingesetzten Systeme zulässt, kann man auch noch größere Schlüssellängen wählen. Im Gegensatz zu symmetrischen Verfahren kann nämlich nicht so einfach bestimmt werden, mit welchem Aufwand Verfahren wie RSA zu brechen sind. Bei sicheren symmetrischen Verfahren ist der Aufwand eindeutig: Die Zeit, die ein Brute-Force-Angriff benötigt, steht in einem festen Verhältnis von Schlüssellänge und eingesetzten Ressourcen (CPU-Geschwindigkeit, Arbeitsspeicher, spezielle Chipsätze usw.). Bei Verfahren wie RSA kommt noch ein weiterer Faktor dazu, nämlich die weiteren, nicht vorhersagbaren Entwicklungen in der Zahlentheorie, speziell bei den Forschungen im Bereich Faktorisierung und diskrete Logarithmen.

Beim IKE-Protokoll sollte man sich sorgfältig darüber Gedanken machen, in welcher Kombination die Schlüssellängen der eingesetzten Algorithmen Sinn machen. Der IPSec-Standard schreibt vor, dass DES mit 56 Bit Schlüssellänge und Diffie-Hellman mit der Oakley-Gruppe 1 (768 Bit) zu unterstützen sind. Dies müssen also die eingesetzten IPSec-Systeme in jedem Fall unterstützen. Diese Kombination bietet aber nur minimale Sicherheit.

Falls man aber eine höhere Sicherheit benötigt, als der DES-Algorithmus bietet, und beispielsweise Triple-DES einsetzt, dann muss auch das IKE-Protokoll entsprechend angepasst werden und wenigstens die Oakley-Gruppe 2 mit einem 1.024-Bit-Modulus, besser 1.536 Bit, verwendet werden. Ansonsten wäre plötzlich das Schlüsseltauschprotokoll das schwache Glied in der Kette. Es nutzt nichts, eine starke Verschlüsselung einzusetzen, wenn die Erzeugung des Schlüssels, von dem ja die ganze Sicherheit abhängt, zu schwach ist.

In Tabelle 11.1 können Sie erkennen, dass eine symmetrische 128-Bit-Verschlüsselung heutzutage auch für hohe Sicherheitsanforderungen als ausreichend angesehen wird. Sichere symmetrische Verfahren sind hinsichtlich ihrer Sicherheit relativ einfach zu bewerten. Da man den Aufwand für die schnellstmögliche Kryptoanalyse (Brute-Force-Angriff) genau kennt, kann man sich durch entsprechend lange Schlüssel auch gegen Angriffe von Geheimdiensten zuverlässig schützen. Allgemein gelten Schlüssellängen von 128 Bit bei einem sicheren Verfahren, z.B. AES, als nicht mehr „knackbar“.

Der Remote Access ist einer der wichtigsten Stützpfeiler heutiger VPNs, da hier das größte Ratiopotenzial liegt. Allerdings kann man hier auch sehr viele unnötige Kosten erzeugen. Bevor wir einige praktische Beispiele diskutieren, sollten zuerst einmal die grundsätzlichen Anforderungen an die wichtigsten VPN-Komponenten wie VPN-Router für Remote Access oder Standortverbindungen und VPN-Gateways für Extranet Access beleuchtet werden.

Der VPN-Router ersetzt den oder die herkömmlichen Remote-Access-Server, in die sich die mobilen Benutzer per ISDN, ADSL, WLAN-Hotspot oder anderer Zugangstechnologie eingewählt haben. Um ein solches System funktionell zu ersetzen, muss eine Reihe von Kriterien beachtet werden:

Die Anzahl der parallelen Tunnel muss mindestens der Anzahl der Benutzer entsprechen, die sich gleichzeitig einwählen dürfen. Man sollte diesen Wert aber mit Blick auf die Zukunft nicht zu niedrig ansetzen. In der Regel stellt man weniger gleichzeitige Verbindungen zur Verfügung, als man Benutzer hat; man überbucht das System. Der Grad der Überbuchung ist ein Wert, der ausschließlich vom Einsatzgebiet des Remote Access VPN bestimmt wird. Die Bandbreiten reichen in der Praxis von einer 1:10-Überbuchung bis hin zur Garantie einer Verbindung für jeden Benutzer. Es gibt nämlich durchaus VPN-Einsatzgebiete, die eine Überbuchung verbieten, z.B. Online-Reservierungssysteme in Reisebüros, Online-Vertragsabschlüsse usw. Hier erfordert das Geschäftsmodell, dass immer eine Verbindung zustande kommen muss.

Der Gesamtdurchsatz ist ein sehr interessanter Faktor, der zunehmend an Bedeutung gewinnt. Zunehmend deshalb, weil er in Zukunft sehr viel höher sein muss als der eines vergleichbaren Remote-Access-Konzentrators. Der Grund liegt darin, dass zunehmend mit höheren Geschwindigkeiten als den 56 KBit/s eines Modems oder 64 KBit/s von ISDN gearbeitet wird. Die ADSL-Technologie ermöglicht Datenraten von mehreren MBit/s. Ein Remote-Access-Server für 600 Benutzer benötigt im schlimmsten Fall einen Durchsatz von $600 \times 64 \text{ KBit/s} = 34 \text{ MBit/s}$, ein Wert, der technisch kein Problem darstellt. Im VPN-Router sähe der schlimmste Fall jedoch so aus, dass alle Benutzer DSL einsetzen und insgesamt $600 \times 2 \text{ MBit/s} = 1,2 \text{ GBit/s}$ an Bandbreite benötigen.

Dies ist natürlich nur ein theoretischer Fall. Bleiben wir auf dem Teppich und nehmen wir an, dass nicht alle Benutzer zur gleichen Zeit Videofilme von irgendwelchen Tauschbörsen laden. Angenommen, es würden im Schnitt nur 10–30% der Bandbreite benutzt, dann betrüge die erforderliche Bandbreite in diesem Beispiel immer noch 120–360 MBit/s. Wenn das System nicht oder nicht stark überbucht ist, sind ohnehin nicht alle Benutzer aktiv, und Werte sind entsprechend niedriger. Bei normalem Verkehrsprofil (Anwendungen wie E-Mail, Client/Server usw.) ist der mittlere Durchsatz auch gleich noch ein Vielfaches niedriger.

Falls jedoch die Remote-Access-VPN-Systeme aus wirtschaftlichen Gründen stark überbucht sind, also für vielleicht 3000 mögliche Benutzer nur 600 Ports oder Tunnel zur Verfügung stehen, dann muss man davon ausgehen, dass in den Stoßzeiten tatsächlich alle 600 Verbindungen gleichzeitig benutzt werden. Der benötigte Durchsatz in diesem Beispiel würde dann auch in höhere Regionen vorstoßen.

Bei den Überlegungen zum Durchsatz sei an dieser Stelle darauf hingewiesen, dass es sich bei einem Internet-VPN dabei natürlich um den Systemdurchsatz mit starker Verschlüsselung (128 Bit oder höher) und IPSec-Komprimierung handeln muss und nicht um einfaches Paketweiterleiten.

Die Art der Verbindung zum Service Provider richtet sich nach verschiedenen Gegebenheiten. Zum einen ist es der benötigte Gesamtdurchsatz, denn die Verbindung zum ISP muss sinnvollerweise mindestens die gleiche Bandbreite aufweisen; auf der anderen Seite fließen bestimmte, geforderte Qualitätskriterien mit ein.

Aus der Bandbreite und etwaigen Vereinbarungen in einem Service Level Agreement leiten sich dann meist die eingesetzte Technologie und die Geschwindigkeit ab.

Je nach geplantem Sollzustand und den durch die Migrationsphase bedingten Anforderungen an einen zeitweisen Parallelbetrieb müssen unter Umständen verschiedene Benutzer-Authentifizierungsdienste unterstützt werden. Praktisch alle Remote-Access-Server arbeiten mit RADIUS, selbst Systeme wie SecurID von RSA oder Microsoft Active Directory (AD) bieten RADIUS-Schnittstellen. Moderne VPN-Router für Remote-Access unterstützen neben RADIUS auch noch andere Verfahren; insbesondere wird zunehmend LDAP-Support angeboten.

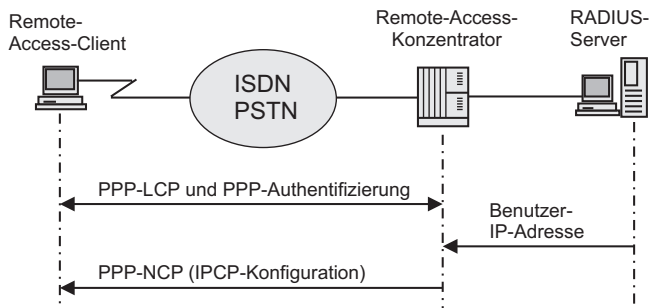


Abbildung 11.1: Beim Remote Access wurden die IP-Adressen per PPP-IPCP zugewiesen.

Falls die Remote-Access-Benutzer bereits Token-Karten zur Authentifizierung benutzen, muss dieses Verfahren ebenfalls unterstützt werden können. Der Idealfall ist der, dass die Anwender ihre bisherigen User-IDs, Passwörter und Token-Karten weiter benutzen können.

Beim Remote Access ist es üblich, dass die Rechner der Benutzer während des Verbindungsaufbaus zum Einwahlsystem von diesem eine für die Zeit der Verbindung gültige IP-Adresse zugewiesen bekommen. Üblicherweise erfolgt die Zuweisung über das PPP-Protokoll, das alle heutigen Einwahlprogramme, z.B. das Microsoft DFÜ-Netzwerk, unterstützen. In diesem Fall bekommt der Remote-Access-Konzentrator, wie Sie in Abbildung 11.1 sehen, bei einer erfolgreichen Authentifizierung vom RADIUS-Server eine IP-Adresse geliefert, die er über IPCP (IP Configuration Protocol) dem Remote-Rechner für die Dauer der Verbindung zuweist.

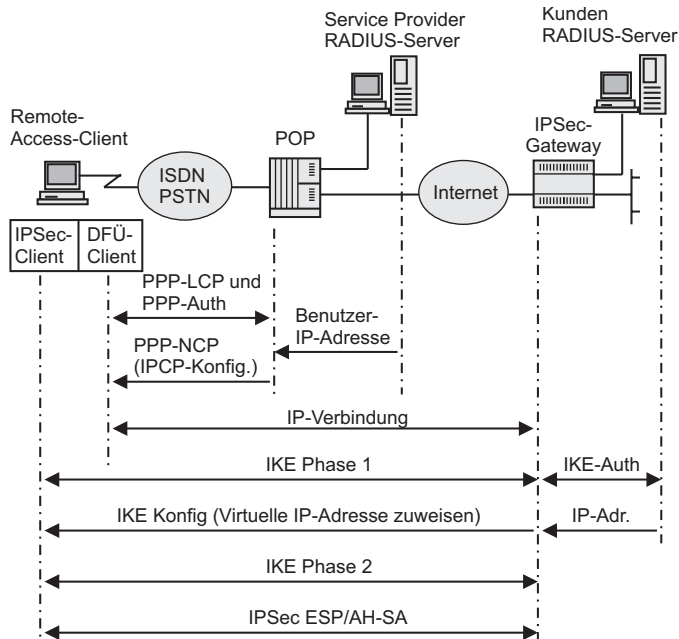


Abbildung 11.2: Im IKE-Protokoll gibt es Erweiterungen, um dem Client die IP-Adresse und andere Parameter zuzuweisen.

Im Fall von IPSec liegt der Fall jedoch anders, wie Sie in Abbildung 11.2 sehen. Hier besteht ja bereits eine PPP-Wählverbindung zum Service Provider, und der Rechner des Anwenders hat von diesem bereits eine offizielle IP-Adresse zugewiesen bekommen. IPSec (im Tunnelmodus) benötigt aber noch eine zusätzliche private IP-Adresse, die in der Regel über entsprechende Informationsnachrichten des IKE-Protokolls an den IPSec-Client übertragen wird. Diese IP-Adresse entspricht der IP-Adresse, die bisher vom Remote-Access-Server zugewiesen wurde. Hierfür müssen, je nach Planung, die gleichen Mechanismen unterstützt werden können, die auch ein Remote-Access-Server bietet. Die IP-Adressen können im Allgemeinen aus folgenden Quellen stammen:

- IP-Adressen aus einem oder mehreren gruppenspezifischen Address Pools
- IP-Adressen von einem RADIUS Server
- IP-Adressen von einem LDAP Server

IP-Adressen von einem DHCP Server

Feste, lokal konfigurierte benutzerspezifische IP-Adressen

Da der Remote Access je nach Ausbaustufe ein beträchtlicher Kostenfaktor ist, gibt es schon seit längerem entsprechende Accounting-Funktionen. Mit den damit aufgezeichneten Daten ist es möglich, die Leistungen und daraus resultierenden Kosten bis auf Benutzerebene auszuweisen. Üblicherweise benutzt man dafür auch RADIUS. Ein brauchbarer VPN-Router erzeugt ebenfalls RADIUS-Accounting-Datensätze und schickt diese zum RADIUS-Accounting-Server. Damit kann man seinen virtuellen Remote Access reibungslos in das bestehende Abrechnungssystem integrieren.

Da Remote Access, genau wie ein VPN auch, öffentliche Netze benutzt, wurde in vielen Installationen bereits reger Gebrauch von den Logging-Funktionen gemacht. Logging dient im Gegensatz zum Accounting weniger zu Abrechnungszwecken als zur Kontrolle eines ordnungsgemäßen und vor allem sicheren Betriebs von Remote-Access-Konzentratoren. Beim Logging kann man sowohl kritische Systemzustände als auch mögliche Angriffe erkennen und darauf reagieren. Gute Logging-Implementierungen erlauben es, verschiedene Logs zu führen, entsprechend der Art der Daten, die gesammelt werden sollen:

Event-Log, zur Speicherung aller Events zu Debugging-Zwecken.

Configuration Log, zur Aufzeichnung wer wann was geändert hat.

Security Log, zur Speicherung von sicherheitsrelevanten Ereignissen.

Viele spezielle VPN- oder Security-Gateways haben einen einfachen, aber wirkungsvollen Schutz gegen eine bestimmte Art von Denial-of-Service-Angriffen (DoS) eingebaut. Falls man nämlich zu viele Logging-Informationen sammelt, bietet man damit leider auch ein hervorragendes Angriffsziel. Dies wird an einem kleinen Beispiel schnell deutlich: Wenn ein IPSec-System alle abgewiesenen und nicht bearbeiteten (Silent-Discard-)Pakete aufzeichnet, dann erfordert dies einen gewissen Aufwand durch die interne Aufzeichnung und das Schreiben auf einen externen Server. Wenn nun ein Angreifer eine Reihe von kleinsten Paketen, die in jedem Fall abgewiesen werden, auf ein VPN-Gateway schickt, dann kann er, ohne große Netzlast auf dem öffentlichen Interface zu erzeugen, das Gateway dazu bringen, nichts mehr zu tun, als Logging-Einträge zu verarbeiten.

Also sollte man sein System sorgfältig konfigurieren und dafür Sorge tragen, dass nur wichtige Ereignisse aufgezeichnet werden.

Beim Einsatz von IPSec ist dies kein Thema, hier wird transparent jedes IP-Paket sicher übertragen. Dies gilt auch für SSL bei Verwendung eines SSL-Clients.

Bei SSL ohne Clients muss für jede Anwendung eine entsprechende Funktionalität im SSL-VPN-Gateway vorhanden sein. Hier scheidet sich bei den Herstellern die Spreu vom Weizen, denn einige haben einen, milde ausgedrückt, limitierten Funktionsumfang, und andere lassen sich jede Applikation extra bezahlen, so dass man auf Benutzerpreise kommt, die jenseits von Gut und Böse liegen. Also aufpassen und genau fragen!

Die Sicherheit eines VPN-Systems kann man in zwei Teilbereiche untergliedern: die Sicherheit der eingesetzten Tunneling-Protokolle und die Sicherheit des Systems selbst. Ersteres ist echt einfach: Wirkliche Sicherheit bieten praktisch nur zwei Protokolle, IPSec und SSL.

Gute Systeme bieten eine reiche Auswahl an Algorithmen, sowohl für SSL als auch für IPSec. Insbesondere bei Remote Access sollten AES und ECC implementiert sein. Ebenso muss der Einbau von Beschleunigerkarten möglich sein, insbesondere auch solche mit IKE-Hardware-Unterstützung und echten Zufallsgeneratoren. Der größte Teil der Remote-Access-Verbindungen läuft über NAT-Router, dafür müssen die IPSec-Implementierungen gerüstet sein.

Einige VPN-Router unterscheiden explizit zwischen öffentlichen (Public Interface) und privaten (Private Interface) Schnittstellen. Bei den besseren Systemen ist auf der öffentlichen Schnittstelle ein so genannter gehärteter IP-Stack implementiert, der von seiner Programmierung her nur einige wenige Protokolle verarbeiten kann und alle anderen Pakete einfach ignoriert. Solche Stacks bieten eine sehr hohe Sicherheit gegen DoS-Angriffe, unerlaubtes Eindringen oder auch Fehlkonfiguration. Ein typischer gehärteter Stack lässt nur die Protokolle AH, ESP, UDP/500, UDP/4500 und TCP/443 passieren – also ausschließlich IPSec, IKE, IKE-NAT-T und SSL.

Die Systemsicherheit selbst ist äußerst kritisch. Das wird leider in vielen Fällen völlig unterschätzt. Versetzen wir uns einmal kurz in einen Angreifer, der auf bestimmte Netzressourcen eines Unternehmens illegal zugreifen will. Er wird, um sich selbst zu schützen, möglichst versuchen, nicht physisch in das Unternehmen selbst einzudringen, sondern dies über das Internet zu tun. Wenn er nun feststellt, dass das Unternehmen in seinem VPN durchgehend IPSec mit starker Verschlüsselung (z.B. Triple-DES) einsetzt, dann wird er es an dieser Stelle erst gar nicht weiter versuchen. Er sucht sich dann einen anderen Schwachpunkt im Netzwerk, und dieser kann im Zugriff auf die IPSec-Gateways im Unternehmens selbst liegen, falls hiergegen keine Maßnahmen getroffen werden. Hier gibt es ein paar Dinge, die man unbedingt beachten sollte:

Kein direkter Out-of-Band-Zugriff: Falls man das Gerät über Modem- oder ISDN-Verbindungen fernwarten möchte, öffnet man damit die klassische Hintertür. Wenn man unbedingt einen Out-of-Band-Zugriff benötigt, dann muss dieser verschlüsselt (SSL, IPSec) erfolgen und stark authentifiziert werden.

Auch der lokale Zugriff auf einen VPN-Router sollte in jedem Fall verschlüsselt erfolgen, entweder über SSL oder noch besser gleich über IPSec. Denken Sie daran: Mehr als 80% aller Angriffe auf Rechner und Netzwerke erfolgen innerhalb eines Unternehmens und nicht von außen.

Die Systeme müssen ihrer Sicherheitseinstufung entsprechend in durch bauliche Maßnahmen sicheren Umgebungen mit einer geeigneten Zugangskontrolle betrieben werden. Ein VPN-Router gehört in ein Rechenzentrum oder andere entsprechend geschützte Umgebungen und nicht in eine Besenkammer.

Beim Thema Administration scheiden sich oft die Geister, denn hier treffen praktisch zwei „Weltanschauungen“ aufeinander: die textbasierende und die grafische Administration. Hier objektiv zu urteilen, ist unmöglich, da beide Varianten sowohl Vor- als auch Nachteile aufweisen und auch sehr stark von persönlichen Vorlieben der Administratoren selbst geprägt sind.

In der Praxis ist es natürlich eine geschickte Lösung, wenn ein VPN-System beide Varianten unterstützt und der Administrator das auswählt, was es ihm erlaubt, seine Tätigkeit am effizientesten auszuüben. Heute üblich sind grafische Oberflächen auf Basis von Webbrowsern oder Java, da hiermit praktisch von jedem Rechner aus auf das VPN-System zugegriffen werden kann. Aber sowohl die grafische als auch die textbasierende Administration können gut oder schlecht implementiert werden und im letzten Fall erhöhte Personalkosten oder, schlimmer noch, auch Sicherheitsprobleme nach sich ziehen. Hier sollte man auf Übersichtlichkeit, Einfachheit und Konsistenz achten – der Administrator sollte genau wissen, was er tut und was seine Änderungen an der Konfiguration tatsächlich bewirken.

Moderne und zukunftsichere Netze setzen zum Teil schon Verzeichnisdienste oder PKIs (Public Key Infrastructure) ein oder planen zumindest, dies in der Zukunft zu tun. Für die Realisierung eines VPN bedeutet dies, dass es die entsprechenden Schnittstellen zu solchen Systemen bereits unterstützen muss.

Im Bereich der Verzeichnisdienste gab es in der Vergangenheit viele herstellereigene Ansätze, die sich aber letztendlich nicht durchsetzen konnten. Beispiele hierfür sind StreetTalk von Banyan und NDS von Novell. Auch Standards, wie z.B. X.500, hielten nur wenig Einzug in praktische Implementierungen, da sie nicht selten etwas überdimensioniert erschienen. Auch LDAP (Lightweight Directory Access Protocol) fristete lange ein Schattendasein, wird aber immer mehr zum Standardprotokoll für Verzeichnisdienste. Dies liegt an der zunehmenden Unterstützung durch Hersteller von Betriebssystemen und Netzwerkkomponenten. Sun/Netscape, Microsoft und Novell setzen immer mehr auf dieses Protokoll, und viele Netzwerkdienste wie Policy Services oder die Verwaltung von Systemprofilen basieren auf LDAP.

Im Bereich der PKIs setzen sich erst langsam die Standards der IETF durch, dadurch haben die verschiedenen Hersteller lange Zeit, ihr eigenen, zu anderen Produkten inkompatiblen Systeme entwickeln können. Es gibt zwar eine Reihe von Standards, zum Beispiel X.509 für das Format von digitalen Zertifikaten, aber die Verteilung und das Management erfolgen meist noch mit proprietären Produkten und Protokollen. Hier muss man sowohl auf die Unterstützung der Standards achten als auch auf die im Unternehmen eingesetzten oder noch einzusetzenden Produkte.

Richtig innovative Systeme haben das CMP (Certificate Management Protocol) implementiert, mit dem VPN-Systeme in Verbindung mit einer geeigneten professionellen CA den kompletten Zertifikats-Lebenszyklus automatisieren können.

Auch der Remote Access ist mittlerweile in vielen Unternehmen zu einem kritischen Faktor geworden, so dass auch hier entsprechende Maßnahmen implementiert sein müssen.

Bei der Redundanz gibt es je nach Kompetenz des Herstellers verschiedene Ansätze, die von der manuellen Verbindungsaufnahme zu einem Backup-VPN-Router bis zum stateful Failover ohne Verbindungsunterbrechung reichen. Alle haben Vor- und Nachteile, die beste Lösung liegt vermutlich irgendwo dazwischen.

Hier muss man zwischen zwei Varianten unterscheiden, den Clients, die nur in Verbindung mit dem VPN-Router eines bestimmten Herstellers arbeiten, und Clients, die relativ offen sind und mit verschiedenen VPN- Routern zusammen arbeiten können.

Erstere Variante sind Clients, die in der Regel zusammen mit der entsprechenden Hardware oder Server-Software ausgeliefert werden. Sie haben meist herstellereigene Erweiterungen, die sie zur Kommunikation mit Systemen anderer Hersteller untauglich machen.

Die zweite Gruppe sind VPN-Clients, die entweder rein standardbasierend oder so weit offen gelegt sind, dass Dritthersteller entsprechende Funktionen einbauen können. Beispiele sind z.B. Microsoft PPTP- oder L2TP/IPSec-Clients, Linux FreeS/WAN (IPSec) oder PGP-Net (IPSec).

Um es vorwegzunehmen: In einem Remote Access VPN entscheidet zu einem maßgeblichen Teil der VPN-Client darüber, ob das VPN Kosten letztendlich einspart oder viel mehr Kosten erzeugt – oder ob es überhaupt in großem Maßstab einsetzbar ist. Das Einsparen von Kosten wurde bereits besprochen. Wodurch kann ein VPN-Client hohe Kosten erzeugen? Ganz einfach: Je größer ein Remote-Access-Einsatz ist, desto mehr Anwender benutzen den VPN-Client und desto mehr multiplizieren sich auch die Probleme, die es damit eventuell gibt. Wenn 100 Clients produktiv eingesetzt werden und ein Problem erzeugen, das eine Hotline pro Woche 30 Stunden kostet und einen Ausfall der Arbeit mit den Clients von 50 Stunden zur Folge hat, dann sind es bei 1000 Systemen schon 300 Stunden Hotline und 500 Stunden Arbeitsausfall. Worauf muss man bei VPN- oder speziell auch IPSec-Clients besonders achten? Die folgende Aufzählung deckt die wichtigsten Punkte ab, die im Folgenden noch ausführlich besprochen werden:

- Einfache Installation ohne Eingaben des Benutzers

- Keine weitere Konfiguration durch den Benutzer notwendig.

- Keine Konfigurationsänderung durch den Benutzer möglich.

- Zentrale Verwaltung und Konfiguration

- Split-Tunneling darf nicht durch den Anwender konfigurierbar sein.

- Integrierte Unterstützung für Token-Authentifizierung

- Integrierte Unterstützung für Authentifizierung mit digitalen Signaturen mit Smart-cards

Benutzertransparentes Zertifikatmanagement
Automatischer Backup-/Failover-Mechanismus
API oder direkter Aufruf per Kommandozeile oder System-Call

Die Installation des VPN-Clients kann bereits viele Kosten sparen, und zwar dann, wenn der Benutzer dies selbst tun kann und nicht aus diesem Grund den Benutzerservice benötigt. Somit sparen sowohl der Benutzer als auch die PC-Administration wertvolle Zeit. Eine gute Installation läuft ohne Abfrage von Parametern durch und kann an die Gegebenheiten eines Unternehmens angepasst werden (Customizing).

Der Benutzer braucht im Idealfall gar nichts zu konfigurieren; die Installation ist damit praktisch von jedem durchzuführen, der bereits einen PC benutzt. Client-Implementierungen, bei denen Anwender irgendwelche Systemdateien manuell editieren müssen oder bei denen Parameter abgefragt werden (Beispiel: „Wollen Sie Perfect Forwarding Secrecy [Ja/Nein]?“), mit denen ein Anwender nichts anfangen kann, sollten von vornherein nicht in Betracht gezogen werden.

Ganz ohne Zutun des Benutzers geht es, wenn im Unternehmen entsprechende Software-Verteilmechanismen eingeführt sind, zum Beispiel Microsoft SMS.

Damit kommen wir auch zum nächsten Punkt, denn bei einer guten VPN-Client-Software braucht der Anwender nicht nur nichts zu konfigurieren, sondern er darf es überhaupt nicht können. Denn es ist der Horror eines jeden Sicherheitsverantwortlichen, wenn die Endanwender in der Lage sind, auf ihren IPSec-Clients irgendwelche Konfigurationsänderungen vorzunehmen – zum Beispiel zum Zweck der „Performance-Steigerung“ die Verschlüsselung abzuschalten. Es gibt zwar die Möglichkeit, dies im VPN-Router zu unterbinden und überhaupt keine IPSec-Sicherheitsassoziation aufzubauen, aber dann existiert wieder der Fall, dass vom Benutzer ein Anruf („Mein Rechner tut’s nicht mehr“) im Help Desk erfolgt. Aber auch andere, nicht sicherheitsrelevante Änderungen, die ein Anwender – vielleicht mit den besten Absichten – vornimmt, können Probleme und Kosten nach sich ziehen.

Idealerweise werden die VPN-Clients zentral konfiguriert, entweder auf dem VPN-Router selbst oder auf einem Directory-Server. Die aktuelle Konfiguration wird dann bei jedem Verbindungsaufbau auf den VPN-Client heruntergeladen (Push Configuration). Neben der Sicherheit und dem Fehlen der Möglichkeit einer falschen Konfiguration durch den Benutzer ist damit auch eine sehr effiziente Möglichkeit der Administration gegeben, und alle Änderungen sind absolut zeitnah. Änderungen für möglicherweise mehrere tausend VPN-Clients werden nur einmal vorgenommen, und jeder Client bekommt sein Update beim nächsten Einwählen – einfacher und effizienter geht es kaum noch.

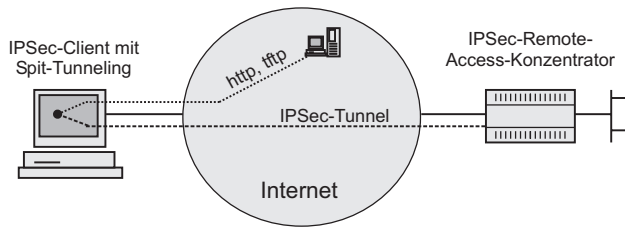


Abbildung 11.3: Split-Tunneling ermöglicht Datenverkehr an einer sicheren Verbindung vorbei ins Internet.

Eine Sache, die auf gar keinen Fall durch den Anwender beeinflussbar sein darf, ist das so genannte Split-Tunneling. Split-Tunneling bedeutet, wie Sie in Abbildung 11.3 sehen können, dass sowohl ein sicherer IPSec-Tunnel in das Unternehmensnetz existiert als auch parallel dazu Datenverkehr mit anderen Gegenstellen, z.B. im Internet, möglich ist – ein fatales Sicherheitsloch, denn damit öffnet man das Unternehmensnetz dem Internet. Das möchte man natürlich nicht, also ist Split-Tunneling in aller Regel unerwünscht, und eine gute Client-Implementierung sorgt dafür, dass, sobald sie aktiv ist, Datenverkehr nur noch durch den Tunnel möglich ist.

Von dieser Regel gibt es allerdings zwei Ausnahmen, in denen es sinnvoll ist, sowohl durch einen Tunnel als auch mit anderen Systemen zu kommunizieren. Die eine ist ein virtuelles lokales Netzwerk auf IP-Ebene, in dem verschiedene Systeme über IPSec miteinander kommunizieren und andere Systeme nicht. Die andere Ausnahme, die Sie in Abbildung 11.4 sehen, sind kleine Heim- oder Zweigbüros, in denen ein oder mehrere Rechner mit IPSec-Clients sowohl mit dem Unternehmensnetz als auch lokal kommunizieren müssen, zum Beispiel um Daten des zentralen SAP-Systems, mit dem sie über einen IPSec-Tunnel kommunizieren, lokal auf einem Netzwerkdrucker auszugeben.

Gute VPN-Systeme, Konzentratoren und die dazugehörigen Clients bieten die Möglichkeit, dass der Administrator zentral eine Liste von Netzwerken, Rechneradressen oder Domainnamen pflegt, mit denen bei freigegebenem Split-Tunneling kommuniziert werden darf. In dem Beispiel in Abbildung 11.4 würde der Administrator Split-Tunneling freigeben und als zulässigen IP-Adressbereich das lokale Klasse-C-Netzwerk 10.1.1.0 eintragen, in dem sich der PC befindet. Somit kann keine Kommunikation am IPSec-Tunnel vorbei mit dem Internet stattfinden. Innerhalb des lokalen Netzwerks können die Systeme jedoch bei einem aktiven Tunnel miteinander kommunizieren.

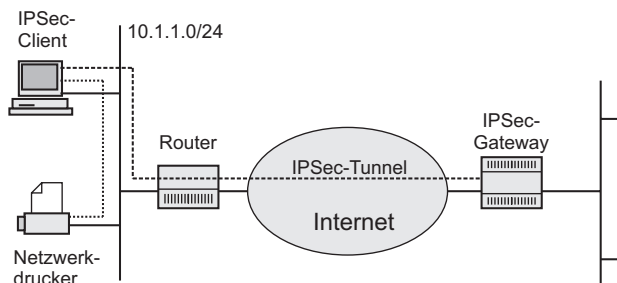


Abbildung 11.4: In kleinen Netzen kann Split-Tunneling von Vorteil sein.

Die verschiedenen Authentifizierungssysteme sollten im Sinne einer guten Handhabbarkeit durch den Benutzer von der Client-Software unterstützt werden. So sollten zum Beispiel bei einer Authentifizierung, beispielsweise mit SecurID-Token-Karten, statt eines Passwortfeldes die beiden Felder für PIN und Token angezeigt werden. Spezielle Funktionen, wie – wenn wir beim Beispiel SecurID bleiben – Next-Token-Modus oder New-PIN-Modus, sollten ebenfalls von der VPN-Client-Software erkannt und aktiv unterstützt werden können.

Das Gleiche gilt auch für den Umgang mit digitalen Zertifikaten, falls eine Authentifizierung mit digitalen Signaturen eingesetzt werden soll. Das ganze Zertifikatmanagement, inklusive der Erzeugung der asymmetrischen Schlüssel und der Certificate Requests, sollte so weit wie nur irgendwie möglich vom Endanwender abgeschirmt werden können. Es ist einem normalen Anwender, also keinem PC- und Sicherheitsexperten, nicht unbedingt zumutbar, manuelle Zertifikatsanforderungen zu erzeugen und diese mit Cut –and Paste per Datei oder E-Mails an eine CA zu übertragen oder erhaltene Zertifikate manuell zu importieren.

Leider muss ich an dieser Stelle sagen, dass die Technologie, die heute in Form verfügbarer Produkte vorliegt, dem nur unzureichend Rechnung trägt. Insbesondere die Implementierung der Internetstandards der IETF zum Thema PKIX (Public Key Infrastructure X.509) lässt bei etlichen Herstellern sehr zu wünschen übrig, und zwar sowohl bei den Herstellern der VPN-Systeme als auch bei Herstellern von PKI-Software. Und hier müssen beide Seiten mitspielen.

Die Unterstützung von Chipkarten (SmartCards) wird auf der Clientseite meist durch die Microsoft Crypto API (MS-CAPI) wahrgenommen. Da praktisch alle Hersteller von PKI-Software und -Hardware MS-CAPI unterstützen, unterstützen sie damit automatisch auch alle Anwendungen, die auf MS-CAPI aufsetzen. Auf diese Art kann zum Beispiel ein VPN-Client, der die Funktionen von MS-CAPI benutzt, jede SmartCard benutzen, die MS-CAPI unterstützt.

Falls der Remote Access über ein VPN eine geschäftskritische Stellung einnimmt, dann sind hier natürlich Maßnahmen zur Ausfallsicherheit erforderlich. Das Ganze sollte weitgehend automatisch erfolgen, der Benutzer sollte weder irgendeine Konfigurationen vornehmen noch manuell ein Backup-System anwählen müssen. Auch hier sollte eine zentrale Konfiguration erfolgen, und die VPN-Clients „lernen“ ihre Backup-Systeme bereits beim Verbindungsaufbau. Natürlich muss auch die letzte Einstellung gespeichert werden, denn wenn das primäre System nicht verfügbar ist, muss ein Ausweichsystem bekannt sein.

Um eine automatische Zuweisung von Backup-Systemen beim Verbindungsaufbau kommt man überhaupt nicht herum, wenn die zentralen VPN-Konzentratoren eine automatische Lastverteilung (Load Balancing) untereinander durchführen. Denn dann weiß man ja nie im Voraus, auf welchem VPN-Konzentrator die aktuelle IPSec-SA gerade terminiert wird, und kann somit auch keine statischen Backup-Adressen verwenden.

Einige, vor allem größere Unternehmen haben spezielle, teilweise eigene entwickelte Oberflächen für den Remote Access ihrer Mitarbeiter. Diese Applikationen pflegen Einwählverzeichnisse, Sicherheits- und Systemeinstellungen und bieten dem Anwender eine einfache, einheitliche und komfortable Anwenderoberfläche. Systemdienste, wie zum Beispiel das Microsoft-DFÜ-Netzwerk, werden von dieser Applikation automatisch gestartet, ohne dass der Anwender etwas dazu tun muss.

Wenn nun zusätzlich zum Einwähldienst noch ein VPN-Client hinzukommt, muss dieser ebenso von der Applikation gestartet werden können. Entweder bietet der Client dafür ein API (Application Programming Interface), also eine Programmierschnittstelle, oder er sollte wenigstens durch einen Kommandozeilenaufruf oder einen Systemaufruf gestartet und mit entsprechenden Parametern, z.B. Zieladresse, User-ID, PIN und Token, versorgt werden können, um dann möglichst im Hintergrund abzulaufen.

Hier hat vor allem Microsoft einen hohen Anteil, dicht gefolgt von FreeS/WAN. Microsoft hat insbesondere bei seinen aktuellen VPN-Clients (L2TP/IPSec) sehr viele Standards implementiert, so dass es den Herstellern von VPN-Routern relativ leicht gefallen ist, ihre Hardware für die Terminierung dieser VPN-Clients zu konfigurieren.

Bei der Auswahl von VPN-Routern, die hauptsächlich mit Microsoft L2TP/IPSec-Clients arbeiten, ist, neben reichlich Performance, besonders darauf zu achten, dass im L2TP-Teil mindestens MS-CHAPv2 als Authentifizierungsverfahren unterstützt wird. Besser noch wäre der Support von EAP und digitalen Signaturen, aber da sieht es selbst bei einigen ansonsten guten Herstellern ziemlich trübe aus.

Der Vorteil dieser Art von Clients ist die Tatsache, dass man sie nicht verteilen muss, denn sie sind Bestandteil bei allen neueren Microsoft-Betriebssystemen (seit Windows 2000). Für ältere Systeme (Windows 98, ME und NT) sind sie nachinstallierbar.

Der Nachteil ist einmal der, dass man mit solchen VPN-Clients die ganzen Add-On-Features, die VPN-Hersteller seit Jahren entwickelt haben, nicht einsetzen kann. Und das sind viele und wichtige Features, vor allem solche, welche die Betriebskosten deutlich reduzieren.

Zum anderen sieht man bei Microsoft das Thema VPN offensichtlich nicht als Kerngeschäft, und entsprechend trübe sieht es mit der VPN-Funktionalität aus: AES: Fehlanzeige, ECC: Fehlanzeige, automatisches Backup/Recovery: Fehlanzeige, automatische Konfiguration: Fehlanzeige; die Liste ließe sich ziemlich lang fortsetzen. Dies dürfte mit ein Grund sein, warum Microsoft-Clients, und das gilt auch haargenau so für FreeS/WAN, sich bisher selten gegen die herstellerspezifischen Clients haben durchsetzen konnten. Oft sind es „strategische“, von wenig Sachkenntnis oder gar vorausgehenden Analysen getriebene Entscheidungen, die 80% des Potenzials moderner VPN-Router brachliegen lassen, nur um mit Microsoft VPN-Clients arbeiten zu können.

Gerade in kleinen und mittleren Netzen, in denen Mobilität nicht geschäftskritisch ist, haben sie sich jedoch einen festen Platz erobern können. In derartigen Szenarien werden die wenigen VPN-Verbindungen meist auch nicht auf VPN-Routern, sondern auf einem Windows 2000 oder Windows 2003 Server terminiert.

Im Fall von sehr kleinen Außenstellen oder Heimarbeitsplätzen, die man kurz unter dem Oberbegriff SOHO (Small Office Home Office) zusammenfasst, setzt man spezielle Systeme ein, die gezielt auf diesen Einsatzbereich zurechtgeschnitten sind. Sie benötigen meist nicht alle Funktionen der „großen Brüder“, dafür oft zusätzliche Funktionen für den SOHO-Bereich.

Üblicherweise benutzt man dafür heutzutage spezielle VPN-Router, oft mit eingebauten DSL-Modems, die sich in normale VPN-Router einwählen. Dabei gilt es, vor allem im Hinblick auf den VPN-Einsatz, zwei grundsätzlich verschiedene Einsatzszenarien zu unterscheiden:

Der VPN-Router verhält sich gegenüber dem zentralen VPN-Router genau wie ein normaler Benutzer und bekommt insbesondere auch eine dynamische IP-Adresse zugewiesen. Der Router „versteckt“ das kleine SOHO-Netzwerk, indem er One-to-Many-NAT einsetzt. Diese Form von NAT (Network Address Translation) setzt dynamisch eine IP-Adresse auf mehrere private Adressen um. Hierbei können Verbindungen nur aus dem SOHO-Netz heraus aufgebaut werden.

Der VPN-Router verhält sich wie ein normaler Router. Er hat eine statische IP-Adresse, und sein lokales Netz kann mittels Routing-Updates im Unternehmensnetz verteilt werden.

Der erste Fall bereitet mit einem geeigneten IPSec-Gateway keine nennenswerten Probleme. Hier gibt es Systeme, die mit entsprechend starker Sicherheit (z.B. IPSec-ESP mit Triple-DES und SHA-1) ausgestattet sind und durch den IKE Aggressive Mode auch von einem Service Provider per PPPoE eine dynamische IP-Adresse zugewiesen bekommen können. Spezielle NAT-Implementierungen oder die Verwendung virtueller IPSec-Interfaces ersetzen auch die One-to-Many-NAT-Funktionalität normaler ADSL-Router.

Im zweiten Fall wird es allerdings etwas teurer, wenn die Funktionalität vollständig übernommen werden und die Lokation vor allem immer erreichbar sein muss. Denn hier muss der Service Provider etwas tun, was er sonst nicht tut und vor allem auch überhaupt nicht mag: Er muss eine statische IP-Adresse vergeben. Natürlich erfolgt so etwas nicht umsonst, aber man ist trotzdem noch immer relativ preisgünstig an eine Außenstelle angebunden.

VPN-Router in diesem Bereich müssen einige bestimmte Mindestcharakteristika vorweisen:

Systeme, die direkt neben dem Arbeitsplatz (Home Office) betrieben werden, müssen lautlos sein. Kein Lüfter soll hier Lärm erzeugen. Abgesehen von gesetzlichen Vorgaben kann man bei Dauerberieselung durch Geräte- oder CPU-Lüfter auch nicht vernünftig arbeiten.

Die Systeme müssen, vor allem wenn es sich um große Netze handelt, preisgünstig sein.

Die Systeme müssen einfach, ohne Fachpersonal vor Ort in Betrieb zu nehmen sein. Sie müssen „remote“ konfigurierbar und wartbar sein.

Kleinere Büros (keine Heimbüros) mit mehr als einem Mitarbeiter müssen in der Regel Redundanzmaßnahmen treffen. Dies ist in der Regel ein ISDN-Dial-Backup im VPN-Router oder sogar ein zweites System.

VPN-Router müssen unter Umständen mit dynamischen WAN-IP-Adressen zurechtkommen.

Je nach Firmenstrategie muss im VPN-Router entweder ein lokaler DHCP-Server arbeiten oder eine DHCP-Relay-Funktion implementiert sein.

Die Sicherheitsfunktionen von IPSec müssen konform zur Security Policy des Unternehmens sein. Ein nicht selten gemachter Fehler ist es, hier aus Kostengründen Ausnahmen zu machen, sei es bei der Verschlüsselungsstärke oder der Authentifizierung.

Oft wird gewünscht, dass alle Funktionen (ADSL, ISDN-Backup, LAN-Switch, VPN-Funktionen usw.) in einem einzigen Gerät vereint sind. So etwas reduziert den Verhaar von Kabeln, Geräten und Steckernetzteilen erheblich und eliminiert dadurch auch zusätzliche Fehlerquellen.

Allerdings gibt es auch gute Gründe, dies nicht zu tun. Ein eingebautes ADSL-Modem schließt eine spätere Erweiterung auf SDSL aus. Ein Ethernet-Interface mit einem externen ADSL-Modem nicht, denn hier würde der Carrier oder Provider einfach nur das ADSL- durch ein SDSL-Modem austauschen müssen.

Auch der integrierte LAN-Switch ist nicht selten falsch dimensioniert: Entweder braucht man nicht alle Ports, oder es sind zu wenig.

Bei solchen Systemen gelten völlig andere Kriterien als bei Remote-Access- oder SOHO-Systemen. Denn hier werden teilweise große Standorte mit Hunderten oder Tausenden von Mitarbeitern verbunden, Rechenzentren angebunden oder Zweigstellennetze aufgebaut. Hier müssen Dinge wie hohe Verfügbarkeit, hoher Durchsatz, Skalierbarkeit, Quality of Service und Secure Routing in den Anforderungskatalog derartiger Komponenten einfließen. Meist werden Router mit Frame-Relay-, ATM- oder Festverbindungen durch VPN-Router ersetzt. Hierbei dürfen keine kritischen Nachteile entstehen, denn solche Verbindungen sind in der Regel unternehmenskritisch.

Die VPN-Gateways müssen an die benötigte oder zu erwartende Last angepasst sein. Insbesondere im Fall von IPSec-ESP und starker Verschlüsselung benötigt man schon einiges an Systemleistung. Als Faustregel kann man sagen, dass der Systemdurchsatz mindestens so groß sein muss wie die Zugangsleitungen. Wenn also eine Anbindung mit einer E3-Schnittstelle zu einem ISP erfolgt und dieser Durchsatz auch vertraglich garantiert wurde, dann sollte dieses IPSec-Gateway mindestens einen Durchsatz von 34 MBit/s bei allen Paketgrößen ab 160 Byte haben – und das vor allem bei der maximal benötigten Anzahl von Tunneln, mit AES, SHA1 und IPSec Compression.

Falls Sie sich nicht im Klaren darüber sind, wie sehr das VPN im Laufe der Zeit wachsen wird, können Sie auch Systeme einsetzen, die mit weiteren Schnittstellenkarten und Hardware-Beschleunigern aufrüstbar sind.

Das Bandbreitenmanagement ist eine Methode, um bestimmten virtuellen Verbindungen innerhalb einer physikalischen Verbindung bestimmte Bandbreiten zuzuteilen beziehungsweise diese zu begrenzen. Damit soll verhindert werden, dass bestimmte Verbindungen anderen Verbindungen benötigte Bandbreite wegnehmen.

Gutes Bandbreitenmanagement erlaubt die Zuteilung von maximalen Bandbreiten und so genannten Burst Rates, die festlegen, um wie viel dieser Wert überschritten werden darf, wenn die gewünschte Bandbreite zur Verfügung steht. Das Ziel des Ganzen ist es, auf der einen Seite bestimmten Tunneln eine Mindestbandbreite zu garantieren, aber auf der anderen Seite auch möglichst alle verfügbare Bandbreite auszunutzen.

Manche VPN-Router unterstützen ein so genanntes Load Balancing, also eine Lastverteilung zwischen den Systemen. Damit kann man, falls erforderlich, die Systemleistung durch zusätzliche Geräte erhöhen und kommt auch gleichzeitig in den Genuss einer gewissen Redundanz. Allerdings haben einige dieser Vertreter IPSec und IKE etwas „aufgeweicht“, so dass hier begründete Sicherheitsbedenken bestehen.

Eine andere Alternative sind externe Load-Balancing-Systeme, die in der Lage sind, IPSec- und IKE-Verbindungen zu erkennen und intelligent auf die am wenigsten belasteten VPN-Router zu verteilen. In der Regel muss man dafür zwei Systeme einsetzen, eins vor und eins hinter den VPN-Router. Letzteres muss dafür sorgen, dass die zurücklaufenden Pakete wieder auf den korrekten VPN-Router gelangen. Der Vorteil einer solchen Lösung ist die extrem hohe Skalierbarkeit und die Tatsache, dass auf den Routern keine Modifikationen an IPSec und IKE notwendig sind.

In großen VPN ist die Skalierbarkeit und auch die Migrationsfähigkeit ein entscheidender Faktor bei der Systemauswahl. Für verschieden große Standorte benötigt man unterschiedlich leistungsfähige Systeme und oft auch Systeme, die in ihrer Leistungsfähigkeit erweiterbar sind. Allerdings müssen die Funktionen und der Grad der Sicherheit auf allen Systemen identisch sein, egal ob es sich um ein Gigabit-System oder ein System zum Anschluss an 2 MBit/s SDSL handelt.

Insbesondere in größeren Standorten kann es notwendig sein, die Leistung der System erhöhen zu müssen. Hier bieten sich VPN-Router an, die mit einem oder mehreren Beschleunigern erweitert werden können.

Ab einer gewissen Leistungsfähigkeit sind VPN-Router auszuwählen, die mit verschiedenen Schnittstellen ausgerüstet werden können. So sollten neben 10/100 MBit/s Ethernet auch Gigabit-Module (Kupfer und Glasfaser) sowie WAN-Module für E3 (HSSI), E1 oder X.21 verfügbar sein. Auch analoge Modem- und ISDN-Adapter machen Sinn, falls man Out-Band-Management benötigen sollte.

Bei ganz kleinen Systemen reichen sinnvolle Festkonfigurationen aus, hier machen E3 oder Gigabit ohnehin keinen Sinn.

Es kann durchaus Sinn machen, in einem VPN-Router auch Frame Relay auf seriellen Interfaces (E1, HSSI, X.21) einzusetzen. Frame Relay ist zwar ein Layer-2-VPN-Protokoll ohne Sicherheitsfunktionen, aber es bietet pro virtueller Verbindung (VC) einen zumindest rudimentären Quality of Service (vgl. Kap. 10). Falls Sicherheit benötigt wird, kann man selbstverständlich IPSec über Frame Relay transportieren. Außerdem bieten gute

Frame Relay Router zusätzlich die Funktion, größere Pakete generell zu fragmentieren. Das hat den Sinn, auf langsamen Schnittstellen die Verzögerung durch die Serialisierung niedrig zu halten und gleichzeitig dafür zu sorgen, dass die Schnittstelle nicht zu lange durch große Pakete blockiert wird und kleine Pakete (VoIP) mit höherer Priorität schneller ausgegeben werden können.

Redundanz und Ausfallsicherheit sind Anforderungen, die aus keinem Netzwerk mehr wegzudenken sind. Entsprechend muss auch ein VPN ausgelegt werden, will man keinen Rückschritt machen.

Hier setzt man am besten auf altbewährte Methoden und verwendet entweder Systeme, die voll redundant ausgelegt sind, oder man schafft Redundanz durch mehrere Geräte. Zwei Geräte allein schaffen allerdings noch keine Redundanz, es müssen auch Methoden existieren, um zwischen den beiden Systemen automatisch und schnell umzuschalten, oder im Netzwerk selbst muss eine entsprechende Intelligenz implementiert sein, die dies im Bedarfsfall tut.

Der erste Fall, nämlich entsprechend ausgelegte Systeme mit passiver Backplane und doppelter Auslegung aller Module und Stromversorgungen, ist in aller Regel eine sehr teure Angelegenheit und hat obendrein noch ein immanentes Problem: Dadurch dass alles in einem Chassis konzentriert ist, kann ein massiver physischer Schaden oder ein Ausfall der Zuleitung trotzdem das komplette System lahm legen.

Die zweite Lösung, zwei identische Systeme einzusetzen und diese am besten räumlich getrennt zu betreiben – vielleicht noch mit zwei verschiedenen Zuleitungen zu unterschiedlichen POPs –, ist eine andere, meist günstigere Lösung. Hierfür gibt es auch ein standardisiertes Verfahren, das zwei Router zu einem redundanten System vereinigen können. Das VRRP (Virtual Router Redundancy Protocol) zum Beispiel wäre solch ein Verfahren. Hierbei ist ein System der VRRP-Master, und sobald dieser ausfällt, übernimmt das zweite Gerät dessen Funktion und auch dessen Netzwerkadressen.

Allerdings ist VRRP nicht das Optimum, denn in seiner standardisierten Version wird nur erkannt, wenn vom lokalen Interface des Master-Routers keine VRRP-Hello-Pakete mehr ankommen. Damit wird nur der Ausfall dieser Schnittstelle, der Ausfall von mit dieser assoziierten Funktionen oder des gesamten Systems erkannt. Das schon einiges – aber viel zu wenig. Was nicht erkannt wird, ist der Ausfall anderer Schnittstellen, zum Beispiel öffentlicher Schnittstellen zum Internet oder virtueller Schnittstellen (Tunnel). Und auf deren Ausfall muss ebenfalls reagiert werden.

Also muss der Standard erweitert werden, und das ist im Moment nur herstellerspezifisch möglich, wobei allerdings immer noch Kompatibilität zum Standard bestehen sollte. Erweiterungen sind in der Regel so genannte Critical Interfaces oder Critical Interface Groups zur Erweiterung der überwachten Schnittstellen, IKE Dead Peer Detection zum Erkennen von „toten“ Tunneln oder Gegenstellen und VRRP-Disable zum zwangsweisen Deaktivieren bestimmter Schnittstellen, wenn ein System seinen Masterstatus verliert. Mit diesen Erweiterungen sind echte Hochverfügbarkeitsanwendungen realisierbar.

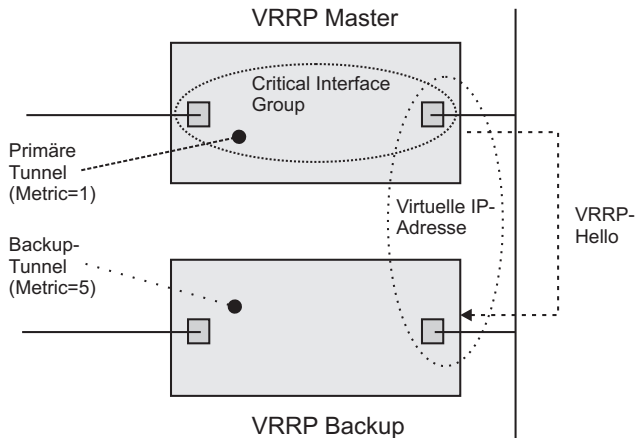


Abbildung 11.5: VRRP hat von einigen Herstellern nützliche Erweiterungen erfahren.

Allerdings bedingt der Ausfall eines VPN-Routers in solchen Szenarien, dass die IPSec-Verbindungen (IKE, IPSec) neu aufgebaut werden müssen.

Ein andere Alternative ist in großen Umgebungen mit externen Load-Balancing-Systemen möglich. Denn diese können implizit auch eine Hochverfügbarkeit garantieren, indem auf einen ausgefallenen oder problembehafteten VPN-Router einfach keine Verbindungen mehr gelegt werden.

In Kapitel 9 wurden bereits die verschiedenen Arten von Quality of Service (QoS) angesprochen, so dass wir uns hier auf ihren praktischen Einsatz konzentrieren können.

Eine flussbasierte Ende-zu-Ende-Qualität mit RSVP wird meist nicht von Internet-Routern unterstützt, so dass zwischen den beteiligten Systemen Strecken existieren, die eine Weiterleitung der Pakete nach dem Best-Effort-Prinzip vornehmen. Also macht RSVP eigentlich keinen Sinn.

Nichtsdestotrotz bieten manche Hersteller RSVP an, um mindestens in den VPN-Routern selbst eine flussbasierende Qualitätssteuerung zu ermöglichen und sich gegenseitig Ressourcen zu reservieren.

Differentiated Services (DiffServ) hat sich als Standard für QoS im Internet bzw. an dessen Grenze, also am Schnittpunkt zwischen Provider und Kunde, weitgehend durchgesetzt. Auch im Intranet ist es das QoS-Protokoll der Wahl, und viele LAN-Switches unterstützen bereits DiffServ. Es bietet sich daher an, die eigene QoS-Strategie über die Grenzen der eigenen Intranets hinaus bis in das Internet zu erweitern.

Allerdings ist im Fall von IPSec eine Sache zu beachten, und zwar bevor man sich ein VPN-Gateway beschafft: Denn hier muss das VPN-Gateway als DiffServ-Edge-Router die IP-Pakete entsprechend der QoS-Policy markieren, falls das nicht bereits an anderer Stelle im Netzwerk erfolgt ist. Denn nach dem Tunneling und der Verschlüsselung der Daten kann der DiffServ-Edge-Router eines Providers keine Markierungen mehr aufgrund von Proto-

koll- oder Portnummern des Originalpakets setzen, da es verschlüsselt ist. Das Einzige, was der Provider erkennen kann, ist das DS-Byte im äußeren IP-Header. Und in dieses Byte wurde vor der Verschlüsselung des Originalpakets dessen DS-Byte kopiert.

Aber Vorsicht, wie in Kapitel 9 besprochen, arbeitet DiffServ in fast allen Systemen auf Interface-Basis und nicht auf der Basis von virtuellen Verbindungen (Tunneln), was in bestimmten Szenarien unangenehme Folgen haben kann.

Diese Funktion ist wichtig, wenn ein System überbucht wird. Hier wird aufgrund von konfigurierbaren Prioritäten festgelegt, wer eine Verbindung aufbauen darf und wer nicht.

Die zweite Funktion von Admission Control ist die Überwachung der Systemlast und das Verhindern neuer Verbindungen, falls diese die bereits aufgebauten Verbindungen beeinträchtigen kann.

DiffServ berechnet die Größe seiner Warteschlangen in der Regel aufgrund der Interface-Geschwindigkeit und fester oder konfigurierbarer Qualitätswerte wie zum Beispiel der Verzögerung. Mit anderen Worten, bei einer maximalen Verzögerung von m Millisekunden und einer Interface-Geschwindigkeit von n MBit/s darf die Priority Queue maximal x Byte groß sein. Nun kann es aber sein, dass die tatsächliche Bandbreite gar nicht der Interface-Geschwindigkeit entspricht. Ein typisches Beispiel ist ADSL, hier erkennt das Ethernet-Interface des VPN-Routers 10 MBit/s, aber in Upstream-Richtung sind es in Wirklichkeit nur 128 KBit/s auf der DSL-Leitung. Somit wären zum einen die Warteschlangen viel zu lang, und der Scheduler würde ebenfalls viel zu schnell Pakete ausgeben, von denen dann fast jedes verworfen würde. Somit ist die Möglichkeit, Schnittstellengeschwindigkeiten unabhängig ihres auf der Media-Ebene erkannten (Auto-negotiation, synchrone Taktsignale usw.) Wertes zu konfigurieren, extrem wichtig.

Da VPN-Router praktisch immer in Ethernet-LAN angeschlossen werden, ist das Mapping des DSCP auf Ethernet Priority (IEEE802.1p) wichtig. Das Mapping sollte frei konfigurierbar sein.

Eine andere optionale Möglichkeit ist der umgekehrte Weg, bei eingehenden Ethernet-Frames den DSCP zu setzen. Auch dies sollte frei konfigurierbar sein.

Routing, das Sorgenkind einiger VPN-Hersteller. Manche ignorieren es und reduzieren ihre Produktpalette auf kleinste Netze mit statischem Routing. Andere schicken Routing-Updates an IPSec-Verbindungen vorbei durch das Internet.

Was ist das Problem? Ganz einfach, IPSec ist ein Unicast-Protokoll, ebenso wie IKE. Und Routing Updates werden per Broadcast oder Multicast verschickt.

Für dieses Problem gibt es natürlich eine Lösung, nämlich die Einkapselung der Multi- oder Broadcast-Pakete in Unicast-Pakete, um diese dann durch einen IPSec-Tunnel zu übertragen. Leider gibt es hier keinen verbindlichen Standard, Nortel benutzt zum Beispiel IPIP-Encapsulation, Cisco GRE- und wahlweise auch IPIP-Encapsulation. Somit wären hier in diesem Beispiel heterogene IPSec-VPN mit dynamischem Routing und IPIP-Encapsulation im Bereich des Möglichen (und wurden auch schon getestet).

Es versteht sich von selbst, dass das Routing nur auf privaten und virtuellen Schnittstellen aktiv sein darf – unter keinen Umständen aber auf öffentlichen Interfaces. Mit einer einzigen Ausnahme, nämlich BGP (s.u.).

Statisches Routing hat in vielen Anwendungsfällen nach wie vor seine Daseinsberechtigung. Und zwar dort, wo man eben statische Verhältnisse hat oder die automatische Konvergenz dynamischer Routing-Protokolle nicht notwendig ist. Falls beispielsweise bei Ausfall einer Verbindung eine alternative Strecke durch andere Maßnahmen in einer Sub-IP-Schicht, z.B. Dial-Backup auf Layer 2, aktiviert wird, benötigt man hierfür kein Routing, denn der Weg zu dem betreffenden Netz ist aus Sicht von IP der gleiche geblieben.

Statisches Routing wird auch in gut durchdachten Designs sehr oft in Verbindung mit anderen Protokollen wie OSPF oder RIPv2 eingesetzt. Oder es ist sogar, z.B. bei IPSec, implizit im Standard, hier der statischen IPSec Security Policy, festgelegt.

RIPv2 ist im RFC 1723 beschrieben und ist der klassische Vertreter der so genannten Distance-Vector-Protokolle, die ihre Routing-Entscheidungen ausschließlich aufgrund der Anzahl der so genannten Hops (Router, Gateways) zwischen jedem Ziel treffen. Das zum Vorgänger RIPv1 um viele Funktionen und Verbesserungen erweiterte Protokoll bietet unter anderem:

- Variable Subnetzmasken (VLSM, Variable Length Subnet Mask)

- Übertragung der Netzmaske im Routing-Update

- Authentifizierung (Passwort)

- Multicasts anstelle von Broadcasts

- In verschiedene Routing-Domänen (die allerdings nicht miteinander kommunizieren können) unterteilbar

- Route-Tags, um zu erkennen, ob und von welchem Routing-Protokoll ein RIPv2 Update entstammt

- Kann optional „Triggered Updates“ versenden

Die Nachteile von RIPv2 wachsen in der Regel mit dem Umfang des Netzwerks, denn hier, in sehr großen Netzwerken, werden einige dieser Limitierungen zum Problem:

- Langsame Konvergenzzeiten von bis über 10 Minuten

- Begrenzte Netzwerkgröße durch den auf 15 limitierten Hop-Count

- Trotz verschiedener Maßnahmen immer noch die Gefahr von Routing Loops

- Die einzige Metrik ist die Anzahl der Gateways (Hops) zwischen zwei Netzwerken.

- In großen Netzen entsteht ein hoher Overhead durch die periodische Übertragung großer Tabellen.

Trotzdem sollte man RIPv2 durchaus in Erwägung ziehen, falls man ein relativ überschaubares Netzwerk betreibt. Denn es ist ein relativ einfaches Protokoll, das wenige Ressourcen in den Routern benötigt und dessen Administration auch vergleichsweise einfach ist. Bei geschicktem Design der Routing-Architektur kann RIPv2 eine sinnvolle Wahl sein, auch unterstützend zu anderen Routing-Protokollen.

Das OSPF-Protokoll nach RFC 2178 erhielt seinen Namen (Open Shortest Path First) von dem zugrunde liegenden SPF-Algorithmus (Shortest Path First) zur Berechnung der Routing-Tabellen. Es ist der bekannteste Vertreter der so genannten Link-State-Protokolle und ist, ebenso wie RIPv2, ein Interior-Gateway-Protokoll. OSPF wurde aus dem Wunsch heraus geboren, ein Routing-Protokoll für sehr große IP-Netze zu haben, das die Nachteile von RIP in diesem Bereich nicht aufweist.

Die wichtigsten Features von OSPF sind:

Sehr schnelle Konvergenz im Bereich weniger Sekunden nach Eintritt einer Änderung in der Topologie, wie Ausfälle von Routern oder Verbindungen, Hinzufügen neuer Netze usw.

In den Routing-Updates werden nur relevante Informationen zur Änderung der Topologie und nicht die komplette Routing-Tabelle übertragen. Außerdem werden die Updates nicht an alle Router verschickt.

Als Metrik können so genannte Routing-Kosten konfiguriert werden, die sich aus verschiedenen, gewichteten Faktoren wie Bandbreite, Verzögerung, Auslastung und Kosten von Leitungen zusammensetzen.

OSPF arbeitet mit einem zweistufigen Hierarchiemodell und teilt ein Netz in verschiedene Bereiche (Areas) auf. So unterscheidet man zwischen Intra-Area- und Inter-Area-Routing, je nachdem, ob nur bereichsinterne Topologieinformationen übermittelt werden oder ob ein Paket über Bereichsgrenzen hinweg geroutet werden muss.

Zur Authentifizierung wurden weitere Verfahren wie MD5 oder Authentication Header (AH) eingeführt, um den Sicherheitsstand weiter zu erhöhen.

OSPF hat jedoch auch durchaus seine Schattenseiten, die teilweise sogar gerade ein Resultat der vielen nützlichen Funktionen und Parameter in diesem Routing-Protokoll sind:

Hohe Anforderungen an die Leistungsfähigkeit der Router hinsichtlich Prozessor und Arbeitsspeicher. Das muss man bei der Auswahl seiner Systeme (Router, Switch, VPN-Gateway usw.) unbedingt beachten, auch hinsichtlich des Wachstums eines Netzwerks.

Die vielen Funktionen und die Arbeitsweise von OSPF beeinflussenden Parameter machen die Konfiguration nicht gerade trivial.

Leider kennt OSPF nur zwei Hierarchiestufen, mehrere wären, vor allem in größeren Netzen, sinnvoll.

Alles in allem ist OSPF ein sehr gutes Protokoll für größere VPNs. Ich kenne VPNs mit über 300 Standorten, die dabei erfolgreich OSPF für ihr VPN-Routing einsetzen.

Das Border Gateway Protocol (BGP) ist ein Exterior Gateway Protocol (EGP), das bis auf wenige Ausnahmen nur im Internet und fast niemals in Unternehmensnetzen Verwendung findet. Es gibt zwei Ausnahmen: extrem große Unternehmensnetze auf eigener Infrastruktur und die Anbindung eines Unternehmensnetzwerks über zwei verschiedene Provider an zwei verschiedenen Punkten. In letzterem Fall braucht man (außer beim Remote Access) BGP, und zwar auf dem öffentlichen Interface.

Die andere Alternative sind große VPNs mit verschiedenen Netzen und Domänen, in denen BGP eingesetzt werden sollte.

Die Lösungsbeispiele in diesem Kapitel sollen die theoretischen Erwägungen der Abschnitte nun mit Leben füllen. Es wurde dabei gezielt nicht auf Sonderlösungen oder spezielle Anwendungsfälle abgezielt, sondern typische Unternehmensszenarien wurden ausgewählt.

In den verschiedenen Beispielen werden einleitend die verschiedenen Anwendungsszenarien und Anforderungsprofile so weit beschrieben, wie es für das Verständnis der Lösungen notwendig ist. Die Beispiele entstammen realen Projekten.

In diesem Buch wurde bisher größter Wert auf Herstellerunabhängigkeit gelegt. Dies ist in diesem Kapitel nur sehr eingeschränkt möglich. Aufgrund meiner Tätigkeit für die Firma Nortel habe ich naturgemäß hauptsächlich mit deren VPN-Komponenten zu tun. Ich habe das Kapitel jedoch weitgehend so angelegt, dass die Lösungen prinzipiell auch mit Systemen anderer Hersteller zu realisieren sind.

Deshalb sei an dieser Stelle Folgendes gesagt: Die Nennung oder Nichtnennung eines Herstellers oder Produkts oder die Häufigkeit der Erwähnung bestimmter Hersteller oder Produkte darf nicht als Wertung verstanden werden.

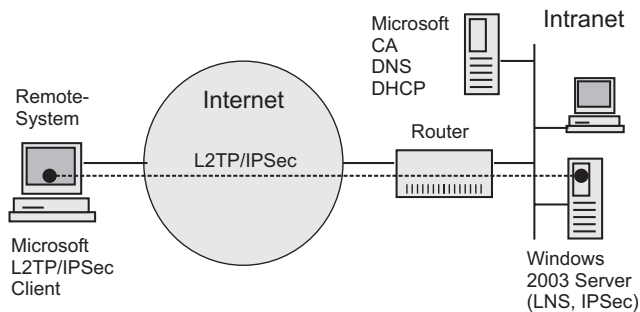


Abbildung 11.6: Die preisgünstige kleine VPN-Lösung in Software

Ein typischer Fall liegt vor, wenn Remote Access bisher über klassischen Dial-Up mit ISDN, analogen Modems oder GSM erfolgte. In diesem Fall existiert in der Regel schon eine Benutzerverwaltung, entweder in einem RADIUS-Server oder in einem Betriebssystem, heutzutage meist Microsoft Windows 2000. Die Einwahl erfolgte dabei entweder über eigen betriebene Remote-Access-Konzentratoren (RAS) oder über L2TP, wenn die physikalischen Verbindungen bei einem Carrier terminiert wurden.

Nehmen wir für unser Beispiel einmal folgende Eckwerte der bisherigen Lösung an, die nicht untypisch für ein mittelständisches Unternehmen sind:

Eigener RAS mit einem ISDN-PRI

Anzahl gleichzeitiger Benutzer: 30 (ISDN, analoog, GSM)

Anzahl angelegter Benutzer: 250 (Überbuchung etwa 1:8)

Applikationen: E-Mail, Filesharing, Datenbankanwendungen, Browser

Access-Bandbreite: 1.920 KBit/s (ISDN-PRI)

Authentifizierung über Radius (Windows 2000 IAS als Front-End zu Microsoft Active Directory)

Die VPN-Lösung wird primär gewählt, weil sich damit die Kosten für den Remote Access drastisch reduzieren lassen. Außerdem soll die Lösung eine höhere Flexibilität als bisher bieten, da die Mitarbeiter auch über GPRS, WLAN-Hotspots und vor allem auch ADSL arbeiten können. Die Anzahl der Benutzer, die gleichzeitig arbeiten können, soll konstant bleiben oder ein wenig größer werden.

Als Internetzugang ist ein 2-MBit-SDSL-Anschluss geplant, intern wird Ethernet eingesetzt, die Anbindung erfolgt über Fast Ethernet. Die Sicherheit ist aufgrund der Übertragung über das Internet zu einem Thema geworden, und starke Verschlüsselung ist ein K.o.-Kriterium. Alle Applikationen sollen mit der bisher gewohnten Qualität benutzt werden können.

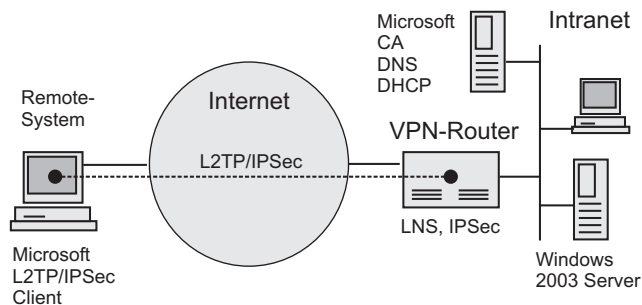


Abbildung 11.7: Mit einem VPN-Router kann man den Server entlasten und die Anzahl der Clients erhöhen.

Daraus kann man im ersten Ansatz folgende Anforderung an ein VPN ableiten:

Die VPN-Clients müssen unterschiedlichste Access-Technologien unterstützen, aus diesem Grund kommt praktisch nur ein Layer-3-VPN, also IPSec in Frage. L2TP/IPSec fällt auch unter diese Kategorie.

Der VPN-Router muss auf der privaten Seite ein Fast Ethernet Interface und auf der öffentlichen Seite (SDSL-Modem) ebenfalls einen passenden Ethernet-Anschluss besitzen. Der Durchsatz mit starker Verschlüsselung muss auch bei kleineren Paketen mindestens 2 MBit/s in beide Richtungen (full duplex) betragen. Die Benutzer müssen mit ihrer bisherigen User-ID über RADIUS authentifiziert werden.

Hier bieten sich drei Lösungen an, die unter Umständen ineinander überführt werden können.

Man realisiert das VPN mit Microsoft-Bordmitteln. Ein Windows 2000 Server dient als VPN-Server, und auf den Client-Rechnern setzt man L2TP/IPSec ein. Für dieses Szenario muss allerdings der Microsoft Certificate Service auf einem der Server installiert und eingerichtet werden, da die Rechnerguthentifizierung mit IPSec und digitalen Signaturen erfolgt. Ebenso müssen die Maschinenzertifikate auf die Rechner verteilt werden. Eine Authentifizierung mit Pre-Shared Key ist zwar technisch möglich, jedoch aufgrund der Schlüsselverteilung nicht handhabbar, nicht einmal in kleinen Netzen.

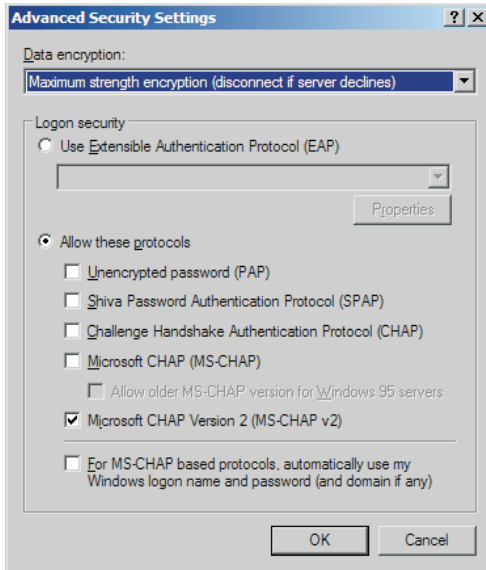


Abbildung 11.8: Man sollte generell auf starke Verschlüsselung setzen und in den Verbindungseigenschaften möglichst CHAP oder MS-CHAP Version 2 konfigurieren.

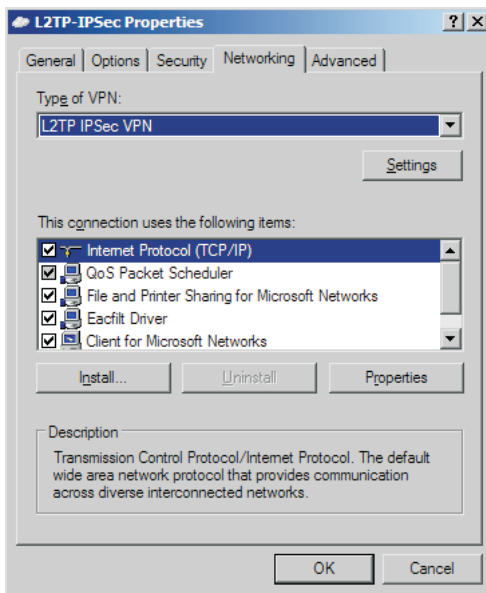


Abbildung 11.9: Wichtig ist, in den Eigenschaften der Verbindung explizit L2TP IPSec VPN zu konfigurieren.

Falls sich die Windows-Server als nicht leistungsfähig erweisen sollten, um auch noch als VPN-Server zu dienen, kann man stattdessen einen kleineren VPN-Router einsetzen. Um nicht vom Regen in die Traufe zu kommen, darf man auf gar keinen Fall einen „normalen“ kleinen Router einsetzen, der per Software-Update plötzlich auch IPSec und Verschlüsselung beherrschen soll. Von solchen Systemen darf man keine Performance-Wunder erwarten. Und 2 MBit/s Minstdurchsatz bei 200-Byte-Paketen heißt normalerweise 6–10 MBit/s im Datenblatt der Hersteller – denn die messen meist nicht mit 200-Byte-Paketen. Ein echter VPN-Router, solche Systeme gibt es im 1.000- bis 2.000-Euro-Bereich, beherrscht auch noch Datenkomprimierung, welche die effektive Bandbreite noch weiter erhöhen kann, und verschiedene Sicherheits-Features, die man in Windows bislang vermisst. Das ist bitte nicht als Kritik an Microsoft zu verstehen, denn deren Kerngeschäft liegt einfach in anderen Bereichen.

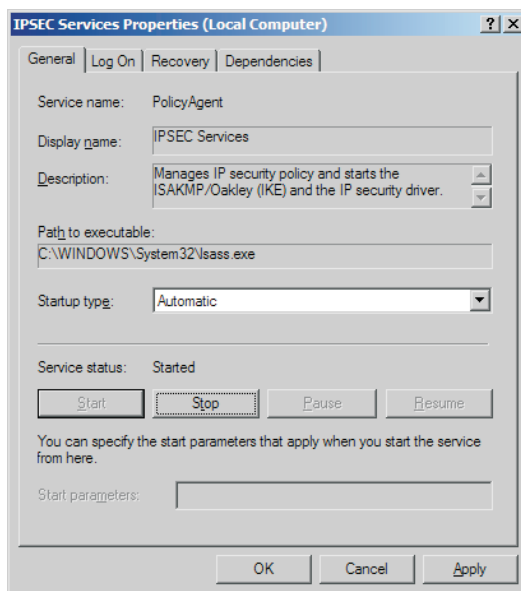


Abbildung 11.10: Der IPSec Policy Agent muss als Systemdienst gestartet sein, ansonsten ist keine Verschlüsselung möglich.

Da hier ebenfalls L2TP/IPSec-Clients eingesetzt werden können, benötigt man auch hier den Microsoft Certificate Service und die Verteilung von Maschinenzertifikaten. Der VPN-Router muss natürlich L2TP/IPSec unterstützen. Bei dessen Konfiguration muss man darauf achten, dass Microsoft nicht alle Verschlüsselungs- und Schlüsselerzeugungsverfahren beherrscht und seine IPSec-Konfiguration entsprechend gestaltet. Man kann z.B. im Windows 2003 Server lediglich DES-40-Bit, DES-56-Bit oder Triple-DES mit Diffie-Hellman 1.024 oder 2.048 Bit auswählen.

Im Remote-Rechner kann der L2TP/IPSec-Client am einfachsten über den Connection Wizzard konfiguriert werden. Als Zieladresse muss die IP-Adresse des öffentlichen Interface des VPN-Routers oder dessen DNS-Name eingegeben werden.

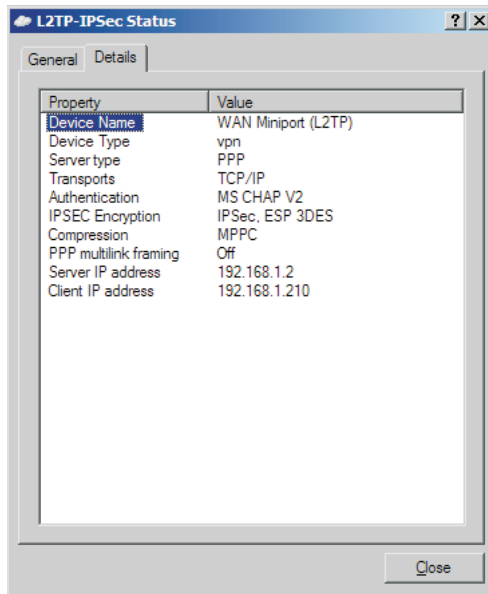


Abbildung 11.11: Nach einem erfolgreichen Verbindungsaufbau kann man sich die aktuellen Einstellungen im Statusfenster anschauen.

Diese Lösung bringt vor allem einen eigenen Client mit, der mit IPSec im Tunnel Mode arbeitet und keine Maschinenzertifikate benötigt. Damit braucht man auch keinen Microsoft Certificate Service, allerdings muss man anstelle der Maschinenzertifikate den VPN-Client selbst verteilen. Mit dieser Lösung ist man in keiner Weise an verschiedene Limitierungen der Windows-Lösung gebunden, sondern kann je nach Hersteller Dinge wie AES-Verschlüsselung, ECC-Public-Key-Verfahren, Datenkompression, PFS, IPSec-Roaming und eine automatische Client-Konfiguration einsetzen.

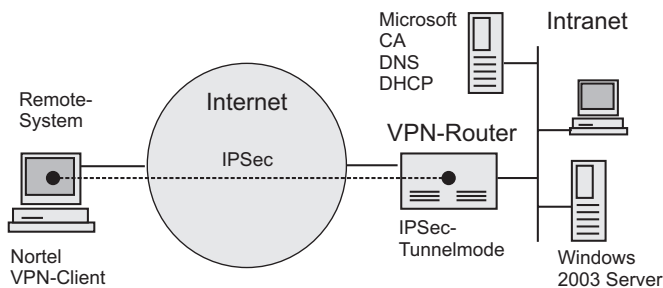


Abbildung 11.12: VPN-Router mit eigenen, speziell abgestimmten Clients bieten in der Regel sehr viel mehr Funktionen als Betriebssystem-Clients.

Optional bieten auch viele Hersteller die Möglichkeit der Authentifizierung per digitaler Signatur in IKE an, wobei hier wieder eine PKI erforderlich ist. Das kann, muss aber nicht, eine MS Certification Authority sein. Alternativ kann man auch Open-Source-

Produkte wie OpenSSL/OpenCA, professionelle PKIs von Spezialisten wie RSA oder Entrust oder öffentliche PKIs wie Teletrust oder Verisign einsetzen.

In diesem Beispiel wird ein IPSec-Client mit einer so genannten Push Configuration eingesetzt, der sich dadurch auszeichnet, dass er bei jedem Verbindungsaufbau seine Einstellungen und Betriebsparameter durch IKE-Config-Nachrichten vom VPN-Router erhält. Das ermöglicht eine zentrale Client-Konfiguration und erfordert keine Konfiguration durch den Benutzer. Dieses Verfahren wurde Mitte der 90er-Jahre durch kleine VPN-Pioniere wie New Oak Technologies (von Nortel übernommen) oder Altiga Networks (von Cisco übernommen) eingeführt und hat heute eine weite Verbreitung gefunden. Der einzige Wermutstropfen dabei ist die Tatsache, dass dieses Verfahren bisher keinen Einzug in einen Standard gefunden hat und damit jeder Hersteller sein eigenes Format benutzt. Abhilfe ist erst mit IKE Version 2 in Sicht.

In unserem Beispiel wird die Security Policy zentral im VPN-Router konfiguriert, hier werden für IKE AES-256-Bit mit ECC-283-Bit und für IPSec-ESP AES-256-Bit und SHA-1 festgelegt. Auch PFS (Perfect Forwarding Secrecy) wird aktiviert. ISAKMP erlaubt das Versenden von Notify-Nachrichten mit dem Typ INITIAL-CONTACT, die in der Regel dann benutzt werden, wenn nach einem unerwarteten Verbindungsabbruch der Gegenstelle signalisiert werden soll, dass eine eventuell existierende alte SA gelöscht und eine neue aufgebaut werden soll. Dies ist ein nützliches Feature bei Medien mit häufigen Verbindungsabbrüchen.

Ebenso wie bei den beiden anderen Lösungen ist hier vieles über Active Directory und damit verbundene Dienste wie DNS und DHCP steuerbar: Benutzerauthentifizierung, Zuordnung zu einer Gruppe im VPN-Router, benutzerspezifische Filter, Remote-Access-Steuerung, IP-Adress-Zuweisung usw. Der Client kann seine Adresse, WINS- und DNS-Informationen über den Microsoft-DHCP-Service beziehen und registriert seinen Namen und seine dynamische IP-Adresse automatisch beim Microsoft DNS. Auf dem VPN-Router braucht im laufenden Betrieb nichts mehr konfiguriert zu werden, das komplette Management erfolgt über Active Directory.

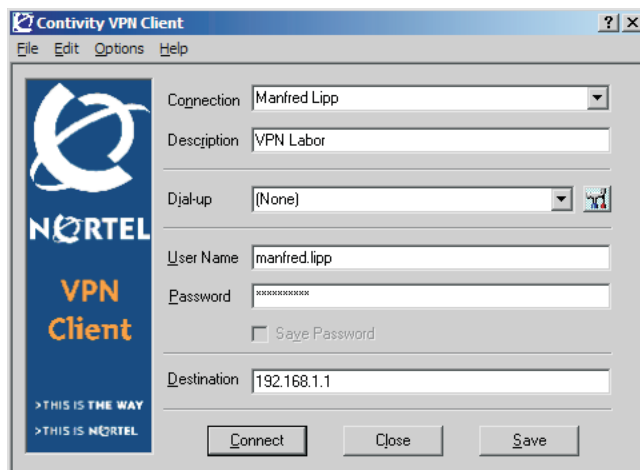


Abbildung 11.13: Der Nortel VPN-Client bietet eine einfach zu handhabende Benutzerschnittstelle.

Im RAS können sich maximal 30 Benutzer gleichzeitig anmelden, und haben dabei über einen ISDN-B-Kanal eine garantierte Bandbreite von 64 KBit/s zur Verfügung. Das ist nicht berauschend viel, aber es ist garantiert. Beim VPN-Router, zumindest bei einem guten, kann man auch die Anzahl gleichzeitiger Benutzer auf verschiedene Arten limitieren. Aber sie haben ohne spezielle zusätzliche Maßnahmen keine garantierte Bandbreite, insbesondere auf ihrem Weg durch das Internet kann einem IP-Paket alles zustoßen, vom blitzschnellen Transport bis zum Verwerfen. Sie können die maximale Bandbreite bekommen oder fast gar keine, je nachdem, was die anderen Benutzer so treiben.

Allerdings bieten gute VPN-Router die Möglichkeit, Bandbreitenmanagement bis auf Benutzerebene zu konfigurieren. Hierbei können den virtuellen Verbindungen, z.B. IPSec-Tunneln, bestimmte, bei richtigem Design auch garantierbare Mindestbandbreiten und Überbuchungsraten zugewiesen werden.

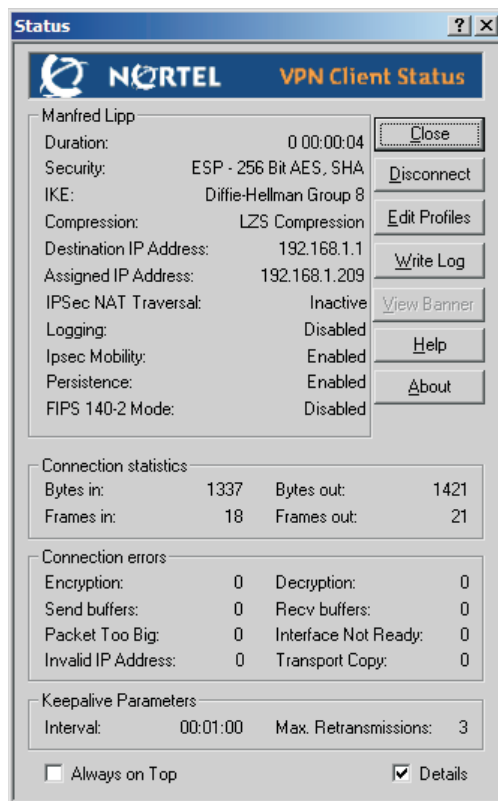


Abbildung 11.14: Sicherheit vom Feinsten: 256 Bit AES und 283 Bit ECC erfüllen auch höchste Sicherheitsansprüche.

Remote Access in großen Netzen

In diesem Beispiel plant ein größeres Unternehmen den Einsatz einer flexiblen Remote-Access-Lösung, die im zweiten Schritt auch externen Benutzern (Partnern, Lieferanten usw.) zur Verfügung gestellt werden soll. Primär sollen Vertrieb und technischer Außendienst die Lösung benutzen. Die IT des Unternehmens ist kompetent genug, um die Software-Verteilung im Griff zu haben, also ist die Installation von Clients auf den Endgeräten weder ein technisches Problem noch ein signifikanter Kostenaufwand.

Es sollen insgesamt 800 mobile Mitarbeiter versorgt werden, die sowohl von unterwegs, von Kundennetzen oder von zu Hause auf das Unternehmensnetz zugreifen. Die Applikationen sind primär E-Mail, webbasierende Anwendungen, Zugriff auf Datenbanken sowie einige spezielle Anwendungen für die Servicetechniker. Da die Firewalls in einigen Kundennetzen ausgehende IPSec-Verbindungen blockieren, sollen alternativ auch SSL-VPN-Zugriff für eigene Mitarbeiter eingesetzt werden können.

Es soll in weiteren Schritten auch externen Personen und Organisationen Zugang zu bestimmten Ressourcen des Unternehmensnetzwerks gewährt werden. Für diese Extranet-Lösung soll ebenfalls SSL eingesetzt werden.

Da die Mobilität der Mitarbeiter essenziell ist, muss eine Reihe von Access-Technologien unterstützt werden: ADSL, ISDN, analoge Modems, GPRS, UMTS, WLAN und Ethernet. Da die Mitarbeiter zu Hause ADSL-Router zur Verfügung gestellt bekommen, muss die VPN-Lösung NAT/PAT-kompatibel sein.

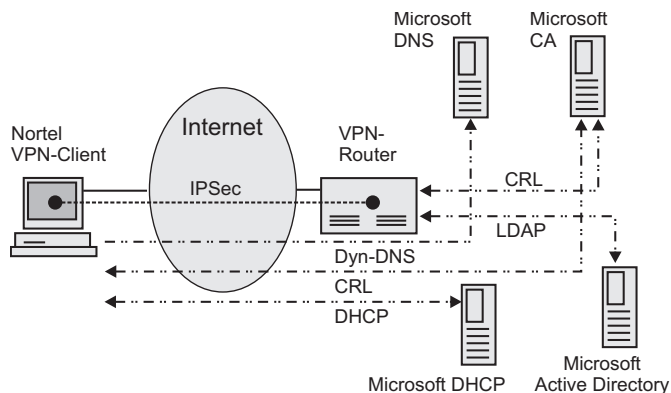


Abbildung 11.15: Die VPN-Lösung hat viele Schnittstellen zu den verschiedensten Diensten.

Das VPN ist kritisch für das Geschäft des Unternehmens, also muss es hochverfügbar sein. Es sollen daher zwei redundante VPN-Systeme in verschiedenen Gebäuden betrieben werden.

Aus nahe liegenden Sicherheitsgründen ist es den Mitarbeitern strengstens verboten, von fremden Rechnern per SSL auf das Unternehmensnetz zuzugreifen. Die Durchsetzung dieser Richtlinie soll durch das VPN erfolgen. Zusätzlich wäre es wünschenswert, wenn auf den Rechnern ein Security Policy Enforcement durch die VPN-Lösung erfolgt.

Weiterhin ist starke Verschlüsselung (ab 128 Bit symmetrisch und ab 1024 Bit asymmetrisch) obligatorisch. Es wird außerdem gerade eine Enterprise-PKI eingeführt, bei der sich die Benutzer mit USB-SmartCards authentifizieren. Die VPN-Lösung muss dies unterstützen.

Im Unternehmensnetz werden Windows 2003 Server, Active Directory Service (ADS), DNS und DHCP eingesetzt, als Client-Betriebssystem wird Windows XP verwendet. Die oben erwähnte PKI basiert auf dem Microsoft Certificate Service. Ziel ist die Verwendung der ADS-User und -Gruppen auch für das VPN. Nur die notwendigsten Konfigurationen sollen auf dem VPN-System erfolgen.

Planung

Die Anzahl der Benutzer gibt bereits einige Kriterien vor, so muss der VPN-Router mindestens 800 gleichzeitige Verbindungen unterstützen und eine ausreichende Bandbreite zur Verfügung stellen. Also muss man überschlägig berechnen oder testen, was an durchschnittlichem Verkehr anfällt. Dies kann man entweder während einer Evaluierungsphase ermitteln oder berechnen. Machen wir einmal Letzteres und schauen uns die Applikationen an: E-Mail ist absolut unkritisch und wird praktisch zwischen den Bursts der restlichen Applikationen mit durchgeschleust. Denn diese sind alle typische Datenapplikationen des Typs Client/Server. Es werden von der Client-Seite kleine Anfragepakete übertragen, und es kommt eine bestimmte Menge von Daten zurück. Isochrone Datenströme, die über längere Zeiten eine gewisse Grundlast erzeugen, kommen in diesem Szenario nicht vor.

Somit kann man berechnen, wie viele Daten von allen 800 mobilen Mitarbeitern pro Stunde bei heftigster Arbeit übertragen werden. Dieser Wert kann auf eine Sekunde umgerechnet werden, und man hat die durchschnittliche Bandbreite in Bit/s. Nehmen wir für unser Beispiel an, es wären dabei 50 MBit/s herausgekommen. Man müsste also von einem Service Provider entweder eine 50-MBit/s-Verbindung oder einen anderen Standardwert aus dessen Portfolio anmieten, was je nach ISP 34, 45 oder 155 MBit/s sein kann.

Allerdings kann man hier noch etwas tricksen, denn die genannten Applikationen schreien geradezu nach Datenkomprimierung. Bei Einsatz von IPSec-Datenkompression (SSL wird ohnehin komprimiert) kann man unter Umständen die effektive Nutzbandbreite verdoppeln. Also würde eine Bandbreite von 34 MBit/s ausreichen. Wenn zu erwarten ist, dass die Zahl der Benutzer signifikant steigt, sollte man ausreichende Leistungsreserven einbauen oder Systeme auswählen, die durch Hardware-Module erweitert werden können.

Damit kann man schon anfangen, sich die Angebote möglicher ISPs einzuholen. Neben der garantierten Bandbreite sollten im Service Level Agreement (SLA) Dinge wie Verfügbarkeiten, Delay, Fehlerrate und vor allem das ausdrückliche Verbot von Paketfilterung durch den Service Provider enthalten sein! Im Hinblick auf zukünftige konvergente Anwendungen kann man sich bei dieser Gelegenheit auch gleich erkundigen, was der ISP in Zukunft denn so zum Thema QoS plant, und dies eventuell in seine Entscheidung einfließen lassen.

Beim Remote Access ist aber unter bestimmten Umständen ein ganz anderer Wert entscheidend: die Anzahl gleichzeitiger IKE-Verbindungsaufbauten pro Sekunde. Normale

VPN-Systeme ohne spezielle Hardware-Beschleuniger für Diffie-Hellman- und RSA-Algorithmen schaffen je nach momentaner Auslastung des Systems drei bis fünf IKE-Verbindungen mit Pre-Shared-Key-Authentifizierung, mit digitalen Signaturen sind es entsprechend weniger. Also könnten sich pro Minute im Durchschnitt etwa 300 Benutzer anmelden. Bei unseren 800 Benutzern könnte es im Extremfall, wenn sich alle zum gleichen Sekundenbruchteil anmelden würden, also bis zu 2 Minuten und 40 Sekunden dauern, bis alle Verbindungen aufgebaut sind. Allerdings nur wenn über die bereits aufgebauten Verbindungen keine Daten übertragen werden. Deshalb sollte man in IKE, wenn die VPN-Hard- und Software dies erlaubt, ECC-Verfahren einsetzen, da diese um ein Vielfaches schneller als die Standard-Schlüsseltauschalgorithmen.

Da Vollredundanz gefordert ist, müssen zwei getrennte VPN-Router eingesetzt werden, von denen jeder alle Benutzer aufnehmen kann. Idealerweise haben die Systeme eine integrierte Backup- und optional eine integrierte Lastverteilungsfunktion, so dass man auf externes Equipment verzichten kann.

Der VPN-Router

Die Leistungskriterien für den VPN-Router sind somit 34 MBit/s Durchsatz auch bei kleinen Paketen. Die beschriebenen Anwendungen erzeugen im Mittel relativ große Pakete, falls in Zukunft noch VoIP hinzukommen sollte, sind sehr viele Pakete im Bereich 160 bis 200 Byte darunter. Also sollte bei 200-Byte-Paketen der Durchsatz mindestens 34 MBit/s sein.

An der Stelle vielleicht ein Tipp: Im Kapitel 4 wurde erwähnt, dass der AES als Software-Implementierung etwa doppelt bis dreimal so schnell ist wie Triple-DES. Falls man also AES einsetzt, kann man, je nach VPN-Router, eventuell ohne zusätzliche Hardware-Beschleuniger auskommen. Auch der Aufbau von IKE-Verbindungen kann deutlich beschleunigt werden, wenn man für das Diffie-Hellman-Verfahren anstelle von Modularer Exponentiation (ModP) besser ECC-Verfahren einsetzt, hier ist der Beschleunigungsfaktor je nach Implementierung etwa fünf bis zehn.

Da im zweiten Schritt auch SSL als VPN-Technologie eingesetzt werden soll, muss der VPN-Router entsprechend erweiterbar sein, oder man muss ein weiteres SSL-VPN-System beschaffen. Einen normalen IPSec-, L2TP- oder PPTP-Router per Software zum SSL-VPN-Gateway umzufunktionieren, wird scheitern, sobald mehr als eine Hand voll Benutzer SSL benutzen. Die Gründe sind vor allem folgende:

Der SSL-Session-Aufbau benötigt ein oder drei RSA-Operationen, für die es in fast keinem handelsüblichen VPN-System (natürlich außer SSL-VPN-Gateways) entsprechende Hardware-Beschleuniger gibt.

SSL-VPN-Gateways führen aufwändige Operationen (Protocol Translation, Application Translation, Proxy) auf Schichten bis hoch zur Applikationsebene durch, die enorme Rechenzeiten verschlingen.

Wenn diese Funktionen einem System, das für so etwas nicht ausgelegt ist, abverlangt werden, kann es zu ziemlichen Performance-Einbrüchen kommen.

Das Problem ist den Herstellern bereits länger bekannt und wird den VPN-Spezialisten, die sowohl IPSec-VPN als auch SSL-VPN anbieten, auf verschiedene Weise gelöst. Juniper Networks war einkaufen und setzt zum Beispiel auf zwei voneinander unabhängige

Lösungen mit einem IPSec-Router (früher: Netscreen) und einem SSL-VPN-Gateway (früher: Neoteris).

Cisco Systems beschreitet den entgegengesetzten Weg, war ausnahmsweise nicht einkaufen und erweitert seine VPN 3000er-Serie einfach mit SSL-Software-Optionen, schenkt seinen Kunden aber reinen Wein ein: Eine SSL-Verbindung nimmt Ressourcen von etwa 20 IPSec-Verbindungen in Anspruch, was bei der Planung berücksichtigt werden muss.

Nortel bietet für einige seiner VPN-Router-Modelle eine spezielle SSL-Beschleunigerkarte an, die neben der Hardware-Beschleunigung für RSA, D/H, Triple-DES und AES auch eine spezielle CPU mit ausreichend Speicher für Application Translation, Protocol Translation usw. eingebaut hat. Durch die Integration in den VPN-Router können bereits vorhandene IPSec-Gruppen und -Benutzer auch für SSL weiter benutzt werden. Diese Lösung soll hier eingesetzt werden. Es wird ein VPN-Router 2700 eingesetzt, der bis zu 2000 Verbindungen terminieren kann. Das SSL-VPN-Modul erlaubt die Terminierung von bis zu 1000 weiteren SSL-Benutzern, ohne dass dafür Systemressourcen des VPN-Routers benutzt werden und dadurch Performance für die Remote-Access-Benutzer verloren geht.

Aus Kostengründen empfiehlt es sich, mit einer kleinen SSL-VPN-Benutzerlizenz, z.B. für 50 Benutzer, anzufangen und diese bei Bedarf zu erweitern. Durch einen speziellen Hardware-Beschleuniger-Chipsatz auf dem Modul kommt man auf mehrere hundert MBit/s Durchsatz und etwa 2000 RSA-Operationen pro Sekunde. Letzterer Wert entspricht 600 unidirektionalen oder 300 bidirektionalen Signatur-Authentifizierungen. Somit ist für den SSL-VPN-Teil ausreichend Leistung vorhanden.

Für Redundanz stellt der VPN-Router eine Failover-Funktion zur Verfügung, die in Verbindung mit den Clients erkennt, wenn eine Verbindung abbricht, und daraufhin der Reihe nach bis zu drei Backup-Systeme zu erreichen versucht. Die Systeme können in verschiedenen Standorten stehen und über verschiedene ISP angebunden werden. Ohne die Notwendigkeit BGP einsetzen zu müssen. Für unser Beispiel werden zwei VPN-Router eingesetzt. Zwischen diesen beiden Routern kann man auch ohne weitere externe Maßnahmen eine Lastverteilung einrichten, welche die Verbindungen so verteilt, dass die beiden VPN-Router gleichmäßig ausgelastet sind.

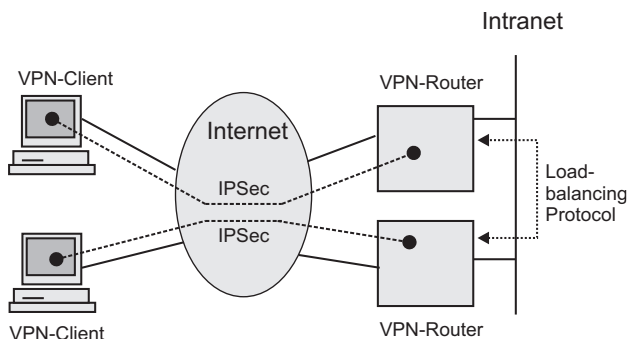


Abbildung 11.16: Lastverteilung und Redundanz in einer Remote-Access-VPN-Lösung

Der VPN-Client

Ein VPN-Client muss einfach zu bedienen und je nach Vorkonfiguration benutzertransparent installierbar sein. Der verwendete Nortel VPN-Client basiert auf der so genannten Push Configuration. Derartige Clients werden während jedes Verbindungsaufbaus vom Gateway mit den notwendigen Parametern konfiguriert. Auf dem Client braucht, kann und darf der Benutzer aus Sicherheitsgründen keine Einstellungen vornehmen. Er muss nur seine Anmeldeinformationen eingeben, der Rest erfolgt automatisch. Man muss daher Konfigurationsänderungen nicht auf allen Clients durchführen, sondern nur ein einziges Mal auf dem VPN-Router.

Der Client arbeitet im IPSec-Tunnel-Mode. Als Initiator muss er alle möglichen IKE- und IPSec-Proposals vorschlagen, aus denen der VPN-Router (der IKE-Responder) sich die passende Policy auswählen kann. Die Informationen über Backup-Systeme und Lastverteilung werden dem Client ebenfalls per Push-Konfiguration zugewiesen.

Im Unternehmen wird eine PKI eingeführt, die Benutzer erhalten SmartCards mit ihrem privaten Schlüssel und ihrem Benutzerzertifikat. Die VPN-Lösung setzt auf die Integration der SmartCards in das Windows XP-Betriebssystem über die Microsoft Crypto-API (MS-CAPI). Damit kann man die SmartCards zum Authentifizieren, Verschlüsseln und Signieren benutzen. Der VPN-Client kann ebenfalls die Systemfunktionen der MS-CAPI benutzen und damit auch die SmartCard der Benutzer. Somit kann die Authentifizierung im VPN auch extrem sicher per digitaler Signatur auf der SmartCard des Benutzers erfolgen. Ein weiterer Vorteil von digitalen Signaturen ist der, dass der VPN-Client trotz der bei Remote Access typischen Zuweisung von dynamischen IP-Adressen im IKE Main Mode arbeiten kann, was ein deutlich höheres Maß an Sicherheit als der IKE Aggressive Mode bietet (vgl. Kap 6).

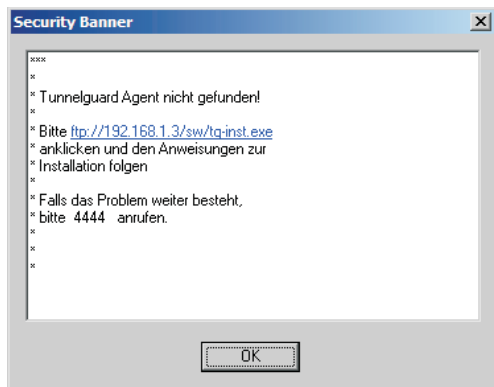


Abbildung 11.17: Fehlgeschlagen: Auf diesem Rechner ist die geforderte Policy nicht erfüllt.

Client Policy Enforcement

Als Option bietet der Nortel VPN-Client eine Funktion namens Tunnel Guard. Diese Software hat die Aufgabe, beim Aufbau der VPN-Verbindung, egal ob SSL oder IPSec, eine festgelegte Security Policy auf dem Clientrechner zu überprüfen. Es können beliebige Prozesse, DLLs, Dateien und Registry-Einträge auf Existenz, Version und Integrität geprüft werden. So kann zum Beispiel geprüft werden, ob ein bestimmter Virens Scanner

und eine bestimmte Firewall aktiv sind, ob bestimmte Prozesse, zum Beispiel Key-Logger, nicht aktiv und bestimmte Dateien an ihrem Speicherort und unverändert sind.

Ist die Policy nicht erfüllt, kann die Verbindung mit einer Meldung sofort wieder angebrochen werden, oder es wird eine eingeschränkte (Filter, Quarantäne-VLAN) Verbindung, zum Beispiel zu einem Update-Server, zugelassen. In der Meldung können auch anklickbare Links enthalten sein, über die der Benutzer online Hilfestellung erhalten kann.

Die Funktion kann auch periodisch wiederholt werden. Es kann auch geprüft werden, ob und, wenn ja, welcher Tunnel-Guard-Agent auf dem Rechner ist, wie zum Beispiel in Abbildung 11.17 zu sehen ist. Hier wurde kein Agent gefunden, was eine Verletzung der Sicherheitsrichtlinie auf dem Endgerät darstellt. Der Benutzer kann sich den Agenten entweder herunterladen oder die Hotline anrufen.

11.8.3 Anbindung von kleinen Büros

In diesem Beispiel soll eine größere Zahl von kleinen Büros oder Heimbüros, auf Neu-deutsch als SOHO (Small Office/Home Office) bezeichnet, an eine Zentrale per VPN angebunden werden. Ein typischer Fall für eine solche Anwendung sind zum Beispiel Reisbüros oder Versicherungsbüros. Es sollen hier 400 Büros angeschlossen werden. Die Applikationen, die benutzt werden sollen, sind vor allem E-Mail, Datenbanktransaktionen und Webanwendungen.

Der VPN-Router in der Außenstelle muss mindestens vier 10/100-MBit/s-LAN- und eine WAN-Schnittstelle zur Verfügung stellen. Für mehr Benutzer ist ein externer Layer 2 Switch einzusetzen. Ein serieller Port ist ebenfalls gefordert, um in einem weiteren optionalen Schritt ein Out-of-Band-Management einsetzen zu können.

In den Außenstellen soll der VPN-Router DHCP-Server- und DNS-Proxy-Funktionen wahrnehmen.

Aus Kostengründen sollen hauptsächlich preisgünstige ADSL-Verbindungen mit einer Downstream-Geschwindigkeit von 2 MBit/s benutzt werden, das ist bei diesem Verkehrsmuster auch kein Problem. Das lästige Trennen der DSL-Verbindungen durch den Provider erfordert, falls auch Verbindungen von der Zentrale aus initiiert werden sollen, dass die VPN-Router in den Außenstellen ihre Verbindungen automatisch gleich wieder aufbauen. Außerdem muss ein Mechanismus in der Art existieren, dass der zentrale VPN-Router den Neuaufbau einer SA (Security Association) von einer Gegenstelle auch dann akzeptiert, wenn zu dieser noch eine alte, inaktive SA besteht. Denn das zentrale System kann in der Regel nicht so schnell erkennen, wenn im Zweigbüro die DSL-Strecke kurz deaktiviert wird.

Ein weiteres Problem, vor allem bei Heimbüros, besteht darin, dass zwar die Verbindung zu dem zentralen VPN-Router sicher ist, aber es nicht ohne weiteres überprüft werden kann, welcher Rechner im Heimbüro an den VPN-Router angeschlossen ist und wer an dem Rechner arbeitet. Hier sind geeignete Maßnahmen erforderlich.

Lösungsmöglichkeiten

Hier gibt es prinzipiell zwei Möglichkeiten:

Man setzt einen VPN-Router in den Außenstellen ein und implementiert zusätzlich eine Authentifizierung per EAP over LAN (EAPoL nach IEEE 802.1X) oder richtet eine Benutzerauthentifizierung an einer zentralen Firewall ein.

Man setzt einen normalen Access-Router ein und installiert auf den Rechnern VPN-Clients. Wenn lokal zwischen den Rechnern oder mit einem Netzwerkdrucker kommuniziert werden soll, muss auf den Clients Split-Tunneling zu dem lokalen Netzwerk freigegeben werden.

Bei der letzten Alternative handelt es sich um ein normales Remote Access VPN, was in den vorhergehenden Abschnitten bereits ausführlich diskutiert wurde. Deshalb beschäftigen wir uns hier mit der ersten Alternative, einem VPN-Router in den Büros.

Der VPN-Router in den Außenstellen

Aus Gründen der Flexibilität ist dabei einem System der Vorzug zu geben, das kein eingebautes Modem besitzt. Dadurch kann man zukünftig einfacher zu anderen Access-Technologien wechseln, insbesondere zu SDSL. Die Leistungsfähigkeit des VPN-Routers sollte daher bei starker Verschlüsselung (Triple-DES oder AES) und 200-Byte-Paketen mindestens 2 MBit/s in beiden Übertragungsrichtungen betragen.

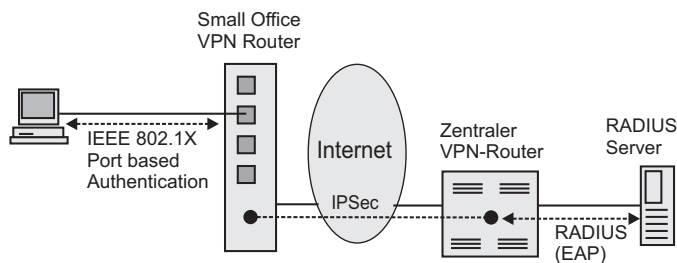


Abbildung 11.18: Sinnvoll: portbasierende Authentifizierung (IEEE 802.1X) ohne RADIUS-Server in der Außenstelle

Ein System, das in diese Kategorie fällt, ist zum Beispiel der Nortel VPN-Router 221. Er besitzt auf der WAN-Seite eine 10/100-Ethernet-Schnittstelle, an die ein DSL-Modem angeschlossen werden kann. Auf der lokalen Seite ist ein 4-Port-Switch eingebaut, der Port-Authentifizierung nach IEEE 802.1X unterstützt und zur Authentifizierung der angeschlossenen Rechner eingesetzt werden kann. Die Benutzer können lokal im VPN-Router oder in einem zentralen RADIUS-Service gepflegt werden.

Trotz des Namens „VPN-Router“ muss nicht geroutet werden, das System ist das Default Gateway für alle lokalen Systeme und schickt jedweden Datenverkehr zur Zentrale.

Ein wichtiger Punkt bei 400 Außenstellen ist auch der Roll-Out. Idealerweise lässt man die VPN-Router gleich in den Außenstellen anliefern, von einem Mitarbeiter anschließen und von der Zentrale aus konfigurieren. Hierfür müssen entsprechende Tools existieren, und vor allem muss die Default-Konfiguration der VPN-Router so ausgelegt sein, dass diese ohne irgendwelche Eingriffe sofort eine Internet-Verbindung aufbauen können, nachdem sie angeschlossen und eingeschaltet wurden.

Kalkulieren wir einmal kurz den Durchsatz: 400 Außenstellen à 2 MBit (Downstream) macht insgesamt 800 MBit/s. Allerdings nur theoretisch, denn keine der Applikationen erzeugt im Mittel hohe Datenraten, im Gegenteil, es wird stoßweiser Datenverkehr mit sehr vielen Pausen erzeugt. Da kommt man in der Zentrale mit sehr viel weniger Durchsatz aus, 100 MBit/s sollten mehr als ausreichend sein. Unter Umständen kann sogar eine 34-MBit/s-Verbindung ausreichen.

Da man in Heimbüro letztendlich nicht die gleiche Gebäude-Zugangskontrolle wie in normalen Büros hat, ist zu überlegen, ob, falls die Unternehmens-Sicherheitsrichtlinien so etwas fordern, zusätzlich eine Firewall eingesetzt werden sollte. Dies kann entweder intern im VPN-Router oder extern geschehen. Die Firewall soll nur den für die Arbeit in den Außenstellen notwendigen Verkehr zulassen und alles andere blockieren.

Was in der Zentrale kritisch ist, ist deren Verfügbarkeit. Denn wenn sie nicht mehr erreicht werden kann, dann kann in keinem einzigen Büro mehr richtig gearbeitet werden. Also sollte hier redundant geplant und zwei Router eingesetzt werden, die am besten noch getrennt an das Internet angebunden werden. Detailliertere Ausführungen zum Thema Redundanz finden Sie im folgenden Abschnitt.

In letzter Zeit werden verstärkt Standortverbindungen, die bisher mit herkömmlicher Router-Technik und Festverbindungen, Frame Relay oder ATM realisiert wurden, durch VPN abgelöst. Ausschlaggebend sind hier fast ausschließlich die einzusparenden Kosten. Allerdings werden an solche VPNs meist deutlich höhere Anforderungen als an die meisten Remote Access VPNs gestellt. Denn hier werden teilweise relativ große und oft auch viele Standorte, nicht selten weltweit verteilt, miteinander verbunden. Neben entsprechendem Datendurchsatz spielen hier vor allem auch Verfügbarkeit, Quality of Service und sicheres Routing eine Hauptrolle.

In dem vorliegenden Beispiel sollen insgesamt fünf größere regionale Niederlassungen und 1500 kleinere Verkaufsstellen mit einer großen Unternehmenszentrale verbunden werden. Die kleinen Standorte werden mit 2 MBit/s ADSL (192 KBit/s upstream) versorgt, größere Standorte mit 34 MBit/s oder 10 MBit/s (Ethernet oder E3/Nx2MBit/s). Die Zentrale des Unternehmens wird von einem Carrier mit zwei 155 MBit/s Glasfaserstrecken zu verschiedenen Zugangspunkten versorgt. Die alte Infrastruktur bestand aus 64-KBit/s-Standardfestverbindungen zu den Verkaufsstellen und 2-MBit-Festverbindungen zu den Niederlassungen.

Die Hauptanwendungen in den Verkaufsbüros sind E-Mail, unternehmensspezifische Datenbankapplikationen, Voice over IP und ein nächtlicher Software-Update-Service. In den größeren Außenstellen werden vor allem E-Mail, Datenbank- und Webapplikationen sowie VoIP eingesetzt.

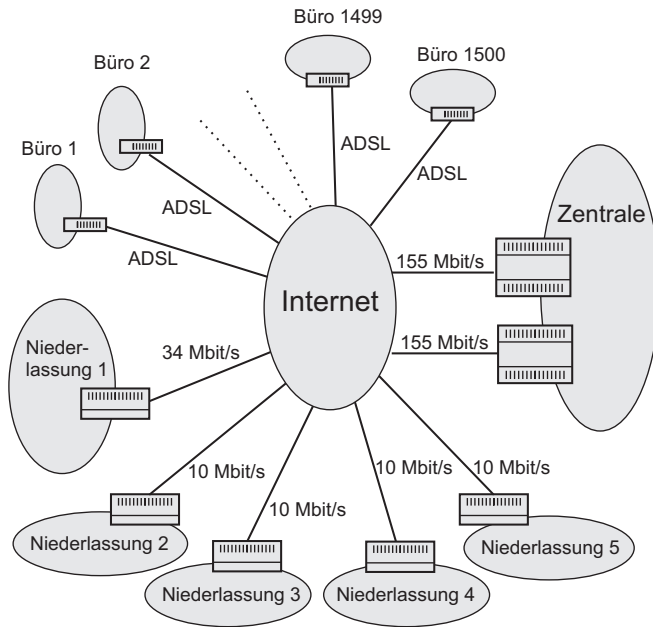


Abbildung 11.19: Das Beispielnetzwerk

In der alten Infrastruktur wurde in Frame Relay und ATM bereits QoS eingesetzt, dies soll hier auch geschehen. Die VoIP-Systeme sind so konfiguriert, dass über die VPN-Strecken sowohl mit G.711- als auch notfalls mit G.729-Codecs (Coder/Decoder, Verfahren zur digitalen Codierung analoger Sprachsignale) gearbeitet werden kann und man pro Standort eine maximale Anzahl von gleichzeitig aktiven Sprachverbindungen festlegen kann. Es wird relativ viel telefoniert, so dass bei der Planung konservativ gerechnet und von einer 50-prozentigen Auslastung der konfigurierten Sprachverbindungen ausgegangen werden muss.

Das Mengengerüst für die Niederlassungen und kleinen Büros sieht wie folgt aus:

Tabelle 11.2: Die verschiedenen Standorte in diesem Beispielszenario

Aufgrund der Bandbreiten, die Niederlassungen bringen es zusammen auf etwa 362 MBit/s (upstream) und die Zentrale bidirektional auf 310 MBit/s, sieht man, dass keine hohe Überbuchung vorliegt, also Engpässe von dieser Seite praktisch nicht auftreten. Die Downstream-Kapazität ist mit 3.146 MBit/s auf den ersten Blick sehr hoch. Aufgrund des Verkehrsprofils der Verkaufsstellen ist das aber nicht als Problem, sondern eher als Überkapazität zu sehen.

Bei der Auswahl der VPN-Router ist zu bedenken, dass aufgrund des hohen Sprachanteils sehr viele Pakete im Größenbereich von 200 Byte, im Extremfall, wenn G.729 eingesetzt werden muss, sogar von 80 Byte erzeugt werden. In der Zentrale müssen gemäß des Mengengerüsts in Tabelle 11.2 maximal 3.300 Sprachverbindungen gleichzeitig terminiert werden können, für die Planung soll, wie oben gefordert, von 50%, also 1.650 Verbindungen ausgegangen werden.

Bei über 1.500 Standorten, die überdies auch noch VoIP einsetzen, sind sowohl bei der Planung als auch der Geräteauswahl und Konfiguration einige wichtige Kriterien zu beachten.

Im Vorfeld muss geklärt werden, welche Verkehrsbeziehungen bestehen und wie die Profile aussehen. Der starke Telefonverkehr erfordert, dass dieser auch mit entsprechender Sprachqualität übertragen werden kann. Man benötigt hinreichend Bandbreite und entsprechende Qualitätsmerkmale (Verzögerung und Verzögerungsvarianz). Die VoIP-Infrastruktur sollte so ausgelegt werden, dass die Anzahl gleichzeitiger Verbindungen von und zu einem Standort so limitiert wird, dass auch noch ausreichend Datenverkehr möglich ist.

Die VPN-Systeme werden mit sehr vielen kleinen IP-Paketen im Bereich von 200 Byte konfrontiert, was bei der Kalkulation des notwendigen Systemdurchsatzes berücksichtigt werden muss. Außerdem besteht eine Wechselwirkung zwischen starker Belastung eines VPN-Routers und der Verzögerung und Verzögerungsvarianz. Mit anderen Worten, wenn die Systemlast steigt, nimmt ab einem bestimmten Schwellenwert die Verzögerung sehr stark zu, bis in den Bereich einiger zehn Millisekunden.

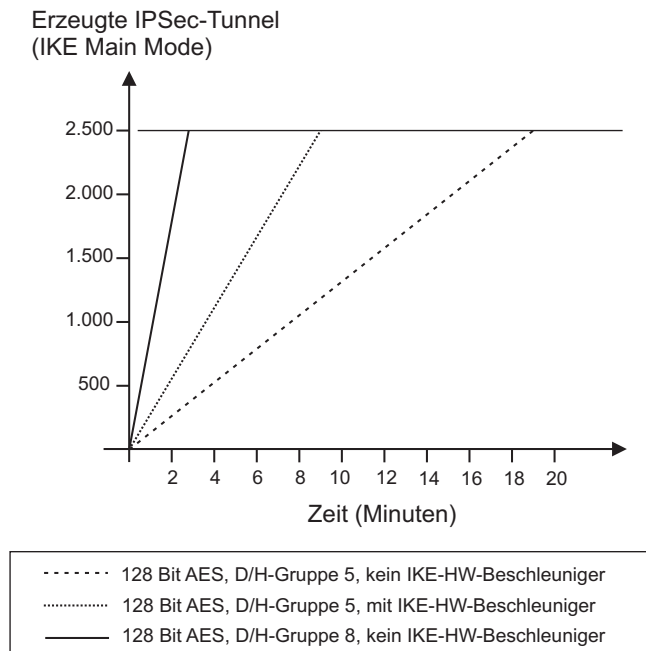


Abbildung 11.20: Je nach Key-Exchange-Algorithmus (D/H-Group) und Hardware können die IKE-Tunnelaufbauzeiten stark divergieren.

Etwas anderes ist ebenfalls zu berücksichtigen, nämlich die Zeit, in der alle Verbindungen aufgebaut werden, beispielsweise nach einem Systemausfall oder wenn ein Backup-System aktiv wird. Hier kann es bei der Anzahl der Verbindungen etliche Minuten dauern, bis alle IPSec-Tunnel aufgebaut sind. Man kann mehrere Strategien anwenden, unter Umständen auch miteinander kombiniert.

Man kann statt einem oder zwei sehr leistungsfähigen VPN-Routern mehrere Systeme mittlerer Leistungsfähigkeit einsetzen. Damit verteilt sich die hohe CPU-Last, die durch die Public-Key-Verfahren zur Schlüsselerzeugung erzeugt wird, auf mehrere Systeme. Außerdem fällt der Ausfall eines Systems dabei nicht so ins Gewicht.

Auch der Einsatz von speziellen Hardware-Beschleunigern, die Public-Key-Operationen in Hardware implementiert haben, kann Abhilfe schaffen und die Geschwindigkeit des Verbindungsaufbaus fast verdoppeln.

Aber auch reine Software-Lösungen können die Ausführung des IKE-Protokolls beschleunigen, so können Algorithmen, die auf der Elliptic Curve Cryptography (ECC) basieren, den Verbindungsaufbau in manchen Fällen um das Sechs- bis Zehnfache beschleunigen.

In Abbildung 11.20 sehen Sie die letzten beiden Varianten im Vergleich. Gemessen wurde dabei die Zeit, die vergeht, bis ein VPN-Router 2.500 IPSec-Tunnel, die ihm gleichzeitig angeboten wurden, aufgebaut hat. Man sieht, dass die ECC-Algorithmen (Diffie-Hellman-Algorithmus mit ECC) auch ohne Hardware-Beschleunigung sehr viel schneller sind als die Diffie-Hellman-Varianten, die auf der modularen Exponentiation über großen Primzahlen (ModP) beruhen.

Insbesondere die IP-Pakete, die Sprache transportieren, müssen bevorzugt weitergeleitet werden, um die Gesamtverzögerung niedrig und damit die Sprachqualität hoch zu halten. Hier sind entsprechende QoS-Maßnahmen, insbesondere DiffServ, gefragt, VoIP wird normalerweise mit EF (Expedited Forwarding) markiert. Ein QoS-Mapping auf das lokale Netzwerk, in der Regel Ethernet mit IEEE-802.1p-Priorisierung, ist ebenfalls wichtig und darf keinesfalls fehlen. Beim Einsatz von DiffServ sollte man nicht vergessen, den IPSec Anti Replay Service abzuschalten (vgl. Kap 9), da sonst der Paketverlust bei den BE- und AF-markierten Paketen unter normalen Bedingungen, also ohne Überlast, extrem ansteigen würde.

Falls im lokalen Netzwerk noch keine Markierung im DSCP (DiffServ Code Point) der IP-Pakete erfolgt ist, muss dies auf alle Fälle im VPN-Router erfolgen. Es kann auch unter Umständen notwendig werden, im DSCP des äußeren IP-Headers einen anderen Wert zu verwenden als im inneren, privaten IP-Header. Das kann in Zukunft notwendig werden, wenn ein fortschrittlicher Carrier oder Service Provider in seinem Netz QoS anbietet und man dafür entsprechende DSCP-Werte liefern muss.

Das Thema QoS für VoIP ist nicht isoliert zu betrachten, sondern muss mit der IPT-Infrastruktur (IPT, IP Telephony) zusammenspielen. So ist es wichtig zu wissen oder noch besser während der Designphase zu beeinflussen, mit welchen VoIP-Codecs gearbeitet wird und ob die Möglichkeit besteht, bei Gesprächen über das VPN, also über die WAN-Verbindungen, mit anderen Codecs als im LAN zu arbeiten. Auch eine Beeinflussung der Anzahl von Voice-Samples pro IP-Paket hat bestimmte Einflüsse.

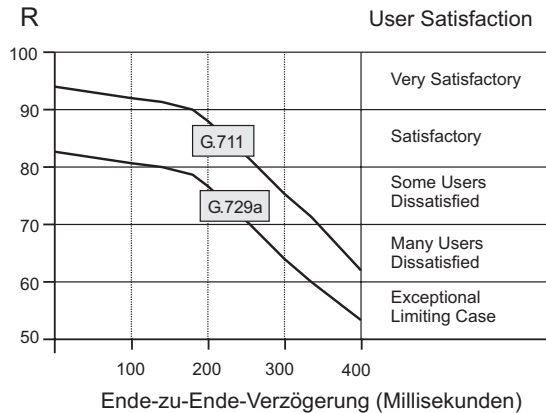


Abbildung 11.21: Die Qualität von VoIP hängt unter anderem vom Codec und der Verzögerung ab.

Allerdings ist die Sache ganz so einfach nicht. In Abbildung 11.21 sehen Sie den Zusammenhang zwischen der vom Benutzer subjektiv empfundenen Gesprächsqualität, dem so genannten R-Wert, und der unidirektionalen Verzögerung für Kodierungen nach der G.711-Norm (64 KBit/s) und der dieser gegenüber achtfach komprimierten Übertragung nach G.729. Man kann sich alleine aufgrund des hohen und damit stark verlustbehafteten Kompressionsfaktors schon vorstellen, dass die Sprachqualität bei G.729 von vornherein schon niedriger ist als die von G.711, das praktisch ISDN-Qualität besitzt. In der Abbildung 11.21 kann man sehen, dass eine verzögerungslose Übertragung mit G.729-Kodierung bereits so schlecht ist wie G.711 mit einer Verzögerung von 250 Millisekunden. Also reagiert G.729 viel empfindlicher auf Verzögerungen und vor allem auch auf Jitter.

Das alleinige Verwenden von G.729 ist also beileibe nicht das Allheilmittel, denn es erfordert eine hohe Übertragungsqualität, die im Internet leider nicht gegeben ist. Aber es ist preiswert, denn die Bandbreite ist wesentlich niedriger als die von G.711. Die Bandbreite von VoIP-Datenströmen hängt nämlich von den ausgewählten Codecs und von der Anzahl der digitalisierten Sprachmitschnitte (Samples) im IP-Paket ab. Die Zusammenhänge finden Sie in Tabelle 11.3. Dabei ist zu beachten, dass aufgrund des Overheads von RTP (Real Time Protocol), UDP und IP die Bandbreite von G.729 keineswegs wie viel leicht erwartet achtmal kleiner ist. Sie ist im Schnitt nur etwa viermal niedriger.

			Größe	Bandbreite auf IP-Layer

Tabelle 11.3: Die Abhängigkeit von Paketgröße und Bandbreite von der Anzahl von Samples in einem IP-Paket

Aufgrund dieser Zusammenhänge kann man nun an die Feinplanung gehen und schauen, ob die ursprünglich geplanten Bandbreiten ausreichend sind und wie man seine QoS-Parameter konfigurieren muss.

In der Zentrale ist, wie einleitend beschrieben, bei der Planung von insgesamt 1.650 Verbindungen auszugehen, das entspricht einer Bandbreite von 132 MBit/s bei Verwendung der hochwertigeren G.711-Codexs und vier Sprachsamples pro IP-Paket. In den kleinen Außenstellen ist durchschnittlich eine Sprachverbindung gefordert, die mit 80 KBit/s (G.711) pro Verbindung noch 112 KBit/s (upstream) für Datenverkehr erlauben würde. Beim erlaubten Maximum von zwei Verbindungen wären für Daten noch 32 KBit/s (upstream) übrig, was bei den vorliegenden Verkehrsprofilen aber ausreichend ist. Downstream hat man mit 2 MBit/s überhaupt keine Probleme.

In den großen Niederlassungen liegen die Verhältnisse ähnlich, hier benötigt man für den Sprachverkehr entweder 2 MBit/s (maximal 4 MBit/s) oder 4 MBit/s (maximal 8 MBit/s) bei 10 bzw. 34 MBit/s symmetrischer Bandbreite.

Somit kann man also problemlos mit G.711-Codexs arbeiten und hat dadurch eine entsprechend hohe Grundqualität der Sprache. Außerdem werden die größtmöglichen VoIP-Paket erzeugt, was dem IPSec-Durchsatz zugute kommt. Bei den 80-Byte-Paketen mit G.729 würden IPSec-Systeme aufgrund des größeren Paket-Overheads einen deutlich niedrigeren Durchsatz produzieren.

Nachdem nun die Datenraten so weit klar sind, kann man seine QoS-Parameter festlegen. VoIP-Verkehr ist Echtzeitkommunikation und sollte auch so behandelt werden. Das heißt, die Pakete müssen spätestens im VPN-Router eine entsprechende DiffServ-Markierung (EF) bekommen und dort auch in die entsprechenden Prioritäts-Warteschlangen kommen. Für den restlichen Datenverkehr kann man bei Bedarf entsprechende Prioritäten vergeben.

Falls die VPN-Router dies zulassen, sollten die Interfaces so konfiguriert werden, dass Bandbreitenengpässe im Transportweg der Daten bereits auf dem VPN-Router abgebildet werden, um diese in die QoS-Steuerung mit einfließen zu lassen. Falls in den kleinen Büros ein ADSL-Modem an die Ethernet-Schnittstelle des VPN-Routers angeschlossen wird, dann erkennt dieser eine Schnittstellengeschwindigkeit von 10 MBit/s und wird seine Warteschlangen entsprechend groß auslegen. Aber in Wirklichkeit kann das Modem nur 192 KBit/s übertragen, und es würden dort, unabhängig von ihrer Priorität,

massiv Pakete verworfen. Wenn man jedoch die Interface-Geschwindigkeit auf einen anderen als den erkannten Wert, hier 192 KBit/s, konfigurieren kann, dann würden die Warteschlangen nicht so groß, die Verzögerungen klein und das kurz vor oder während einer Überlast auftretende Dropping auf unkritische Pakete beschränkt.

Ausfallsicherheit

Auch bei der Ausfallsicherheit gibt es für unser Szenario verschiedene Möglichkeiten zur Realisierung. Im Wesentlichen unterscheidet man zwischen speziellen Hochverfügbarkeitsmechanismen wie VRRP und den Möglichkeiten, die man durch den Einsatz dynamischer Routing-Protokolle hat. Außerdem muss man festlegen, ob die Redundanz-Funktionalität auf den zentralen Systemen oder in Außenstellen arbeitet – oder beides.

VRRP, das Virtual Router Redundancy Protocol, ist ein Standard, der eine Hochverfügbarkeitslösung für Router beschreibt. Dieses Protokoll ist mittlerweile in praktisch jedem Router, jedem Layer 3 Switch, in jedem Gateway und jedem VPN-Router implementiert. Die Funktionen, die im Standard definiert sind, sind allerdings etwas rudimentär. Der Master-Router verschickt von seinem lokalen (privaten) Interface VRRP-Hello-Pakete. Falls der Backup-Router keine Pakete mehr bekommt, nimmt er an, dass der Master ausgefallen ist, und übernimmt die volle Masterfunktion, inklusive dessen IP- und MAC-Adresse. Damit erkennt man allerdings nur ein defektes lokales Interface oder einen Totalausfall. Wenn aber andere, wichtige Schnittstellen oder virtuelle Interfaces, z.B. IPSec-SAs, ausfallen sollten, wird dies vom Standard nicht abgedeckt.

An diesem Punkt greifen die herstellereigenen Erweiterungen ein. Es werden zusätzliche Funktionen wie Critical Interface oder Interface Groups angeboten, um weitere Schnittstellen zu überwachen oder mehrere Schnittstellen zu gruppieren und diese Gruppe wie ein einziges Interface zu verwalten. Gute Implementierungen bieten sogar an, Schnittstellen gezielt abzuschalten, falls ein System seinen Masterstatus verliert.

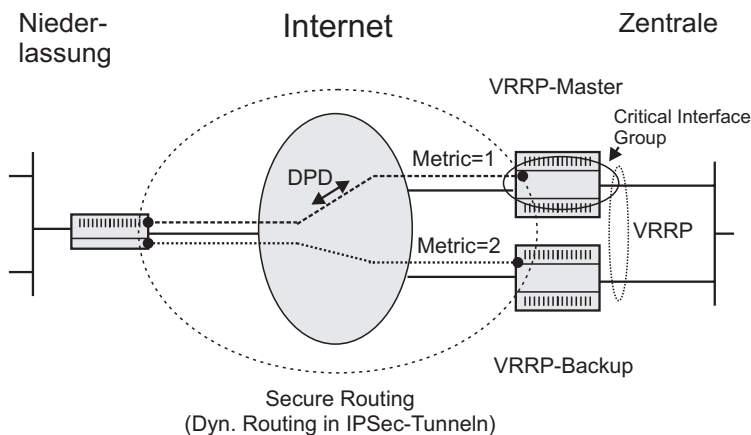


Abbildung 11.22: VRRP in Verbindung mit dynamischem Routing erlaubt das Design hoch verfügbarer VPNs.

In Abbildung 11.22 ist, mit den beiden zentralen Systemen und einer beispielhaften Außenstelle, dargestellt, wie das Ganze funktioniert. Der VRRP-Master bekommt alle Pakete, die zu einer der Außenstellen gesendet werden müssen, und leitet sie weiter. Die IPSec-Tunnel haben eine Routing-Metrik von 1. Fällt der Router, eines seiner Interfaces oder ein Tunnel, der in einer virtuellen Interface-Gruppe enthalten ist, aus, dann übernimmt der VRRP-Backup die Masterfunktion. Über das dynamische Routing erfahren die Gegenstellen, dass der Tunnel mit der niedrigen Metrik nicht mehr verfügbar und daher der zweite Tunnel mit der höheren Metrik zu verwenden ist.

Ein Nachteil scheint es zu sein, dass der Backup-Router gewissermaßen arbeitslos ist und erst im Fehlerfall des Masters zu arbeiten beginnt. Das muss nicht sein, man kann VRRP auch so konfigurieren, dass beide VPN-Router für den jeweils anderen sowohl Master als auch Backup sind und sich den Datenverkehr teilen. Sobald ein System ausfällt, übernimmt das andere den ganzen Datenverkehr.

Es kann auch wichtig sein, einzelne IPSec-Tunnel zu überwachen und im Fall eines Ausfalls andere Verbindungen zu benutzen. Dafür muss IKE Dead Peer Detection (DPD) aktiviert werden, um schnell genug eine tote Gegenstelle oder eine unterbrochene Verbindung zu erkennen.

Redundanz über IP-Routing zu erhalten, ist ebenfalls eine Möglichkeit, allerdings kann es in großen Netzen einige Zeit dauern, bis die Daten über Ersatzwege transportiert werden können. Allerdings wird in Netzen bestimmter Größe ohnehin fast immer Routing oder Layer 3 Switching (was das Gleiche ist) eingesetzt, und die VPN-Router müssen dies ebenfalls beherrschen. In solch einem Szenario kann man, wie Sie in Abbildung 11.23 sehen, auch auf dynamisches Routing in den IPSec-Tunneln verzichten und den Failover-Mechanismus auf die Außenstellen verlagern. Hier erkennen die VPN-Router, wenn eine Verbindung zur Zentrale ausgefallen ist, und bauen einen anderen Tunnel zum Backup-System auf. Der zentrale VPN-Router, der die neue Verbindung annimmt, verteilt anschließend ein Routing-Update im lokalen Netz.

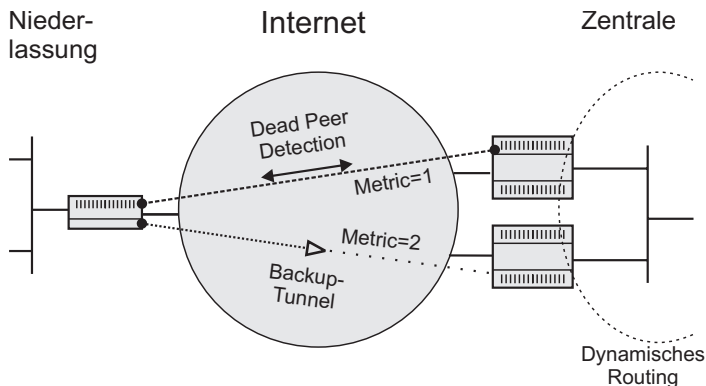


Abbildung 11.23: Auch ohne VRRP kann man hoch verfügbare Netze aufbauen, allerdings erfolgt die Umschaltung nicht ganz so schnell.

Speziell für kleine Außenstellen, die nur mit geringer Bandbreite angebunden werden, gibt es die Möglichkeit, Dial-Backup zu verwenden, also eine ausgefallene DSL- oder Festverbindung temporär durch eine Wählverbindung, meist ISDN, zu ersetzen. Gerade in Deutschland wird dies, obgleich betriebswirtschaftlich meist sehr fragwürdig, sehr oft für kleine Büros oder Heimbüros gefordert.

Hier findet meist nur ein Verbindungs-Backup statt, der Ausfall eines Gerätes wird nicht abgefangen. Manche Systeme bieten allerdings auch andere Möglichkeiten zur Aktivierung der ISDN-Verbindung als nur den Ausfall der primären Schnittstelle. So kann man in guten Systemen auch aufgrund einer gelöschten Route, der Nichterreichbarkeit bestimmter Systeme per PING oder zu bestimmten Tageszeiten die Backup-Verbindung aktivieren.

Beim ISDN-Dial-Backup gibt es prinzipiell drei Szenarien, wovon eines sehr kostenintensiv ist:

Die richtig teure Möglichkeit ist die, eine eigene ISDN-Einwählinfrastruktur aufzubauen. Hier müssen je nach Größe des Netzwerks ein oder mehrere Primärmultiplex-Anschlüsse (PRI, S2M) angemietet und ein eigener Remote-Access-Konzentrator betrieben werden. Hier kann IPSec benutzt werden.

Deutlich günstiger wird es, wenn die ISDN-Calls im POP (Point of Presence) eines Carriers oder Service Providers terminiert und per L2TP in das Kundennetz getunnelt werden. Hier sollte IPSec benutzt werden.

Die kostengünstigste Möglichkeit ist die, über das ISDN-Interface eine normale Internet-Verbindung aufzubauen. Allerdings ist das auch die unsicherste Möglichkeit, denn hier muss man sich wieder auf das Internet verlassen. Hier muss IPSec benutzt werden.

Je nach den Features der eingesetzten VPN-Router kann man die Geschichte elegant konfigurieren, und zwar so, dass für die Backup-Verbindung kein Tunnel aufgebaut werden muss, sondern der alte weiter benutzt wird. Das wird möglich, wenn der IKE-Initiator seine Verbindung nicht fest von einem bestimmten Interface aus aufbauen muss, sondern der VPN-Router den besten Weg (Route) festlegen kann.

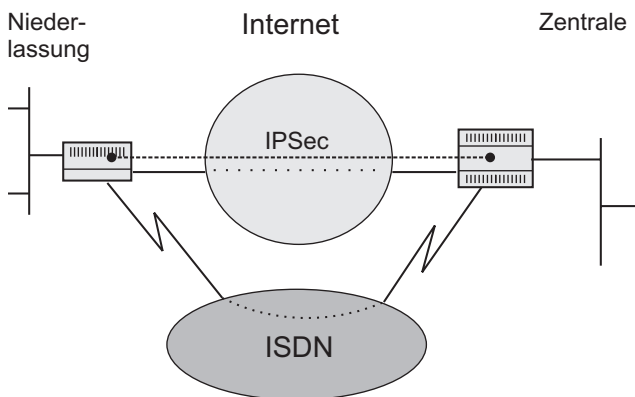


Abbildung 11.24: Hier bleibt bei Ausfall der primären Verbindung der Tunnel bestehen. Die IPSec-Pakete werden jedoch über die ISDN-Wählverbindung transportiert.

In ganz großen Umgebungen, wenn mehr als zwei zentrale VPN-Router eingesetzt werden, kann man auch mit externen Systemen zur Lastverteilung und Backup arbeiten. Dies sei hier nur kurz erwähnt, denn wie solch ein Design aussieht, hängt entscheidend von der Funktionalität der VPN- und Load-Balancing-Systeme ab.

A Anhang

A.1 Literatur

A.1.1 Access- und Netzwerktechnologie

Smith, Philip: Frame Relay – Principles and Applications. Addison-Wesley 1993.

Wilde, Alexander: SDH in der Praxis. VDE Verlag 1999.

Ginsburg, David: Implementing ADSL. Addison-Wesley 1999.

Starr, Thomas u.a.: xDSL: Eine Einführung. ISDN, HDSL, ADSL und VDSL. Addison-Wesley 2000.

Huitema, Christian: IPv6 Architektur und Implementierung. Addison-Wesley 2000.

Moy, John T.: OSPF. Anatomy of an Internet Routing Protocol. Addison-Wesley 1998.

Doraswamy, Naganand, Harkins, Dan: IPSec. Der neue Sicherheitsstandard für das Internet, Intranet und virtuelle private Netze. Addison-Wesley 2000.

Spenneberg, Ralf: VPN mit Linux. Grundlagen und Anwendung Virtueller Privater Netzwerke mit Open Source-Tools. Addison-Wesley 2004.

Lee, Thomas: Microsoft Windows 2000 TCP/IP Protocols and Services Technical Reference. Kapitel 20 und 21. Microsoft Press 2000.

Shea, Richard: L2TP. Implementation and Operation. Addison-Wesley 2000.

Carlson, James D.: PPP. Design and Debugging. Addison-Wesley 1998.

Schneier, Bruce: Angewandte Kryptographie. Protokolle, Algorithmen und Sourcecode in C. Addison-Wesley 1996.

Wobst, Reinhard: Abenteuer Kryptologie. Methoden, Risiken und Nutzen der Datenverschlüsselung. 3. Auflage Addison-Wesley 2001.

Ihringer, Thomas: Diskrete Mathematik. Teubner Verlag 1999.

Pflegger, Charles P.: Security in Computing. Prentice Hall 1997.

- Strobel, Stefan:** Firewalls für das Netz der Netze. Sicherheit im Internet, Einführung und Praxis. dpunkt.verlag 1997.
- Stallings, William:** Sicherheit im Internet. Anwendungen und Standards. Addison-Wesley 2001.
- Kaufman, Charlie, Perlman, Radia, Speciner, Mike:** Network Security. Private Communication in a Public World. Prentice Hall 1995.
- Bitzer, Frank, Brisch, Klaus M.:** Digitale Signatur. Grundlagen, Funktion und Einsatz. Springer-Verlag 1999.
- Warwick, Ford, Baum, Michael S.:** Secure Electronic Commerce. Building the InfraStructure for Digital Signatures and Encryption. Prentice Hall 1997.
- Beutelspacher, Albrecht, Jörg Schwenk, Klaus-Dieter Wolfenstetter:** Moderne Verfahren der Kryptographie. Von RSA zu Zero-Knowledge. 4. Auflage. Vieweg 2001.
- Huston, Geoff:** Internet Performance Survival Guide. QoS Strategies for Multiservice Networks. Wiley & Sons 2000.
- Black, Daryl P.:** Building Switched Networks – Multilayer Switching, QoS, IP Multicast, Network Policy and Service Level Agreements. Addison-Wesley 1999.
- Kuppinger, Martin:** Microsoft Windows 2000 im Netzwerk. Planung, Installation und Verwaltung von Windows 2000-Netzwerken. Microsoft Press 2002.
- Spohn, Darren L.:** Data Network Design, Third Edition. McGraw-Hill/Osbourne 2002.
- Kouti, Sakari, Mika Seitsonen:** Inside Active Directory. A System Administrator's Guide. Addison-Wesley 2002.
- Izzo, Paul:** Gigabit Networks. Standards and Schemes for Next-Generation Networking. John Wiley & Sons 2000.
- Raeppele, Martin:** Sicherheitskonzepte für das Internet. Grundlagen, Technologien und Lösungskonzepte für die kommerzielle Nutzung. 2. Auflage. dpunkt.verlag 2001.

Die folgenden Links bieten Zugriff auf einige der im VPN- und Sicherheitsbereich wichtigsten Organisationen und Unternehmen:

URL: <http://www.icsalabs.com>

URL: <http://www.vpnc.org>

URL: <http://www.ietf.org>

URL: <http://www.bsi.de>

URL: <http://www.nist.gov>

URL: <http://www.nsa.gov>

A

- Abel'sche Gruppe 130
- ABR 62
- Access-Technologien 353
- Accounting 68
- ACT 200
- Active Directory 409
- Adaptive-Chosen-Plaintext-Angriff 106
- Adleman, Leonard 148
- Admission Control 401
- Adressmanagement 72
- ADSL 362
 - Modem 364
 - Splitter 364
- Advanced Encryption Standard 127
- Advanced Research Projects Agency 24
- AES 128
- Analogmodem 370
- Anti-Clogging-Token 200
- ANX 30
- ARPA 24
- ARS 174
- Asymmetric DSL 362
- Asynchronous Transfer Mode 22, 61
- ATM 22, 61
 - Service-Klassen 61
- Authentication Header 184
- Authentifizierung 69

B

- B2B 29
- B2C 29
- BECN 60
- Benutzerauthentifizierung 51, 99
- Benutzerautorisierung 52
- Betriebskosten
 - Senkung 32
- Betriebssicherheit 54
- Breitbandkabel 366
- Brute-Force-Angriff 105, 114, 384

C

- CBC 171
- CBR 62
- CCP 298, 301
- Certificate Management Protocol 390
- Challenge Handshake Authentication Protocol 298–299
- CHAP 298–299
- CHAP-Challenge 300
- Chiffriermaschinen 103
- Chosen-Plaintext-Angriff 106
- Cipher Block Chaining 126
- Cipher Suite 261
- CIR 61
- Client Policy Enforcement 415
- CMP 390
- Codec 419
- Collision Domain 48
- Committed Information Rate 61
- Compression Control Protocol 298
- Compulsory Tunneling 80
- Congestion 60
- Cookies 200

D

- DARPA 25
- Data Encryption Standard 98, 118
- Datenintegrität 51
- Datenkommunikation 19
- Datennetzwerke 23
- Datenvertraulichkeit 49
- DDoS 52
- Dead Peer Detection 250
- DE-Bit 61
- Decapsulation 75
- Denial of Service 52
- Denial-of-Service-Angriff 34, 99, 171
- Dense Wavelength Division Multiplexing 20

- DES 118, 384
 - Algorithmus 120
 - Entschlüsselung 123
 - Funktion 121
 - S-Boxen 121
 - Schlüsseltransformation 120
 - Teilschlüssel 121
 - Triple-DES 123
- Dezentralisierung 31
- DHCP 72
- DHCP-Server 72
- Differentiated Services 63, 335, 400
- Differentiated Services Code Point 335
- Differentielle Kryptoanalyse 106
- Diffie, Whitfield 147
- Diffie-Hellman 199, 384
- Diffie-Hellman-Merkle-Verfahren 147
- Diffie-Hellman-Verfahren 153
- DiffServ 63, 335, 400
 - Anti Replay Service 344
 - Architektur 335
 - ARS 344
 - Assured Forwarding 340
 - Best Effort 341
 - Class Selector 340
 - Edge Router 335, 337
 - Expedited Forwarding 340
 - PHB 339
 - PHB Router 338
- DiffServ Code Point 63
- DiffServ-Router 341
- DiffServ-Service-Klassen 337
- Diffusion 107
- Digital Subscriber Line 22, 362
- Digitale Festverbindungen 59
- Digitale Standardfestverbindungen 367
- Digitale Synchrone Hierarchie 22
- Digitale Zertifikate 71, 394
- Directory Services 73
- Distributed DoS 52
- DNS (Domain Name System) 72
- DoS 52
- DPD 250
- DSCP 63, 335, 339
- DSL 22, 362, 385
- DSLAM 364
- DWDM 20
- Dynamic Host Configuration Protocol 72

E

- EAP 301
- EAP over LAN 417
- EAPoL 417
- ECC 158
- ECDH 160
- ECDLP 158
- ECHELON 102
- Edge-Router (DiffServ) 335
- EIR 61
- Elliptic Curve Cryptography 158
- Elliptic Curve Diffie-Hellman 160
- Elliptic Curve Discrete Logarithm Problem 158
- Encapsulating-Security-Payload 187
- Encapsulation 75
- Ende-zu-Ende-Modell 77
- Enigma 103
- Ethernet 48
- Eulersche Totientenfunktion 153
- Excess Information Rate 61
- Exklusiv-oder-Verknüpfung 112
- Extended Authentication 254
- Extensible Authentication Protocol 301

F

- Fair Queueing 330
- Falltür-Funktion 147–148
- FEC 85
- FECN 60
- Federal Information Processing Standard Publication 118
- Feistel-Netzwerke 111, 120
- Feistel-Zyklus 111
- Fermats Theorem 153
- Festverbindung
 - Digital 2MS 368
 - Digital 2MU 368
 - Digital 64S 368
 - Digital 64S2 368
- FIFO 329
- FIPS-Pub 118
- Firewalls 52, 70
- First In First Out 329
- Forwarding Equivalence Class 85
- FQ 330

Frame Relay 21, 59
Frame Relay Forum 21
FRF 21

G

G.711 419
G.729 419
Galois-Feld 130
Geburtsparadoxon 166
Gefahrenanalyse 94
General Packet Radio Service 353
Global System for Mobile
Communication 353
Globalisierung 31
GPRS 353, 355
GSM 40, 353
GSM-Netz 353

H

Hash-based Message Authentication
Code 169
Hash-Funktionen 163
HDLC 296
Hellman, Martin 147
High Speed Serial Interface 370
Hintertür-Funktion 147
HMAC 165, 169
HMAC-MD5-96 173
HMAC-SHA1-96 173
Hot Spot 360
HSSI 370
HTTP 67
HTTPS 257
Hyper Text Transfer Protocol 67

I

IEEE 802.1/p 58
IEEE 802.11 360
IEEE 802.1Q 58
IEEE 802.1X 417
IETF 39

IKE 216, 384
Aggressive Mode 216, 233
Authentifizierung 224
Dead Peer Detection 250
Hardware-Beschleuniger 240
Main Mode 216, 225
NAT 242
NAT Discovery 243
NAT Keep-Alive 249
NAT-D Payload 243
NAT-OA Payload 246
New Group Mode 216
Quick Mode 216, 237
R_U_THERE Message 250
Sicherheit 252
UDP Encapsulation 246
Initialisierungsvektor 127
Initiator 202
Integrated Services 334
Integrated Services Digital Network 371
Interface Rate Limiting 401
Interface-Sicherheit 53
International Standardization
Organisation 24
International Telecommunication
Union 21
Internet 25
Backbone 37
Verfügbarkeit 39
Internet Engineering Task Force 39
Internet Key Exchange 195
Internet Key Exchange-Protokoll 216
Internet Security Association and Key
Management Protocol 196
Internet Service Provider 38
Internet-Protokoll 24
Interoperabilität 73
Intra-Provider-Modell 76
IntServ 334
IP 24
Class of Service 37
CoS 37
Protokoll 37
IP Control Protocol 298
IP Telephony 422
IP Version 6 26
IPCP 302

IPSec 26, 81, 98

- Adressmanagement 387
- Anti_Replay Service 174
- Anti Replay Window 179
- Architektur 176
- Attribute 217
- Authentication Header 184
- Authentifizierung 173
- Authentizität 99
- Client 82
- Compression 412
- Datenintegrität 98
- Datenverschlüsselung 171
- DiffServ 346
- Encapsulating Security Payload 187
- Hardwarebeschleuniger 193
- Implementierungen 191
- Integritätsprüfung 173
- Komprimierung 193
- Lifetime 179
- Mode 179
- Paketauthentifizierung 170
- Paketintegrität 169
- Paketverarbeitung 183
- Performance 192
- PMTU Parameters 179
- QoS 347
- SAD 177
- Schlüsselmanagement 99
- Security Association Database 177
- Security Parameter Index 178
- Security Policy 180
- Security Policy Database 180
- Selektoren 180
- Sequence Number 179
- Sequence Number Overflow 179
- Sicherheitsassoziation (SA) 177
- SPD 180
- SPI 178
- Standardisierung 176
- Transport-Modus 181
- Tunnel Destination 179
- Tunnel-Modus 181
- Zukunft 194

IPSec Domain of Interpretation 195

IPSec secured L2TP 312

IPSec-DOI 196

IPT 422

IPv6 26

IPX/SPX 23

ISAKMP 196

- Aggressive Exchange 213
- Austauschvorgänge 212
- Authentication Method 219
- Authentication Only Exchange 213
- Authentifizierung 197
- Base Exchange 212
- Cookie 202
- Delete Payload 210
- Denial-of-Service-Angriff 199
- Encryption Algorithm 219
- Field Size 218, 221
- Group Attributes 220
- Group Order 219, 221
- Hash Algorithm 219
- Hash Payload 208
- Identity Protection Exchange 213
- Informational Exchange 213
- Key Length 218, 221
- Nonce Payload 208
- Notification Payload 209
- Phasen 211
- Pre-Shared Secret 197
- PRF 218, 220
- SA Life Duration 218, 220
- SA Life Type 220
- SA Payload 204
- Schlüsseltausch 198
- Sicherheit 197
- Sicherheitsassoziationen 196
- Signature Payload 208
- Vendor ID Payload 211

ISDN 21, 371

B-Kanal 372

BRI 371

D-Kanal 372

S0-Bus 372

ISDN-Router 396

ISO 24

ISP 38

ITU-T 21

K

Kabelmodem 366

Klassifizierer 337

Kollision 163

Kollisionen 166

Konfiguration 66

Konfusion 107
Konvergenz 17
Kryptoanalyse 101, 105
Kryptografie 107
Kryptologie 101

L

L2TP 79, 295, 303
 Attribute Value Pairs 308
 AVP 308
 AVP-Verschlüsselung 309
 Call 304
 Compulsory Tunneling 306
 Datenpakete 306
 LAC 305
 LNS 305
 PPP-Tunneling 305
 Sicherheit 311
 Steuerungspakete 306
 Tunnelaufbau 310
 Tunnelauthentifizierung 311
 Tunneling-Modelle 305
 Virtueller LAC 306
 Virtueller Remote Access 304
 Voluntary Tunneling 306
L2TP/IPSec 295, 312, 314, 405
 DiffServ 350
 Paketformat 314
 Performance 314
 QoS 351
L2TP/IPSec-Transport 295
Label Edge Router 85
Label Information Base 84
Label Switch Router 85
Label Switched Path 85
Label-Distribution-Protokoll 85
LAN 48
 Switching 48
 V-LAN 48
Lauschangriff 47
Lawineneffekt (Permutation) 110
Layer 2 Forwarding 87
Layer 2 Tunneling Protocol 79, 295
Layer-2-Tunneling 78
Layer-3-Tunneling 81
LCP 298
LDAP 73, 390

LDP 85
LER 85
LIB 84
Lightweight Directory Access
 Protocol 73, 390
Link Control Protocol 298
LSP 85
LSR 85
Lucifer-Chiffre 111, 118
LZS 302

M

MAN 29
Management 66
Management Information Base 67
Man-in-the-Middle-Angriff 154
Massachusetts Institute of Technology 148
MD5 164
Meet-in-the-Middle-Angriff 125
Merkle, Ralph 147
Message Digest 5 164
Metropolitan Area Network 29
MIB 67
MIB-Variablen 68
Microsoft Certificate Service 405
Microsoft Point-to-Point Encryption 87
Migrationsphase 71
MIT 148
MobiKE 247
Mobile IKE 247
Mobile Internet 28
Mobilität 31
Modulare Arithmetik 150
MPLS 82
MPPC 302
MPPE 87
MS-CHAP 300
MS-CHAPv2 87, 300
Multi Protocol Label Switching 82

N

NAT 72, 242, 396
NAT/PAT-Systeme 242
National Bureau of Standards 118
National Institute of Standards and
 Technology 118

National Security Agency 102, 118
NAT-Traversal 246
NBS 118
Network Address Translation 72, 242, 396
NIST 26, 118, 127
NSA 26, 102

O

Oakley Key Determination Protocol 213
Oakley-Gruppe 2 _ 5 214
Oakley-Gruppe-1 214
Öffentlicher Schlüssel 115
Open Systems Interconnect Reference
Model 23
OSIRM 23
Out-of-Band-Zugriff 389

P

Padding 117
Paket-Authentifizierung 50
PAP 299
Password Authentication Protocol 298
PAT 242
P-Box 110
PDH 20
Per Hop Behaviour 335
Perfect Forwarding Secrecy 216
Permanent Virtual Circuit 60
Permutation
Expansion 110
Kompression 110
PFS 216
PHB 335
PHB-Router 338
PKI 70, 390, 415
PKIX 394
Plesiochrone Digitale Hierarchie 20
Point of Presence 44
Point-to-Point Protocol 295
Point-to-Point Tunneling Protocol 87
Policy-Server 338
POP 44
Port Address Translation 242
PPP 295
PPP over Ethernet 364

PPPoE 364
PPP-Verbindungsaufbau 302
PPTP 87, 146
Pre-Imaging Attack 167
Pre-Master Secret 265
Primärmultiplexanschluss 41
Primzahlen 149
Erzeugung 150
Häufigkeit 149
Privater Schlüssel 116
PRNG 162
Provider-Enterprise-Modell 76
Pseudo Random Number Generator 162
Pseudo-Zufallszahlen 161
Public Key Infrastructure 390
Public Key Infrastructure X.509 394
Public-Key-Infrastruktur 70
Public-Key-Kryptografie 148
Public-Key-Verfahren 383
Push Configuration 392, 409
PVC 60

Q

QoS 56, 317
Bandbreite 325
Congestion Delay 324
Delay 322
Fehlerrate 325
Flussbasierend 333
Im LAN 57
Im VPN 64
Im WAN 58
In ATM 61
In dig. Festverbindungen 59
In Frame Relay 59
In IP 63
In Sprachnetzen 56
Jitter 324
Klassenbasierend 335
Kriterien 320
Processing Delay 323
Propagation Delay 323
Serialization Delay 323
Verfügbarkeit 320
Verzögerung 322
Verzögerungsvarianz 324
Quality of Service 56, 317
Queueing 329

R

- Rabin-Miller-Algorithmus 150
- RAC 41, 296
- RADIUS 69
- Random Early Discard 328
- Random Function 161
- RC4 141
 - Implementierung 144
 - Initialisierung 142
 - Kryptoanalyse 145
 - KSA 142
 - PRGA 142
- Real Time Protocol 422
- RED 328, 332
- Remote Access Concentrator 41, 296
- Replay Attack 170, 174
- Request for Comment 38
- Responder 202
- Ressource Reservation Protocol 334
- RFC 38
- Rijndael 129
 - Algebraische Angriffe 140
- Rijndael-Algorithmus 133
- Rivest, Ronald 142, 148
- RND 161
- Rotormaschinen 103
- Routing 401
- RSA 148, 199, 383
 - Faktorisierung 156
- RSA-Verfahren 155
- RSVP 334, 400
- RTP 422

S

- S2M 41
- SA Authentication Algorithm 217–218
- SA Life Duration 217
- SA Life Type 217
- Satellitenverbindungen 360
- S-Box 108
- Schlüsselmanagement 50
- SDH 22, 369
- SDSL 365
- Secure Hash Algorithm 164
- Secure Hash Standard 164

- Secure Routing 401
- Secure Socket Layer 258
- SecureID 394
- Security Policy 69
- Service Level Agreement 37, 381
- SHA 164
- Shamir, Adi 148
- Shaper 337
- Sicherheit
 - Analyse 94
 - Unternehmensdatennetze 93
- Sicherheitsstrategie 69
- Sicherheitstechnologie 93
- Simple Network Management Protocol 66
- SLA 37, 381, 412
- Small Office/Home Office 396, 416
- SNA 23
- SNMP 66
- SOHO 396, 416
- SONET 22, 25
- Spaltentransposition 109
- Split-Management 47
- Split-Tunneling 393
- SSL 28, 100, 257, 411
 - Alert Protocol 276
 - Anti-Replay Service 268
 - Cipher Suite 261
 - Compression 278
 - Connection State 269
 - Diffie-Hellman 263
 - Fragmentierung 278
 - Handshake Protocol 270, 274
 - Hardware-Beschleuniger 282
 - Integritätssicherung 267
 - Kryptographie 261
 - Payload Protection 280
 - Performance 282
 - Record Protocol 277
 - RSA 263
 - Schlüsselerzeugung 262
 - Session State 269
 - Session-Schlüssel 265
 - Sicherheit 289
 - SSL-VPN 283
 - Verschlüsselungsalgorithmen 266

- SSL-Sicherheit
 - Bleichenbacher Angriff 290
 - Cipher Suite Rollback Angriff 291
 - Falsche Zertifikate 290
 - Key-Exchange Algorithm Rollback-Angriff 292
 - Version Rollback-Angriff 291
 - Web-Spoofing 289
- SSL-VPN 257, 413
 - Browser-based 284
 - Client-based 285
 - DiffServ 348
 - Enhanced Browser-based 285
 - Nagle-Algorithmus 350
 - Network Clients 287
 - QoS 349
- SSL-VPN-Gateway 288
- SSS
 - Change Cipher Spec 276
- Strict Priority Queueing 331, 333
- Substitution 107
 - Monoalphabetisch 108
 - Nichtlinear 109
 - Polyalphabetisch 108
- SVC 60
- Switched Virtual Circuit 60
- Symmetric DSL 365
- Symmetrische Verfahren 383
- Synchronous Optical Network 22, 25

T

- TCO 32
- TCP 326
 - Flusssteuerung 326
 - Nagle-Algorithmus 328
 - Slow Start-Algorithmus 327
 - Window 326
- TCP/IP-Modell 24
- TLS 100, 260
- Token-Karten 394
- Total Cost of Ownership 32
- Transmission Control Protocol 326
- Transport Layer Security 260
- Trap 66

- Trap Door Function 147–148
- Triple-DES 123, 384
- Tunneling 75
- Tunneling-Modelle 76
- Tunneling-Protokolle 75

U

- Übertragungstechnologie 19
- Überwachung (Management) 68
- UBR 62
- UDP Encapsulation 247
- UMTS 356
 - Luftschnittstelle 357
 - Phase 1 358
 - Phase 2 359
 - Zonen 358

V

- VBR 62
- VC 20
- Verfügbarkeit 54
- Verkehrsbeziehungen 376
- Verkehrskategorien 318
- Verkehrsprofil 376
- Verschleierung 103
- Verschlüsselung 97, 103
 - Asymmetrische 115
 - Datenblock 116
 - Datenstrom 117
 - Public Key 113
 - Secret Key 113
 - Symmetrische 113
 - Verfahren 113
- Verzeichnisdienste 73, 390
- Virtual Circuit 20
- Virtual LAC 305
- Virtual Router Redundancy Protocol 399
- VLAN
 - Im LAN 58
- VoIP 419
- Voluntary Tunneling 80

VPN 17

- Administration 390
- Arten 40
- Ausfallsicherheit 378
- Autorisierung 378
- Bandbreitenmanagement 398
- Branch-Office 43
- Clients 391
- Design 373
- Dienste 46
- Durchsatz 385
- Extranet 45
- IP 46
- IP-Adressmanagement 387
- IP-VPN 37
- Ist-Aufnahme 374
- Loadbalancing 398
- Logging 388
- Marktentwicklung 36
- Migrationsphase 382
- QoS 342
- Quality of Service 400
- Redundanz 399
- Remote-Access 41
- Security Policy 383
- Sicherheit 389
- Site-to-Site 43
- Systemdurchsatz 397
- Übergangsphase 382

VPN-Design

- Lösungsbeispiele 404
- Remote Access 404, 411
- Standortverbindungen 418

VPN-Markt 35

VPN-Routing 401

- BGP 403
- OSPF 403
- RIPv2 402
- Statisch 402

VRRP 399, 424

W

WDM 20

Webbasiertes Management 67

Weighted Fair Queueing 328

WEP 145

WFQ 328

Windows 2000

- Server 405

- Virtual LAC 305

- VPN-Client 312, 395

- VPN-Protokoll 295

- VPN-Server 405

Windows 2003 Server 412

Windows XP

- Virtual LAC 305

- VPN-Client 312, 395

- VPN-Protocol 295

Wireless LAN 360

WLAN 360

World Wide Web 27

WRED 332

X

X.25 20

X.509 390

XAUTH 254

Z

Zufallszahlen 161

Zugriffsschutz 53



Copyright

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt. Dieses eBook stellen wir lediglich als persönliche Einzelplatz-Lizenz zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschliesslich

- der Reproduktion,
- der Weitergabe,
- des Weitervertriebs,
- der Platzierung im Internet, in Intranets, in Extranets,
- der Veränderung,
- des Weiterverkaufs
- und der Veröffentlichung

bedarf der schriftlichen Genehmigung des Verlags.

Insbesondere ist die Entfernung oder Änderung des vom Verlag vergebenen Passwortschutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: info@pearson.de

Zusatzdaten

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. Der Rechtsweg ist ausgeschlossen.

Hinweis

Dieses und viele weitere eBooks können Sie rund um die Uhr und legal auf unserer Website



herunterladen