Kapitel 6

VPN-Verfahren

Um die drei Ziele einer vertrauenswürdigen Daten-Übertragung – Vertraulichkeit, Authentikation und Integrität – mit einem VPN zu realisieren, müssen eine Reihe von Überlegungen durchgeführt werden. Dieses Kapitel hilft bei der Auswahl von Protokollen, die den gesicherten Transport von Daten und den Austausch der Schlüssel durchführen. Mit dem Protokoll IPSec wurde ein in die Zukunft weisender Standard geschaffen, doch die Eignung von IPSec für eine große Anzahl von Algorithmen macht eine sorgfältige Recherche der Fähigkeiten von VPN-Produkten vor dem Kauf erforderlich. Sonst sind bei IPSec-fähigen Produkten unterschiedlicher Hersteller Probleme vorprogrammiert. Insbesondere beim (automatischen) Austausch geheimer Schlüssel kann IPSec mit sehr unterschiedlichen Verfahren kombiniert werden.

Neben IPSec sind noch eine ganze Reihe anderer Verfahren innerhalb von VPNs implementiert. Microsoft ist mit seinen Produkten erst ab Windows 2000 auf den IPSec Standard umgestiegen – genauer gesagt: hat sich dem IPSec-Standard angenähert, aber eigentlich eine proprietäre Lösung geschaffen – und zahllose VPN-Realisierungen nutzen noch das alte PPTP-Protokoll, trotz seiner Unzulänglichkeiten und Risiken. In der Unix-Welt, wo IPSec in modernen Implementierungen der Standard ist, wurden ältere VPNs häufig mit dem Protokoll Secure Shell (SSH) realisiert. Die Zukunft gehört aber IPSec mit dem Schlüsselaustausch über IKE (Internet Key Exchange), so dass moderne VPN-Implementierungen IPSec und IKE in jedem Falle zur Verfügung stellen müssen.

6.1 VPN-Protokolle

6.1.1 IPSec

Die Grundideen des VPN-Standardprotokolls IPSec wurden im Zusammenhang mit dem neuen Internet-Protokoll IPv6 entwickelt. IPSec wurde von der Internet Engineering Task Force (IETF) definiert. Im Gegensatz zur bisherigen IP-Version 4 können hier optionale zusätzliche Header angegeben werden, mit denen eine erweiterte Funktionalität auf Protokollebene realisiert werden kann.

Zwei dieser Header wurden zu Zwecken der Authentikation und Verschlüsselung definiert. Ihre Namen sind »Authentication Header« (AH) und »Encapsulated Security Payload« (ESP). IPSec kann aber ebenso in Netzwerken nach dem bishe-

rigen IPv4-Standard implementiert werden. Statt der zusätzlichen Header findet dann eine Erweiterung des normalen IP-Header statt. In diesem Fall sind AH und ESP Datenstrukturen innerhalb des Headers. Da die Absicherung der Kommunikation mit IPSec auf IP-Ebene stattfindet, stehen die zusätzlichen Datenstrukturen zwischen dem IP-Header und dem Header für die nächsthöhere Netzwerk-Ebene (TCP oder UDP). AH und ESP können einzeln oder auch gemeinsam eingesetzt werden, wobei dann innerhalb des Netzwerkpakets der AH-Header vor dem ESP-Header stehen muss. Die beiden Header selbst enthalten keine Informationen über die zur Absicherung eingesetzten Algorithmen und Schlüssellängen, sondern nur einen Verweis (Security Parameter Index, SPI) auf eine Datenstruktur mit diesen Informationen (Security Association, SA).

Mit der AH-Datenstruktur kann gewährleistet werden, dass eine eventuelle Manipulation von Daten auf dem Weg durch das Netzwerk entdeckt wird. Außerdem findet die Authentikation des Absenders der Pakete statt. Beim ausschließlichen Einsatz des AH-Headers findet keine Verschlüsselung über IPSec statt. Vertraulichkeit kann dann nur über eine Verschlüsselung außerhalb des IPSec-Protokolls ermöglicht werden.

Mit Hilfe von ESP können Vertraulichkeit der Übertragung, Authentikation des Absenders und Integrität der Daten garantiert werden, da neben der Verschlüsselung auch ähnliche Mechanismen wie in AH definiert werden können. Im Unterschied zu ESP bezieht sich die Authentikation von AH auch auf den IP-Header, so dass die Kombination von AH und ESP Vorteile im Sicherheitsbereich bietet, allerdings mehr Ressourcen auf den beteiligten Rechnern benötigt.

Kryptographische Verfahren in IPSec

IPSec ist von seinem Ansatz her flexibel und zur Zusammenarbeit mit praktisch jedem kryptographischen Verfahren bereit. In der Praxis haben sich allerdings bestimmte de-facto-Standards herausgebildet, die von den meisten IPSec-Implementierungen unterstützt werden:

- Verschlüsselungsalgorithmus: DES im CBC-Modus, Triple-DES im CBC-Modus, AES
- Hash-Algorithmus: SHA-1 oder MD5, jeweils mit HMAC
- Authentikation: durch RSA-Signaturen mit X.509-Zertifikaten, RSA ohne Zertifikate (öffentliche Schlüssel wurden zuvor ausgetauscht) oder durch Pre-Shared Keys
- Austausch von Session-Keys: mittels Diffie-Hellman

Transport- und Tunnelmodus

IPSec kann im Transport- oder im Tunnelmodus betrieben werden. Im Transportmodus wird der IP-Header des ungesicherten IP-Pakets übernommen und nur sein Datenteil verändert. Bis auf das Feld »Länge des IP-Paketes« und die Prüf-

summe bleibt der alte IP-Header unverändert. Im Tunnelmodus hingegen wird das gesamte IP-Paket in die Nutzdaten des IPSec-Pakets übernommen, so dass die alte IP-Adresse nicht mehr notwendigerweise sichtbar sein muss. Bei 1:1-VPNs, die beispielsweise zwischen zwei Firewall-Systemen eingerichtet werden, wird der Tunnelmodus genutzt. Damit bleiben die echten IP-Adressen der Kommunikations-Partner einem Angreifer verborgen. Bei 1:n- oder m:n-VPNs kommt in der Regel der Transportmodus zum Einsatz. In den Abbildungen 6.1 und 6.2 werden die beiden Modi jeweils für die Versionen IPv4 und IPv6 am Beispiel eines TCP-Paketes dargestellt.

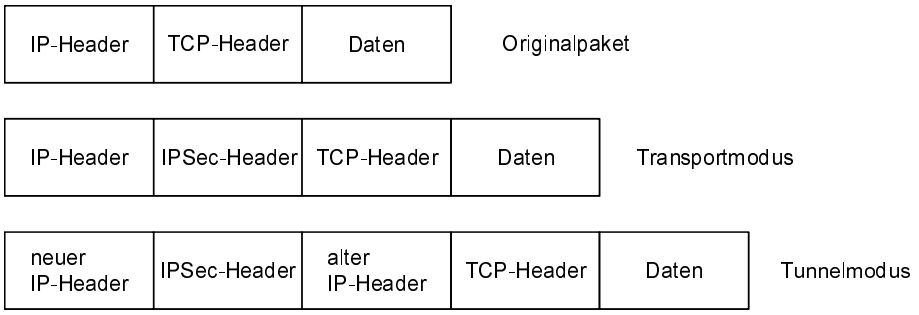

Abb. 6.1: IPSec-Modi bei IPv4

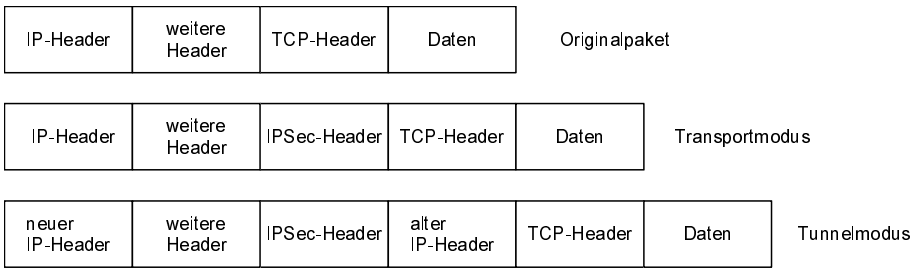

Abb. 6.2: IPSec-Modi bei IPv6

Security Association (SA)

Die Security Associations (SAs) können anschaulich als Definitionen von Filtern verstanden werden, durch die die IP-Pakete beim Versand beziehungsweise beim Empfang geleitet werden. Die Zuordnung zwischen Paket und SA erfolgt über den Pointer »Security Parameter Index« (SPI). Abgehende Pakete werden mit den gewünschten Authentikations- beziehungsweise Verschlüsselungsalgorithmen abgesichert, ankommende Pakete wieder in den allgemein lesbaren Zustand (ohne AH und ESP) gebracht. Werden ESP und AH kombiniert, müssen auch zwei SAs

angegeben werden. Alle SAs eines Rechners werden in einer Datenbank abgelegt, die »Security Associations Database« (SAD) genannt wird. SAs für ankommende und abgehende Pakete sind wegen der zumeist eingesetzten asymmetrischen Verfahren stets getrennt anzugeben. Beim Aufbau einer IPSec-Verbindung werden die SAs zwischen den beiden Partnern ausgehandelt und in der SAD abgelegt (Abb. 6.3). Dabei hat jeder Rechner intern eine Sammlung von möglichen Parametern als Anforderungs-Policies abgelegt, die er mit denen des Partners abgleicht und woraus er die Parameter mit der höchstmöglichen (vordefinierten) Priorität wählt.

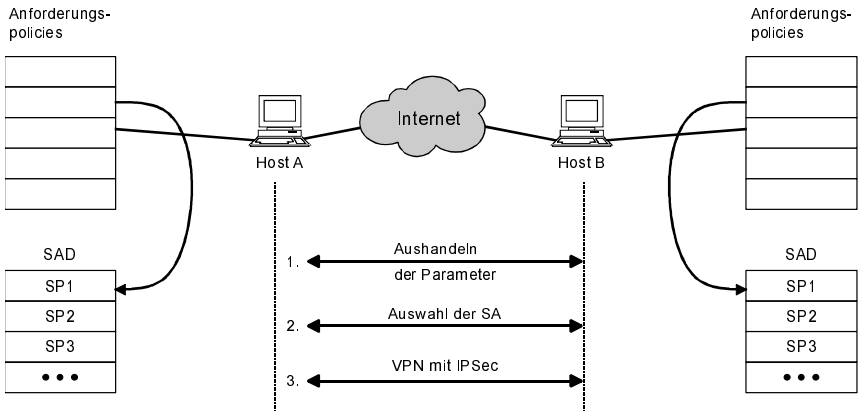

Abb. 6.3: Aushandeln von Security Associations

Jede SA besteht aus einer Datenstruktur, in der die Adresse des Kommunikationspartners, die ausgehandelten Verfahren zur Verschlüsselung, Authentikation, Hashwert-Bildung, die bei der Verschlüsselung beziehungsweise Authentikation eingesetzten kurzlebigen Session-Keys sowie die Gültigkeitsdauer der SA angegeben werden. Ein Beispiel für eine AH-SA zeigt Tabelle 6.1.

IP-Adresse des Partners	205.48.34.263
Security Parameter Index	7B750AC6
Filter-Transformation	AH, MD5 mit HMAC
Session-Key (hier für HMAC)	73DA6710BC651801
weitere SA-Attribute (z. B. Gültigkeitsdauer)	5 Minuten oder 100 KByte

Tabelle 6.1: Beispiel für eine Security Association

Die »Security Policy Database« (SPD) ist eine Sammlung von Regeln, ähnlich denen von Paketfiltern. Für alle Klassen von ankommenden und abgehenden Paketen muss angegeben werden, ob die Pakete mittels IPSec behandelt sind

beziehungsweise behandelt werden müssen (Verweis auf eine SA in der SAD), ob es sich um IP ohne IPSec handelt oder ob die Pakete verworfen werden müssen. Die Einträge in der SPD müssen geordnet sein, da der erste auf das Paket passende Eintrag verwendet wird. Alle im Regelwerk nicht explizit behandelten Pakete werden verworfen.

Der AH-Header

Der AH-Header sichert das gesamte IP-Paket mittels eines MAC ab, einschließlich der unveränderlichen Header-Daten. Flaggen, die sich auf dem Weg des Pakets durch das Internet ändern können, sind vom Schutz durch AH ausgenommen. Abbildung 6.4 zeigt den Einbau des AH-Headers in IP-Pakete am Beispiel eines TCP-Pakets im IPv4-Standard, die Struktur des AH-Headers selbst kann der Abbildung 6.5 entnommen werden.

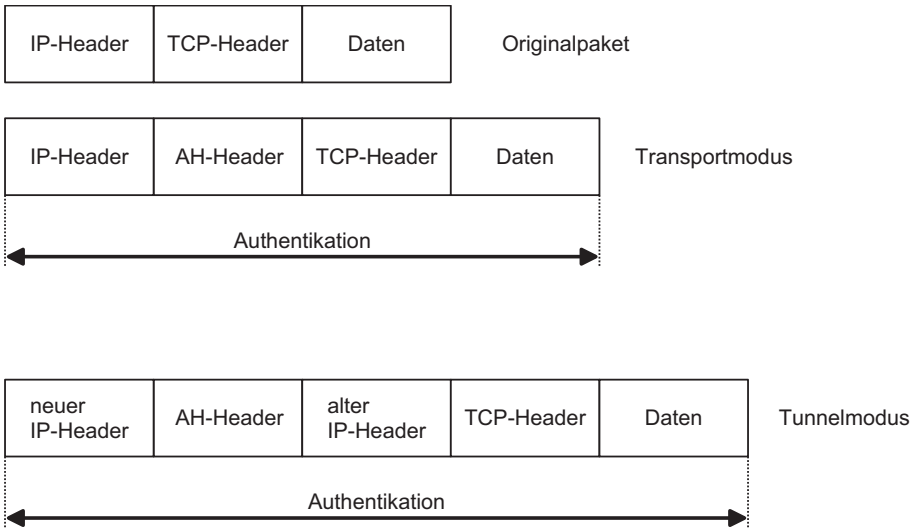

Abb. 6.4: AH-Header im Transport- und Tunnelmodus (IPv4)

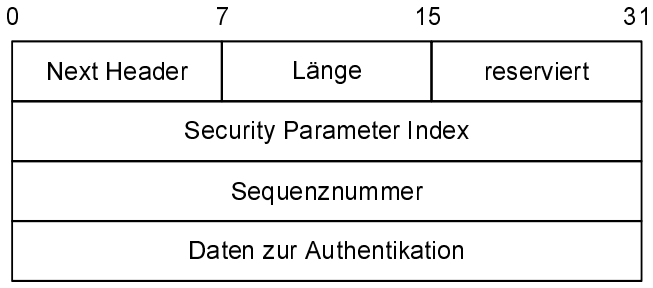

Abb. 6.5: Struktur des AH-Headers

Dabei bedeuten die einzelnen Felder:

- »Next Header« (8 Bit) gibt den Typ der Daten an, die hinter dem AH-Header folgen.
- Die Länge (8 Bit) gibt die Gesamtlänge des AH-Headers an.
- Der reservierte Bereich (16 Bit) ist bis zu einer späteren Verwendung stets zu 0 zu setzen.
- Der Security Parameter Index (SPI, 32 Bit) ist zusammen mit der IP-Zieladresse und der Tatsache der AH-Authentikation ein eindeutiger Verweis auf die für dieses Paket verwendete Security Association (SA).
- Die Sequenznummer (32 Bit) ist ein Zähler, mit dem Replay-Angriffe (böswillige Wiederholung von Paketen) erkannt werden können. Sie wird mit jedem gesendeten Paket um eins erhöht. Die Auswertung der Sequenznummer ist optional und wird in der dazugehörigen SA angegeben.
- Die Authentikationsdaten AD (32 Bit) schließlich enthalten einen verschlüsselten Hashwert oder ähnliches, um die Integrität der Daten überprüfen zu können.

Der AH-Header hat den IP-Protokolltyp 51.

Der ESP-Frame

Mittels des ESP-Frames wird der gesamte auf ESP folgende Teil des Pakets verschlüsselt und authentisiert, unter IPv6 einschließlich etwaiger nachfolgender Header. Die Authentikation über ESP bezieht sich allerdings nur auf den Bereich zwischen dem ESP-Header und dem am Ende des Pakets stehenden ESP-Trailer, wie Abbildung 6.6 am Beispiel eines TCP-Paketes zeigt. Die Struktur des ESP-Frames (Header, Nutzdaten, Trailer) ist in Abbildung 6.7 angegeben.

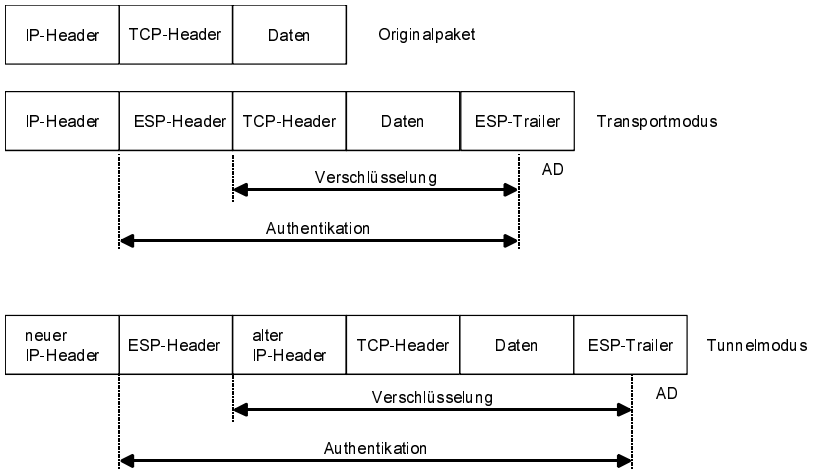

Abb. 6.6: ESP-Header im Transport- und Tunnelmodus (IPv4)

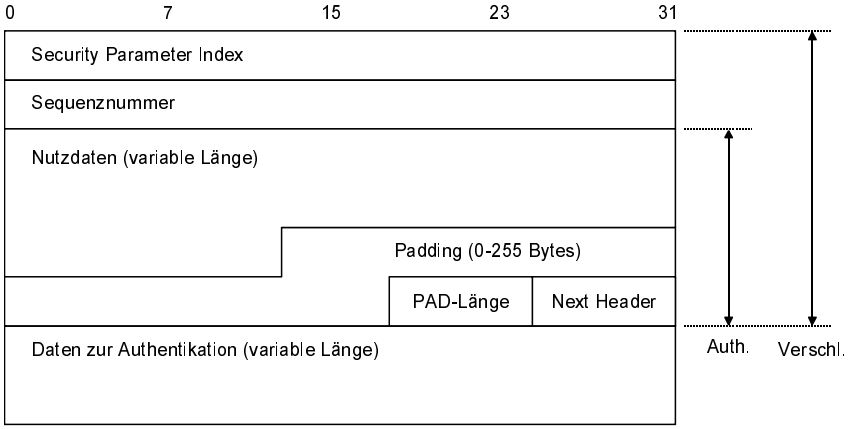

Abb. 6.7: Struktur des ESP-Frames

Dabei haben die einzelnen Felder die folgenden Bedeutungen:

- Der Security Parameter Index (SPI, 32 Bit) hat dieselbe Bedeutung wie beim AH-Header.
- Die Sequenznummer (32 Bit) hat dieselbe Bedeutung wie beim AH-Header.
- Die Nutzdaten (TCP/UDP-Header, Applikationsdaten) können bei Bedarf um etwaige Initialisierungsvektoren für die Entschlüsselung erweitert werden.
- Die Padding-Daten füllen die Nutzdaten bei Bedarf auf, wenn zum Beispiel ein Entschlüsselungsalgorithmus eine feste Blocklänge verlangt.
- Die PAD-Länge gibt die Länge der Padding-Daten an.
- Im Next-Header-Feld ist (wie beim AH-Header) vermerkt, von welchem Typ die Nutzdaten sind.
- Die Authentikationsdaten (AD) haben – im Gegensatz zum AH-Header – eine variable Länge. Die Länge dieses Felds ist implizit durch den verwendeten Algorithmus vorgegeben, der in der zugehörigen SA vermerkt ist.

Der ESP-Header hat den IP-Protokolltyp 50.

Werden AH und ESP kombiniert, so ergibt sich die in Abbildung 6.8 angegebene Paketstruktur.

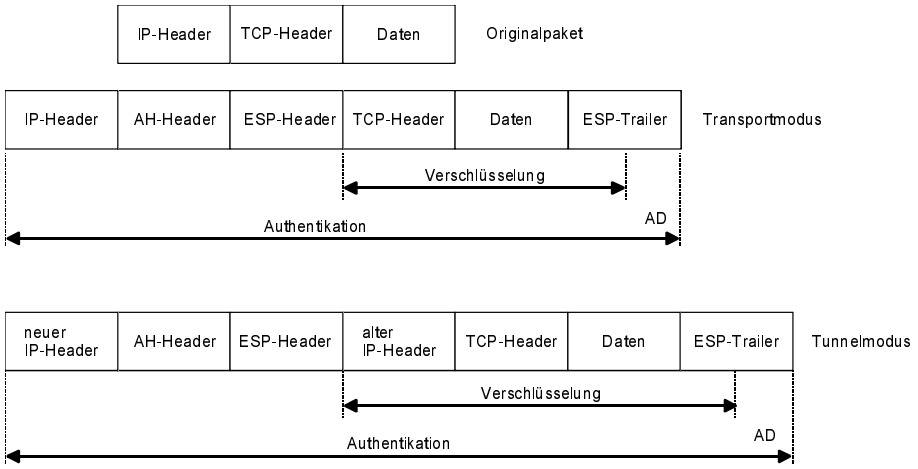

Abb. 6.8: AH/ESP-Header im Transport- und Tunnelmodus (IPv4)

Verarbeitung von abgehenden Netzwerkpaketen

Soll eine Netzwerkverbindung unter IPSec aufgebaut werden, wird zunächst in der Security Policy Database (SPD) nachgeforscht, wie die dazugehörigen Pakete zu verarbeiten sind. Als Suchkriterien dienen dabei die gewünschte Verarbeitung (AH, ESP oder beides), die Ziel-IP-Adresse sowie der Security Parameter Index (SPI). Wird in der Datenbank eine passende Security Association (SA) gefunden, findet mit Hilfe dieses Filters der Umbau des Pakets gemäß dem IPSec-Standard statt. In bestimmten Fällen (beispielsweise AH und ESP) kann auch ein Set von SAs gefunden und angewandt werden.

Falls noch keine passende SA im Zugriff ist, muss diese ermittelt und in der Datenbank abgelegt werden. Dazu müssen unter Umständen Verhandlungen mit anderen Rechnern vorgenommen, Schlüssel ausgetauscht und Algorithmen festgelegt werden. Am Ende dieser Operationen liegt bei beiden Partnern die passende SA (beziehungsweise ein Set von SAs) vor. Von diesem Punkt an werden die Netzwerkpakete umgewandelt wie in der SA beschrieben.

Verarbeitung von ankommenden Netzwerkpaketen

Ankommende IP-Fragmente von Netzwerkpaketen werden zunächst zusammengesetzt. Dann wird, ebenso wie bei abgehenden Verbindungen, zunächst nach einer passenden SA gesucht. Wird keine gefunden, wird das Paket verworfen und ein Eintrag in eine Fehler-Logdatei vorgenommen.

Wird eine SA oder ein Set von SAs gefunden, wird das Paket durch die damit definierten Filter geschleust und erhält seine ursprüngliche Form zurück. Dann findet nochmals ein Vergleich mit der Security Policy Database (SPD) statt, ob das Paket gemäß seinen dort festgelegten Spezifikationen korrekt verarbeitet wurde. Anschließend wird das Paket dem normalen Netzwerkstack zur Verfügung gestellt. Handelt es sich beim Empfänger um ein VPN-Gateway oder ein Firewall-System und befindet sich das Paket im Tunnelmodus, wird es an den endgültigen Empfänger weitergesendet.

6.1.2 Point-to-Point Tunneling Protocol (PPTP)

Bei der Implementierung von VPNs in der »Microsoft-Welt« hat das als veraltert geltende PPTP noch immer einen großen Stellenwert, ist es doch das einzige Protokoll, das von Microsoft vor der Einführung von Windows 2000 offiziell unterstützt wurde. PPTP ist eine Erweiterung des älteren »Point-to-Point Protocol« (PPP), mit der ein Tunneln von PPP-Paketen über IP-Netze wie das Internet möglich wird. Dabei werden bezüglich Authentikation und Verschlüsselung keine neuen Verfahren definiert, da PPP beziehungsweise seine Erweiterungen selbst schon diese Punkte abdecken. Die Aufgabe von PPTP beschränkt sich im Wesentlichen auf den Aufbau und Betrieb des Tunnels, über den eine oder mehrere PPP-Verbindungen gesendet werden (Multilink). Zur Unterscheidung der einzelnen Links wurde ein weiterer Header (GRE) eingebaut. Den gesamten Header bis hin zu den Nutzdaten zeigt Tabelle 6.2.

MAC-Header (Ethernet, Token-Ring, FDDI etc.)
IP-Header
GRE-Header
PPP-Header der getunnelten Verbindung(en)
IP-Header der getunnelten Verbindung(en)
TCP/UDP-Header der getunnelten Verbindung(en)
Nutzdaten der getunnelten Verbindung(en)

Tabelle 6.2: PPTP-Paketaufbau

Außer dem eigentlichen Tunnel mit den verschlüsselten und authentisierten Nutzdaten wird beim Aufbau einer PPTP-Verbindnung eine zweite Steuerverbindung auf TCP-Basis genutzt. Über den TCP-Port 1723 wird eine Client-Server-Verbindung aufgebaut; die an dieser Verbindung beteiligten Systeme werden »PPTP Access Concentrator« (PAC) und »PPTP Network Server« (PNS) genannt (Abb. 6.9). Bei Multilink-Verbindungen können mehrere PPP-Verbindungen über einen PAC geführt werden, zum Beispiel wenn zwei Netze über Firewalls und PPTP miteinander gekoppelt werden.

Folgende Funktionen werden von der TCP-Steuerverbindung wahrgenommen:

- Auf- und Abbau des PPTP-Tunnels
- Kontrolle der ordnungsgemäßen Arbeit des Tunnels durch »keep alive«-Pakete
- Auf und Abbau von physikalischen Verbindungen, zum Beispiel Telefonverbindungen
- Steuerung des PPP-Verkehrs und Fehlerbehandlung

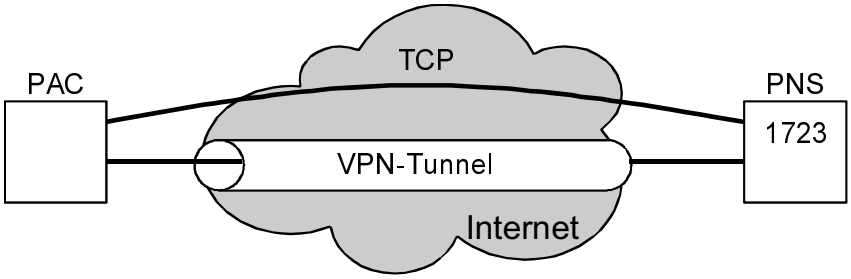

Abb. 6.9: PPTP-Verbindung

PPP-Protokoll

PPP wurde entwickelt, um unterschiedliche Protokolle über Punkt-zu-Punkt-Verbindungen zu schicken. In der Praxis wird meist IP in die PPP-Pakete »eingepackt«, doch können auch andere Protokolle wie zum Beispiel IPX eingesetzt werden.

PPP ist im OSI-Schichtenmodell in Ebene 2 (Media Access Control MAC) eingeordnet. Ebenso wie bei seinen »Kollegen« Ethernet oder Token Ring können mehrere Verbindungen gleichzeitig über eine Kommunikations-Strecke übertragen werden. PPP unterstützte in seiner ursprünglichen Version nur die Authentikation über die nachfolgend beschriebenen Protokolle PAP und CHAP, die aktuelle Variante erlaubt eine Verschlüsselung nach DES-CBC.

Authentikation

Die Authentikation über das »Password Authentication Protocol« (PAP) kann heutigen Sicherheitsanforderungen nicht genügen. Es erhält von der Zugangssoftware die Kennung und das Passwort des Benutzers und übermittelt diese mit PPP-Paketen im Klartext an den Partner, und zwar so lange, bis eine positive oder negative Quittung erhalten wird oder die Verbindung vom Partner beendet wird.

Das »Challenge Handshake Authentication Protocol« (CHAP) hingegen ist ein Challenge-Response-Verfahren, bei dem die Windows-Passwörter in Form eines bei jeder Anmeldung wechselnden MD4-Hashs übertragen werden. Replay-Attacken sind somit unmöglich. In den meisten Implementierungen von CHAP

authentisiert sich der Client am Server, doch eine zusätzliche Authentikation des Servers kann problemlos in die Kommunikation integriert werden.

Bewertung der Sicherheit

Obwohl sich PPTP im Bereich der Microsoft-Netzwerke allgemein durchgesetzt hat, muss das Design des Protokolls als kritisch eingeschätzt werden. Die TCP-Steuerverbindung zwischen PNS und PAC ist weder authentisiert noch verschlüsselt, so dass hier ein breites Spektrum von Angriffen denkbar ist. Diese reichen von der Übernahme der Verbindung (Hijacking) bis zur Störung durch manipulierte oder eingestreute Pakete. Wenn auch die im VPN-Kanal getunnelten PPP-Pakete sicher sind, kann der Netzwerkverkehr durch Angriffe auf die Steuerverbindung (Denial of Service, DoS) empfindlich gestört werden.

Die Authentikation über einen MD4-Hash entspricht ebenso wie eine gewöhnliche DES-Verschlüsselung nur dem Stand der Technik vor einigen Jahren, so dass die Zukunftsaussichten von PPTP nicht rosig sind. Microsoft hat in seinem Betriebssystem Windows 2000 als Alternative eine Absicherung über IPSec implementiert. Microsoft erfüllt leider nicht den vollständigen Standard, sondern hat sich ihm lediglich angenähert, was aber – wie schon erwähnt – eigentlich eine proprietäre Lösung ist.

6.1.3 Secure Shell (SSH)

Die von der finnischen Firma SSH entwickelte »Secure Shell« ist von ihrer Grundkonzeption her eine Applikation, die unter UNIX als Ersatz für das wegen seiner Klartext-Übermittlung von Passworten und Daten als unsicher einzustufende Telnet dient. Seit der Einführung von IPSec ist die Bedeutung von SSH als VPN-Protokoll stark gesunken. Außer dem Unix-SSH sind auch zahlreiche Windows-Implementierungen verfügbar (Abb. 6.10). Analog zu einer Telnet-Verbindung nimmt ein SSH-Client eine Verbindung zu einem SSH-Server auf, allerdings werden bei der Authentikation der Verbindung und bei der nachfolgenden Datenübertragung kryptographische Verfahren aktiviert. Im Unterschied zu Telnet (Port 23) wartet der Server bei SSH an Port 22 auf die Verbindungsaufnahme eines Clients. Als Weiterentwicklung von Telnet hat sich SSH bei der Fernadministration beispielsweise von Routern oder Servern längst durchgesetzt.

Die Entwickler von SSH haben das Protokoll mit einer großen Flexibilität ausgestattet, so dass im Prinzip jedes beliebige Protokoll über eine SSH-Verbindung getunnelt werden kann. Dadurch entsteht allerdings ein gewisser Verwaltungs-Overhead, da sich das SSH-Protokoll im IP-Stack oberhalb der TCP-Ebene ansiedelt. Dennoch hat SSH wegen der einfachen Möglichkeit der Realisierung von VPNs eine große Verbreitung auch in diesem Bereich gefunden.

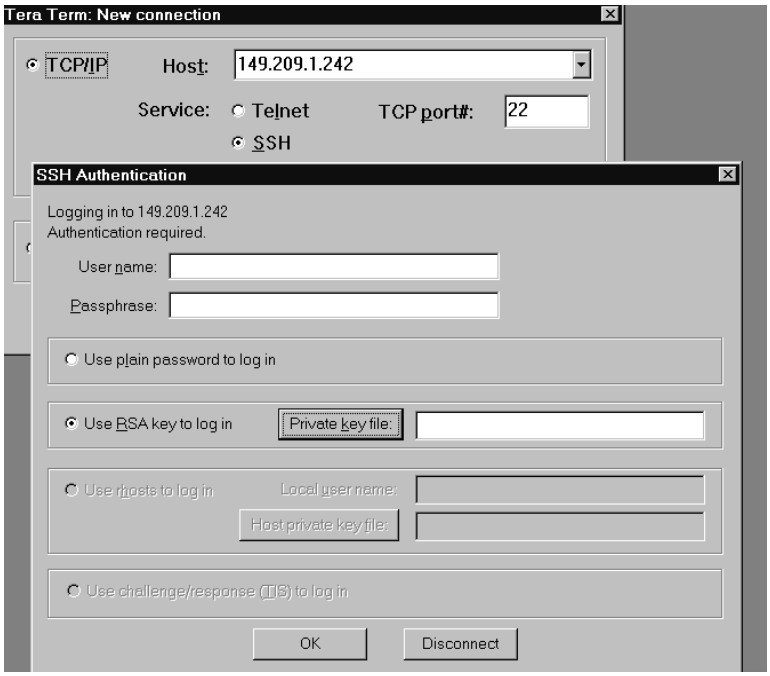

Abb. 6.10: SSH-Client unter Windows

Der Verbindungsaufbau bei SSH arbeitet dreistufig. Zunächst wird unter Benutzung des »SSH Tranport Layer Protocols« eine einseitig authentisierte Verbindung vom Client zum Server aufgebaut, bei der nur der Server seine Authentizität nachweist. Steht diese Verbindung, authentisiert sich auch der Client mittels des SSH Authentication Layer Protocols.

Damit steht ein authentisierter und verschlüsselter Tunnel zwischen Client und Server zur Verfügung, über den sich dann mittels des »SSH-Connection Protocols« beliebige andere Protokolle einspeisen lassen (Abb. 6.11).

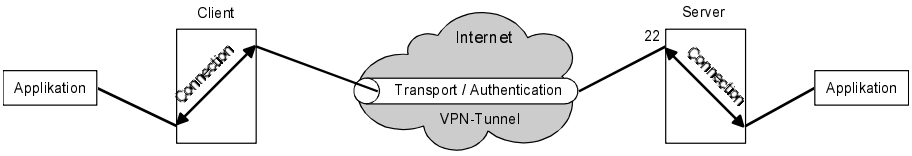

Abb. 6.11: Die Elemente einer SSH-Verbindung

Kryptographische Verfahren in SSH

In den aktuellen Implementierungen von SSH kommen folgende kryptographischen Verfahren zum Einsatz:

- Verschlüsselungsalgorithmus: Triple-DES im CBC-Modus, Blowfish (CBC), Twofish (CBC), ARCFOUR, IDEA (CBC), CAST
- Hash-Algorithmus: SHA-1 oder MD5, jeweils mit HMAC
- Authentikation: durch Diffie-Hellman oder Zertifikate nach X.509, SPKI oder PGP
- Austausch von Schlüsseln: mittels Diffie-Hellman

Die Implementierung von AES ist nur eine Frage der Zeit.

SSH-Transport Layer Protocol

Dieses Protokoll ist das allgemeine Transportmedium für SSH-Verbindungen. Es umfasst, ähnlich wie IPSec, die verschlüsselten und authentisierten Daten. Im Unterschied zum Security Parameter Index (SPI) von IPSec werden bei einer schon aufgebauten SSH-Verbindung keine Pointer auf die eingesetzten Algorithmen mehr übertragen; die gesamte »Buchführung« für die gesendeten und empfangenen Pakete müssen Client und Server unabhängig voneinander erledigen. Den Aufbau des Transport Layer Protocols zeigt Tabelle 6.3.

MAC-Header (Ethernet, Token-Ring, PPP, FDDI etc.)
IP-Header
TCP-Header
SSH-Paketlänge in Bytes
Padding-Länge in Bytes
Nutzdaten
Padding
Message Authentication Code (MAC)

Tabelle 6.3: SSH-Transport Layer Protocol

Das eigentliche SSH-Paket beginnt bei der Angabe der SSH-Paketlänge und endet mit dem MAC-Feld. Wie bei IPSec dienen die Padding-Daten dazu, feste Blocklängen für Blockverschlüsseler zu erzeugen. Verschlüsselung und Authentikation umfassen das gesamte SSH-Feld mit Ausnahme des MAC-Felds.

Zum Schutz von Replay-Attacken durch das Wiederholen bereits gesendeter Pakete wird in die Berechnung des MAC eine Sequenznummer eingebaut, die bei jedem gesendeten Paket um eins erhöht wird:

MAC = HMAC (Sequenznummer++, unverschlüsseltes SSH-Paket)

Beim Vergleich von SSH und IPSec fällt auf, dass sich die Sicherungsmechanismen von SSH nur auf das SSH-Paket beziehen. Aus der Sicht des TCP-Pakets handelt es sich um einen reinen Schutz der Nutzdaten, IP- und TCP-Header bleiben unverändert.

Das erste Byte der Nutzdaten ist eine SSH-Kennung und gibt den Typ des Pakets gemäß Tabelle 6.4 an.

SSH-Message-Number (dezimal)	Bedeutung
Transport Layer Protocol:	
1-19	Basisfunktionen des Transport-Layers (Disconnect, Ignore, Debug etc.)
20-29	Verhandlung über Algorithmen
30-49	Pakete für den Schlüsselaustausch
User Authentication Protocol:	
50-59	Basisfunktionen des User Authentication Protocol
60-79	Pakete für die User-Authentikation
Connection Protocol:	
80-89	Basisfunktionen des Connection Protocols
90-127	Nachrichten, die den einzelnen Kanälen des Tunnels zugeordnet sind
Reserviert für Client-Protokolle:	
128-191	reserviert
Erweiterungen:	
192-255	Erweiterungen

Tabelle 6.4: SSH-Message-Numbers

SSH-Authentication Protocol

Schon beim Aufbau einer SSH-Verbindung hat sich der Server authentisiert. Über das Authentication Protocol muss sich jetzt auch der Client authentisieren. Dazu sendet ihm der Server eine Liste mit Authentikations-Verfahren, die er beherrscht. Der Client kann diese Liste »nach Lust und Laune« abarbeiten. Die Bandbreite der möglichen Verfahren ist groß, sie reicht von der Angabe eines einfachen Passworts bis hin zu zertifizierten Schlüsseln.

SSH-Connection Protocol

Nach dem Aufbau eines zweiseitig authentisierten Tunnels können nun Applikationen einen Übertragungs-Kanal in diesem Tunnel eröffnen, Nutzdaten übertragen und den Kanal wieder schließen. Diese Funktionen werden über das Connection Protocol realisiert. Eine Reihe von Kanälen sind vordefiniert (Tabelle. 6.5).

Kanal-Typ	Bedeutung
session	Öffnen einer interaktiven Session
pty-req	UNIX-Pseudoterminal
x11	X-Windows
auth-agent	Client-Authentikation ohne SSH
auth-ssh1-agent	Client-Authentikation über die alte SSH-Version 1
env	Übermittlung einer UNIX-Environment-Variablen
shell	Start einer Shell
exec	Ausführung eines Programms
subsystem	Ausführung eines vordefinierten Subsystems (z. B. FTP-Transfer)
tcp-ip-forward	Tunneln von beliebigen Client-Server-Verbindungen

Tabelle 6.5: SSH-Kanäle

Zum Aufbau eines VPN bietet sich der TCP/IP-Forwarding-Kanal an, das Tunneling von UDP-Paketen ist jedoch nicht möglich. Das schränkt den Einsatzbereich von SSH etwas ein. Es gibt allerdings eine Erweiterung der aktuellen Version 2 von SSH, die alle UNIX RPC-Dienste in die Absicherung einbezieht. Damit kann ein Teil der UDP-Dienste doch noch in das SSH-Protokoll integriert werden.

6.2 Schlüsselaustausch – Methoden/Protokolle

Eine zentrale Frage bei der Implementierung von symmetrischen Verschlüsselungs-Algorithmen, wie sie auch in einem VPN zum Einsatz kommen, ist der Austausch des geheimen Schlüssels. Dieser muss beiden Seiten bekannt sein, darf aber keinesfalls in die Hände von Angreifern geraten. Eine Übertragung des Schlüssels im Klartext über das Internet scheidet damit von vorneherein aus. Beide Seiten müssen sich über einen gemeinsamen Schlüssel einig werden, ohne dass ein »Lauscher an der Wand« aus den über die Kommunikationsstrecke gesendeten Informationen einen Hinweis auf den Schlüssel enthält, der es ihm erlaubt, den Schlüssel schneller als mit einem Brute-Force-Verfahren zu ermitteln.

Da beim Aufbau einer VPN-Stecke nicht jedes Mal ein menschlicher Eingriff erfolgen kann, müssen die beteiligten Rechner automatisch nach einem vorgegebenen Key-Management-Protokoll arbeiten und mit diesem den gemeinsamen Schlüssel ermitteln und übertragen. Ist ein automatischer Austausch des Schlüssels nicht möglich, bleibt als (mühevolle) Alternative nur der manuelle Austausch. Für den automatischen Schlüsselaustausch wurden zwei grundlegend unterschiedliche Konzepte entwickelt, die jedoch beide in gängigen VPN-Implementierungen zu finden sind:

- Die Schlüsselinformation wird im IP-Paket selbst übertragen, aber natürlich so, dass ein Angreifer mit den aufgefangenen Daten den Schlüssel nicht rekonstruieren kann. Diese Form des Schlüsselaustauschs wird als geschlossenes Sicherheitssystem bezeichnet und hat den (akademischen?) Vorteil, dass sie innerhalb der IP-Ebene und somit in der zugeordneten Ebene des TCP/IP-Stack stattfindet. Für die Ebene der TCP/UDP-Protokolle oder gar der Applikationen ist dieser Schlüsselaustausch völlig transparent.
- Die Schlüsselinformation wird außerhalb der eigentlichen Kommunikation der beteiligten Applikationen übertragen. Dazu müssen eigene Hilfsapplikationen installiert werden, die das Key-Management-Protokoll »out-of-band« abwickeln und dafür sorgen, dass eine Applikation erst dann mit der Kommunikation beginnt, wenn beide Seiten den Schlüssel kennen. Der praktische Vorteil solcher Verfahren ist die mögliche Integration in eine offene Infrastruktur und die dadurch entstehenden Synergieeffekte.

Ein wichtiger Begriff beim Key-Management ist die »Perfect Forward Secrecy«. Mit ihr kann erreicht werden, dass nach dem Knacken eines Schlüssels oder Verfahrens nur zukünftige Nachrichten von einem Unbefugten entschlüsselt werden können. Die in der Vergangenheit abgesetzten verschlüsselten Nachrichten können hingegen nicht in Klartext verwandelt werden. Perfect Forward Secrecy wird durch die Verwendung kurzlebiger, von einem (Pseudo-)Zufallsgenerator erzeugter und mittels Diffie-Hellman ausgetauschter Schlüssel erreicht, die nach Gebrauch gelöscht werden. Wird ein Schlüssel oder das Verfahren der Schlüsselgenerierung später gebrochen, kennt der Angreifer die früheren Zufalls-Schlüssel nicht und sieht sich einem (erneuten) Brute-Force-Problem gegenüber gestellt.

Der VPN-Standard IPSec arbeitet mit einer Reihe von Algorithmen zum Schlüsselaustausch zusammen. Das am weitesten verbreitete Verfahren ist IKE, es sind aber auch Implementierungen mit SKIP oder manuellem Schlüsselaustausch vorhanden.

6.2.1 Pre-Shared key

Von einem Pre-Shared Key wird gesprochen, wenn der gemeinsame Schlüssel nicht erst beim Aufbau der Verbindung ermittelt und ausgetauscht wird, sondern sich bereits auf den beteiligten Rechnern befindet:

- Die Schlüssel wurden manuell übertragen (zum Beispiel per Diskette oder verschlüsselter E-Mail). Diese Methode wird auch als »Manual keying« bezeichnet.
- Die Schlüssel sind in die Geräte fest eingebaut (zum Beispiel ins EPROM).

Pre-Shared Keys haben gravierende Nachteile. Zum einen ist es bei einer größer werdenden VPN-Infrastruktur kaum noch möglich, die benötigten Schlüssel zu verteilen und einzuspielen. Sieht man einmal von der äußerst leichtsinnigen Vari-

ante ab, allen beteiligten Geräten den selben Schlüssel für die gesamte Kommunikation untereinander mitzugeben, lassen sich die bei n VPN-Knoten erforderlichen Schlüssel K zu

$K = n(n-1)/2$

berechnen. Eine Firma mit nur 10 Standorten, bei der jeder Standort mit den anderen über ein VPN kommunizieren soll, benötigt schon 45 verschiedene Schlüssel. Auf jedem der 10 Gateways ins Internet müssen die benötigten 9 Schlüssel gepflegt und gegebenenfalls ersetzt werden. Bei 50 Standorten sind schon 1225 Schlüssel erforderlich und auf jedem der 50 Gateways befinden sich 2450 Schlüssel, was von keinem Administrator mehr zu verwalten ist.

Ein weiterer Nachteil von Pre-Shared Keys ist der nur relativ seltene Schlüsselwechsel. Moderne Verschlüsselungs-Verfahren wechseln ihren Schlüssel oft schon nach Minuten (oder einigen 100 KByte), um selbst bei einem erfolgreichen Angriff auf einen der Schlüssel eine Entschlüsselung der VPN-Daten zu verhindern. Ohne besondere Maßnahmen ist ein schneller Schlüsselwechsel mit dem Pre-Shared-Key-Verfahren nicht möglich.

Pre-Shared-Key-Verfahren kommen daher nur in kleinen VPN-Strukturen zum Einsatz, oder wenn aus organisatorischen oder technischen Gründen keine andere Lösung möglich ist. Ein klassischer Einsatzfall für Pre-Shared Keys sind Geräte unterschiedlicher Hersteller, die zwar vom eingesetzten Protokoll her, nicht aber vom Verfahren des Schlüsselaustauschs her zueinander kompatibel sind.

6.2.2 Simple Key Management for Internet Protocols (SKIP)

SKIP ist ein von Sun Microsystems entwickeltes Verfahren, das den gerade aktuellen Schlüssel im Paket selbst verschickt. Dieser Schlüssel wird deshalb auch Paketschlüssel genannt. Theoretisch könnte jedes Paket mit einem anderen Schlüssel bearbeitet werden, so dass ein Knacken von SKIP aussichtslos erscheint. Der Paketschlüssel darf natürlich nicht im Klartext übertragen werden. Er wird mit einem zweiten Schlüssel (Master Key) verschlüsselt, der aber niemals über die Leitung gesendet wird. Eine Folge von Master Keys wird aus einem geheimen »Superschlüssel« berechnet, der mittels Diffie-Hellman oder aber nach einem Pre-Shared-Key-Verfahren zu den beiden Partnern übertragen werden kann.

Die Länge der IP-Pakete vergrößert sich drastisch, da zusätzlich der verschlüsselte Paketschlüssel mit übertragen werden muss und, falls der Verschlüsselungs-Algorithmus eine bestimmte Blocklänge voraussetzt, ein Padding-Feld vorhanden sein muss. Durch diese Maßnahmen kann eine Fragmentierung der IP-Pakete nötig werden, was bei korrekter Konfiguration der Netzwerkinterfaces aber keine Probleme bereiten sollte.

Bei einem n:m-VPN müssen an der gesicherten Kommunikation mehr als zwei Partner teilnehmen. Unter SKIP ist deshalb die Angabe einer Namespace-ID (NSID) möglich, die einen Namensraum mit allen beteiligten Rechnern angibt. Alle Knoten mit gleicher NSID können ohne Einschränkung gleichgewichtig die SKIP-Pakete bearbeiten.

Kryptographische Verfahren im SKIP-Protokoll

SKIP macht keine Vorgaben oder Annahmen über die zur Authentikation bzw. Verschlüsselung der Pakete eingesetzten Algorithmen, so dass im Prinzip der Einbau beliebiger Verfahren möglich ist.

SKIP im Detail

SKIP ist von seiner Definition her flexibel, der Einsatz in IPv4 und IPv6 ist möglich. Es arbeitet mit IPSec zusammen, so dass die dort definierten Header AH und ESP genutzt werden können. Durch das Mitsenden des Schlüssels in jedem Paket vereinfacht sich die Verarbeitung in den beteiligten Hosts und Gateways ganz enorm. Dabei ist der Paketschlüssel Kp nicht direkt der für die Authentikation oder Verschlüsselung genutzte Schlüssel. Aus Kp lassen sich nach einem in SKIP definierten Verfahren zwei Schlüssel ableiten, von denen einer zur Authentikation, der andere zur Verschlüsselung genutzt wird.

Abbildung 6.12 zeigt die Verwandlung eines Pakets vom Typ IPv4 in ein SKIP-Paket.

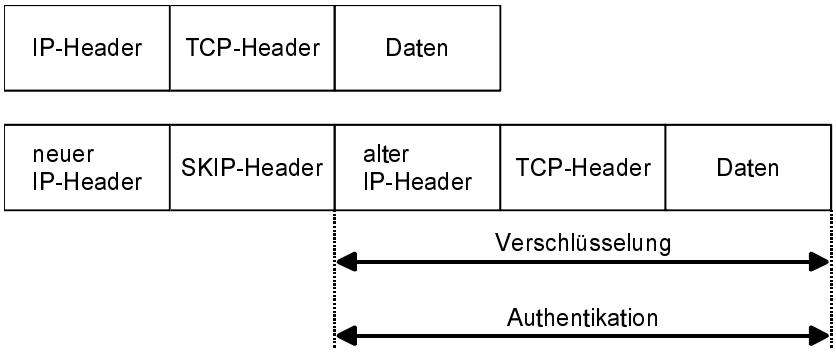

Abb. 6.12: SKIP unter IPv4

Da das gesamte ursprüngliche IP-Paket verschlüsselt wird, arbeitet SKIP immer im Tunnelmodus. Der Transportmodus kann aber durch die Umsetzung

neuer IP-Header = alter IP-Header (Modifikation von Länge und Prüfsumme)

leicht emuliert werden. Der SKIP-Header selbst hat den in Abb. 6.13 angegebenen Aufbau:

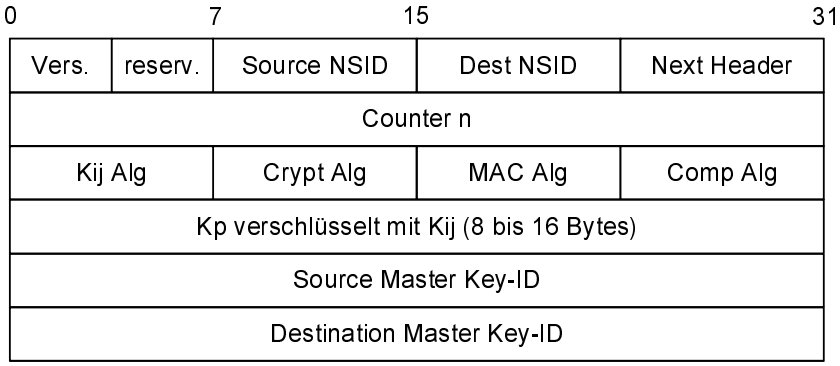

Abb. 6.13: Der SKIP-Header

Dabei haben die einzelnen Felder folgende Bedeutung:

- Die Version gibt die Versionsnummer des SKIP-Protokolls an.
- Die reservierten Bits werden zur Zeit alle auf Null gesetzt.
- Die NSID (Namespace-ID) wird eingesetzt, wenn der »Superschlüssel« mehr als nur zwei Netzwerkknoten bekannt ist. Dann bezieht sich der jeweilige Master Key nicht mehr nur auf bestimmte Rechner, sondern auf einen »Namensraum« von Rechnern. Dieser Namensraum wird durch die NSID spezifiziert. Falls eine oder beide NSID nicht angegeben sind, sind die Werte zu Null gesetzt.
- Das »Next Header«-Feld gibt – wie üblich – den Typ des nächsten Headerfelds an, beispielsweise AH oder ESP.
- Der Zähler n entspricht den Sequenznummern von AH oder ESP und dient der Bekämpfung von Replay-Attacken.
- Die nächsten Felder spezifizieren die eingesetzten Algorithmen zur Ermittlung der Master Keys Kij, zur Verschlüsselung und Authentikation (MAC) sowie zur Komprimierung.
- Es folgt der Paketschlüssel Kp, verschlüsselt mit dem Master Key Kij.
- Die letzten Felder (Master-Key-ID) werden nur benötigt, wenn die dazugehörige NSID ungleich Null ist, wenn also theoretisch mehr als zwei Partner in die Kommunikation eingebunden sein können. Dann steht in diesen Feldern ein Verweis auf den endgültigen Empfänger beziehungsweise den tatsächlichen Sender.

Der SKIP-Header hat den IP-Protokolltyp 57.

Durch die Kombination der MSID und NSID ist es möglich, auf dem Weg eines Pakets bestimmte Netzwerkknoten (Gateways, Firewalls etc.) in die Überprüfung der Pakete einzubinden, ohne dass diese die wirkliche Absenderadresse beziehungsweise die endgültige Zieladresse sind. Sie müssen lediglich zum gleichen Namensraum wie der Absender beziehungsweise Empfänger gehören.

Versand von SKIP-Paketen

Beim Versand eines SKIP-Paketes sind folgende Operationen erforderlich:

- Der aktuelle Master Key Kij wird aus dem beiden Seiten per Diffie-Hellman (oder auch durch manuelle Übertragung) bekannten »Superschlüssel« und einem Inkrement berechnet.
- Der aktuelle Paketschlüssel Kp wird beispielsweise mit einem Zufallsgenerator ermittelt.
- Aus Kp werden die Teilschlüssel für Authentikation und Verschlüsselung berechnet.
- Das Paket wird mit diesen Schlüsseln authentisiert und verschlüsselt.
- Kp wird mit dem gerade aktuellen Master Key verschlüsselt.
- Der neue Header wird gebildet und das Paket wird versendet.

Dabei sollte der Paketschlüssel Kp häufig geändert werden, zum Beispiel alle 5 Minuten oder nach 100 KBytes Daten. Der Master Key kann etwa jede Stunde geändert werden, abhängig vom Geheimhaltungsgrad und Verkehrsaufkommen. Durch das Inkrement wird gewährleistet, dass niemals derselbe Master Key ein zweites Mal eingesetzt wird.

Empfang von SKIP-Paketen

Der Empfänger verfährt folgendermaßen:

- Der aktuelle Master Key wird berechnet
- Der aktuelle Paketschlüssel Kp wird mit dem Master Key entschlüsselt
- Aus Kp werden die Teilschlüssel für Authentikation und Verschlüsselung berechnet
- Das Paket wird entschlüsselt und authentisiert
- Die Nutzdaten werden den höheren Schichten im Netzwerkstack zugeführt

Zusammenarbeit von SKIP und IPSec

Da SKIP neben dem reinen Schlüsselaustausch Möglichkeiten zur Authentikation und Verschlüsselung bietet, ist eine Integration von SKIP und IPSec kein großes Problem. Dabei steht der SKIP-Header jeweils vor den AH- bzw. ESP-Headern. Die AH- und ESP-Header selbst bleiben unverändert. Mit SKIP sinkt der Verwaltungs-Overhead für IPSec. Die Filterfunktionen SA brauchen weniger komplex zu sein, da die Authentikations- und Verschlüsselungsfunktionen ja bereits in SKIP enthalten sind.

Bewertung von SKIP

Beim Einsatz von SKIP ist die Bedrohung durch Hacker gering. Das Knacken eines einzelnen Paketschlüssels Kp bringt so gut wie nichts, und die Berechnung des Master Keys aus einer Reihe entschlüsselter Kp ist praktisch ausgeschlossen. Ein

Problem könnte der mögliche Diebstahl des »Superschlüssels« sein, der allerdings nur auf sehr wenigen Netzwerkknoten abgelegt ist und niemals über das Netz übertragen wird – auch nicht in verschlüsselter Form.

Brute-Force-Attacken sind durch den raschen Schlüsselwechsel praktisch unmöglich. Auch ein Angriff auf die Algorithmen hat nur eine begrenzte Wirkung, es sein denn, der gewählte Algorithmus ist so schwach, dass eine Echtzeit-Entschlüsselung möglich ist.

6.2.3 Internet Key Exchange (IKE)

IKE ist ein Verfahren, das auf den beiden Standards »Internet Security Association Key Management Protocol« (ISAKMP) und »Oakley« basiert. IKE, ISAKMP und Oakley wurden von der Internet Egeneering Task Force (IETF) definiert (siehe RFC 2408, 2409 und 2412 unter www.ietf.org).

ISAKMP stellt einen Protokoll-Rahmen für einen Schlüsselaustausch in zwei Phasen zur Verfügung. Oakley ist ein Perfect-Forward-Secrecy-Ansatz zum Austausch von Schlüsseln auf Basis von Diffie-Hellman. IKE ist das Standardverfahren zum Schlüsselaustausch bei IPSec.

In der ersten Phase wird ein gesicherter und authentisierter Kanal zwischen den beiden Partnern aufgebaut. Dabei besteht die Möglichkeit, alternativ ein besonders schnelles (Aggressive Mode) oder ein besonders sicheres (Main Mode) Verfahren einzusetzen. Im Main Mode werden zunächst Pre-Authentikations-Token – genannt Cookies – ausgetauscht. Diese Datenstrukturen sind kurze Hash-Werte und benötigen zu ihrer Generierung und Versendung kaum Ressourcen. Dabei muss das Cookie zunächst vom Client angefordert und anschließend dessen Empfang explizit bestätigt werden. So können Denial-of-Service-Angriffe durch die bloße Wiederholung von Anmeldeversuchen (Flooding) bekämpft werden. Danach werden über Diffie-Hellman Schlüssel ausgetauscht, die über Zertifikate abgesichert werden können. Anschließend einigen sich die Partner über ein gemeinsames Verfahren zur Verschlüsselung. Diese Festlegung wird – wie bei IPSec – Security Association (SA) genannt. Der schnellere Aggressive Mode überspringt den Austausch der Cookies, die Verbindungsaufnahme wird dadurch anfälliger gegen Angriffe.

Zwischen der ersten und der zweiten Phase kann ein Zwischenschritt eingebaut werden, der den Betrieb von IKE zusammen mit einer Gruppe von Netzwerkknoten (n:m-VPN) ermöglicht. Im »New Group Mode« kann eine Diffie-Hellman-Gruppe gebildet werden, deren Mitglieder sich dann untereinander verständigen können.

In der Phase 2 werden weitere Security Associations (SA) ausgehandelt, auf die IPSec dann unmittelbar zugreifen kann.

Kryptographische Verfahren in IPSec

IKE unterstützt eine große Anzahl kryptographischer Verfahren:

- Verschlüsselungsalgorithmus: DES, Triple-DES, IDEA, Blowfish, RC5, CAST (alle im CBC-Modus), ECP, EC2N, AES
- Hash-Algorithmus: Tiger, SHA-1 oder MD5
- Authentikation: durch DSS-Signaturen (ElGamal), RSA-Signaturen mit X.509-Zertifikaten, RSA ohne Zertifikate, Pre-Shared Keys
- Austausch von Session-Keys: mittels Diffie-Hellman

IKE im Detail

Da IKE ein sehr flexibles Protokoll ist, haben die hin und her geschickten Pakete einen komplexen Aufbau. Deshalb wird an dieser Stelle auf die Wiedergabe des Aufbaus der kompletten Pakete verzichtet. Statt dessen wird eine schematische Darstellung der einzelnen Kommunikationsschritte gegeben.

Zunächst wird (optional) vom Initiator der Verbindung ein Cookie erzeugt und gesendet. Dieser besteht im wesentlichen aus einem Hashwert, der aus einer geheimen Zahl, den IP-Adressen sowie den Portnummern der beiden Partner generiert wird. Der Empfänger des Cookies, Responder genannt, prüft, ob er schon bestehende Anforderungen des Initiators hat (Flooding?) und antwortet mit einem ähnlich gearteten Hash. Das Austauschen der Cookies hat keinerlei kryptographische Funktion, es dient nur dazu, Ressourcen beim Responder zu schonen.

Phase 1 des IKE-Protokolls baut eine abgesicherte und authentisierte Verbindung zwischen Initiator und Responder auf. Je nach dem Authentikations-Mechanismus und dem gewählten Modus ergeben sich unterschiedliche Abfolgen von Paketen, die in den folgenden Tabellen leicht vereinfacht dargestellt sind.

Initiator	Richtung	Responder
ein oder mehrere Vorschläge für eine Security Association (SA)	→	
	←	Auswahl einer SA
Diffie-Hellman-Austausch	→	
	←	Diffie-Hellman-Austausch
digitale Signatur (ggf. mit Zertifikat)	→	
	←	digitale Signatur (ggf. mit Zertifikat)

Tabelle 6.6: Main-Modus mit Authentikation über digitale Signatur

Initiator	Richtung	Responder
ein oder mehrere Vorschläge für eine SA, Diffie-Hellman-Austausch	→	
	←	Auswahl einer SA, Diffie-Hellman-Austausch, digitale Signatur (ggf. mit Zertifikat)
digitale Signatur (ggf. mit Zertifikat)	→	

Tabelle 6.7: Aggressive-Modus mit Authentikation über digitale Signatur

Initiator	Richtung	Responder
einer oder mehrere Vorschläge für eine SA	→	
	←	Auswahl einer SA
Diffie-Hellman-Austausch, Zufallszahl n mit dem öffentlichen Schlüssel des Responders verschlüsselt	→	
	←	Diffie-Hellman-Austausch, Zahl n mit dem öffentlichen Schlüssel des Initiators verschlüsselt
Message Authentication Code (MAC)	→	
	←	MAC

Tabelle 6.8: Main-Modus mit Authentikation über öffentliche Schlüssel

Initiator	Richtung	Responder
ein oder mehrere Vorschläge für eine SA, Diffie-Hellman-Austausch, Zufallszahl n mit dem öffentlichen Schlüssel des Responders verschlüsselt	→	
	←	Auswahl einer SA, Diffie-Hellman-Austausch, Zahl n mit dem öffentlichen Schlüssel des Initiators verschlüsselt, MAC
MAC	→	

Tabelle 6.9: Aggressive-Modus mit Authentikation über öffentliche Schlüssel

Initiator	Richtung	Responder
ein oder mehrere Vorschläge für eine SA	→	
	←	Auswahl einer SA
Diffie-Hellman-Austausch	→	
	←	Diffie-Hellman-Austausch
MAC	→	
	←	MAC

Tabelle 6.10: Main-Modus mit Authentikation über Pre-Shared Key

Initiator	Richtung	Responder
einer oder mehrere Vorschläge für eine SA, Diffie-Hellman-Austausch	→	
	←	Auswahl einer SA, Diffie-Hellman-Austausch, MAC
MAC	→	

Tabelle 6.11: Aggressive-Modus mit Authentikation über Pre-Shared Key

Alle weiteren IKE-Operationen (New Group, Phase 2) werden über diesen sicheren Kanal abgewickelt und können daher auf rechenaufwändige Public-Key-Operationen verzichten. Wird Perfect Forward Secrecy benötigt, ist (mindestens) ein weiterer Diffie-Hellman-Schlüsselaustausch erforderlich.

Die über das Netz geschickten Pakete ähneln von ihrer Struktur denen in Phase 1, wenn alle Operationen mit öffentlichem Schlüssel ausgelassen werden.

Bewertung von IKE

IKE ist durch die Verwendung von gut bekannten und analysierten Standard-Verfahren als sicher zu bezeichnen. Ein Nachteil ist der komplizierte Ablauf der Verhandlungen zwischen den beiden Partnern, der »out-of-band« abgewickelt wird. Dennoch ist IKE der Standard der Zukunft, vor allem wegen seiner problemlosen Integration in das IPSec-Verfahren.

6.2.4 Schlüsselaustausch bei SSH

Die Secure Shell (SSH) nutzt zum Austausch der Algorithmen und Schlüssel ein proprietäres Protokoll. Als Ergebnis dieser Prozedur entsteht ein geheimer Schlüssel, der beiden Seiten bekannt ist, und ein Hash-Wert, der als Identifikation für die Session dient. Als Standard-Verfahren sind Diffie-Hellman und SHA-1 implementiert, doch auch andere Algorithmen sind denkbar.

Zunächst müssen die Algorithmen über den Austausch mit bestimmten Netzwerk-Paketen ausgehandelt werden. Tabelle 6.12 zeigt den schematischen Aufbau.

Wert	Bedeutung
SSH_MSG_KEXINIT	Identifikation aus Abb. 6.15
cookie	Zufallszahl
kex_algorithms	Liste von Algorithmen für den Schlüsselaustausch
server_host_key_algorithms	Liste von Algorithmen für den öffentlichen Schlüssel des Servers
encryption_algorithms_client_to_server	Liste von Algorithmen für die Verschlüsselung vom Client zum Server
encryption_algorithms_server_to_client	Liste von Algorithmen für die Verschlüsselung vom Server zum Client
mac_algorithms_client_to_server	Liste von Algorithmen für die Berechnung des MAC vom Client zum Server
mac_algorithms_server_to_client	Liste von Algorithmen für die Berechnung des MAC vom Server zum Client
compression_algorithms_client_to_server	Liste von Algorithmen für die Daten-Kompression vom Client zum Server
compression_algorithms_server_to_client	Liste von Algorithmen für die Daten-Kompression vom Server zum Client
languages_client_to_server	Liste von sprachabhängigen Angaben (nach RFC 1766) für die Kommunikation vom Client zum Server
languages_server_to_client	Liste von sprachabhängigen Angaben (nach RFC 1766) für die Kommunikation vom Server zum Client
first_key_packet_follows	Flagge, die angibt, ob das erste Paket für den Schlüsselaustausch im Anschluss folgt
0	für Erweiterungen reserviert

Tabelle 6.12: Aufbau der Pakete für die Ermittlung von Algorithmen

Dabei können jeweils mehrere Verfahren (mit abnehmender Priorität) angegeben werden, die durch Kommata voneinander getrennt werden müssen. Haben sich die beiden Rechner über die Verfahren geeinigt, werden geheimer Schlüssel und Hashwert ermittelt.

Bewertung des Schlüsselaustauschs bei SSH

Da der Schlüsselaustausch bei SSH fest vorgegeben ist, besteht keine Alternative zu dem Verfahren. Seine Komplexität liegt zwischen SKIP und IKE, wegen des Einbaus bekannter Verfahren ist eine hohe Sicherheit gewährleistet.