Sichere Datenübertragung über das Internet mittels IPSec

Studienarbeit

Vorgelegt von Thilo Rusche
zur Begutachtung durch Prof. Dr. Klaus Brunnstein

14. Dezember 1999

Universität Hamburg

Fachbereich Informatik

Arbeitsbereich Anwendungen der Informatik
in Geistes- und Naturwissenschaften

Inhaltsverzeichnis

Abbildungsverzeichnis

Einleitung

Das Internet-Protokoll (*Internet Protcol*, IP) ist das Basisprotokoll, auf dem alle anderen Protokolle zur Übertragung von Daten über das Internet aufbauen. Bei der Entwicklung des IP unter der Ägide des Verteidigungsministeriums der Vereinigten Staaten hatte der Aspekt der Sicherheit nicht die höchste Priorität, so daß das Protokoll eine Reihe von inhärenten Sicherheitslücken aufweist, die zu Angriffen auf die Datenübertragung genutzt werden können. Es bedarf nur eines relativ geringen Aufwandes, um den Inhalt von IP-Paketen zu lesen, alte Pakete erneut zu senden oder den Inhalt oder die Adressen solcher Pakete zu fälschen. Dies bedeutet, daß weder die Authenzität der Daten und des Absenders noch deren Vertraulichkeit gewährleistet sind.

Das rasante Wachstum des Internets, insbesondere des World Wide Web, die damit verbundene Verfügbarkeit von kostengünstiger weltweiter Kommunikation, und insbesondere der schnell wachsende Markt für Online-Handel aller Art hat in den letzten Jahren das Bedürfnis nach vertraulicher Kommunikation und der Sicherheit von an das Internet angebundenen Netzwerken vor Angriffen schlagartig erhöht.

Die Internet Engineering Task Force (IETF), ein loser Zusammenschluß von Einzelpersonen, Firmen und Gremien, die sich mit der Weiterentwicklung und Standardisierung der Internet-Technologie befassen, entwickelt daher seit Mitte der neunziger Jahre einen Satz von Protokollen, die, unter dem Begriff Internet Protocol Security (IPSec) zusammengefaßt, die Integrität, Authenzität und Vertraulichkeit des Datentransfers über das Internet auf Netzwerkebene sicherstellen sollen. Die vorliegende Arbeit dokumentiert den derzeitigen Stand der Entwicklung auf diesem Gebiet und untersucht den Grad der mittels dieser Protokolle erreichbaren Sicherheit.

Der erste Abschnitt gibt einen kurzen Überblick über das Internet-Protokoll. Dabei wird nur insoweit auf die Details der Architektur eingegangen, als es zum Verständnis der Sicherheitsprobleme des Protokolls notwendig erscheint. Diese werden anschließend anhand eines Beispiels näher erläutert.

Der zweite Abschnitt behandelt die kryptographischen Methoden, die eine sichere Da-

tenübertragung erst ermöglichen. Diese sind zum Verständnis der IPSec-Protokolle unabdingbar. Es wird eine kurze Einführung in symmetrische und asymmetrische Verschlüsselungsverfahren gegeben, und die gängigsten Algorithmen werden vorgestellt. Anschließend erfolgt ein kurzer Ausblick auf die derzeit laufenden Bemühungen um einen zukunftsfähigen Verschlüsselungsstandard.

Abschnitt drei beschreibt die Protokolle von IPSec im Detail. Nach einer Einführung in die Architektur, die sich im wesentlichen am RFC 2401[1] orientiert, folgen Darstellungen der drei wesentlichen Protokolle von IPSec. Bei der Diskussion der IPSec-Protokolle bleiben Aspekte, die nicht unmittelbar im Zusammenhang mit der Sicherheit stehen, weitgehend unberücksichtigt.

Im vierten Abschnitt werden die vorgestellten Protokolle einer kritischen Analyse unterzogen. Dies geschieht anhand verschiedener Angriffsmethoden, die in der Praxis einen Großteil der IP-basierten Sicherheitsvorfälle ausmachen, und der Untersuchung der IPSec-Protokolle hinsichtlich ihrer Wirksamkeit gegen diese Angriffe. Außerdem wird auf die Problematik eingegangen, die durch eine Verschlüsselung ohne gleichzeitige Authentifizierung entsteht.

Abschließend erfolgt die Darstellung einer Demonstration von IPSec im praktischen Einsatz. Diese wurde im Labor des Arbeitsbereichs AGN des Fachbereichs Informatik der Universität Hamburg mit der Software VPN+ durchgeführt, die zu diesem Zweck freundlicherweise von der Firma Datafellows zur Verfügung gestellt wurde. Anhand ausgewählter Angriffe wird die Wirksamkeit der IPSec-Protokolle ebenso veranschaulicht wie das verbleibende Risiko bei ihrem unsachgemäßen Einsatzes.

[1] RFC steht für "Request for Comments", ein seit 1969 etabliertes Verfahren für Austausch und Weiterentwicklung von Internettechnologie, aus dem eine Vielzahl von Standards hervorgegangen ist. RFC 2401: *A Security Architecture for the Internet Protocol*, [rfc2401]

1

Sicherheitsaspekte im Internet

Die Kommunikation über das Internet hat in den letzten Jahren einen Umfang angenommen, den bei der Entwicklung der zugrundeliegenden Technologie niemand vorausahnen konnte. Das bei jeder Art von Datenverkehr über das Internet eingesetzte Internet-Protokoll, welches Anfang der achtziger Jahre unter Federführung des amerikanischen Verteidigungsministeriums entwickelt wurde[1], weist eklatante Sicherheitslücken auf, die zu Angriffen aller Art auf die Datenübertragung ausgenutzt werden können. Zahlreiche solcher Angriffe sind dokumentiert. So führt der Jahresreport 1998 des Computer Emergency Response Team (CERT) der Carnegie-Mellon-Universität über 3700 Sicherheitsvorfälle mit Auswirkungen auf fast 19000 Sites auf [2].

Daß die mangelnde Sicherheit des Internet-Protokolls ein Problem mit weitreichenden Auswirkungen darstellt, ist erst in jüngster Zeit zunehmend in das öffentliche Bewußtsein vorgedrungen. Der sprunghafte Anstieg der Kommunikation vertraulicher Daten über das Internet, insbesondere im Zusammenhang mit dem elektronischen Handel, dem sogenannte E-Commerce, hat zu einem erhöhten Bedürfnis nach dem Schutz dieser Daten und den an das Internet angebundenen lokalen Netzwerken geführt. Unter dem Schlagwort "Virtuelle Private Netzwerke" (*Virtual Private Network*, VPN) bemühen sich immer mehr Unternehmen und andere Organisationen, die Kommunikation zwischen einzelnen Teilnetzwerken über das Internet vor möglichen Angriffen zu schützen. Eine Reihe von kommerziellen Produkten stellt hierzu verschiedene Mechanismen zur Verfügung, die jedoch schon aufgrund ihrer unterschiedlichen Aufsetzpunkte in den einzelnen Schichten der gängigen Kommunikationsarchitekturen in den wenigsten Fällen untereinander kompatibel sind. Während diese Produkte zudem nur einzelne Kommunikationsprotokolle höherer Ebenen (zum Beispiel das *Hypertext Transfer Protocol* zur Übertragung von Webseiten) oder gar nur einzelne Applikationen (wie einen Browser) schützen, bleibt das Problem des inhärent unsicheren IP-Protokolls dadurch ungelöst.

[1] Genauer für die "Defense Advanced Research Project Agency", von deren Name sich die Bezeichnung ARPANET für den Vorgänger des heutigen Internet ableitet. Die endgültige Spezifikation des IP datiert vom September 1981 und findet sich im RFC 791 ([Pos81])

[2] [Cer98]

1.1 Das Internet-Protokoll

Das Internet-Protokoll ist innerhalb der TCP/IP-Protokollsuite für die Netzwerkebene zuständig; diese Ebene ist verantwortlich für das Routing (die Wegwahl der Pakete von Absender zu Empfänger) und ermöglicht die verbindungslose Kommunikation[3]. Dazu wird der von der Transportebene kommende Datenstrom in IP-Pakete unterteilt, die in der Größe den Beschränkungen der physikalischen Schicht Rechnung tragen (Fragmentierung). Jedes dieser IP-Pakete stellt den eigentlichen Daten einen Abschnitt, Header genannt, voran, der alle notwendigen Informationen enthält, die für eine Weiterleitung des Pakets bis zu seinem Zielort notwendig sind. Dies sind insbesondere die Adressen des Absenders und des Empfängers, die jeweils in einem 32 Bit großen Feld enthalten sind. Weitere Felder des Headers enthalten unter anderem die Versionsnummer des IP selbst (wir beschränken uns bei dieser Betrachtung auf die zur Zeit verbreitete Version IPv4, die durch die Versionsnummer 4 gekennzeichnet ist), die (variable) Länge des Headers, die Länge des gesamten IP-Pakets, eine eindeutige Identifikationsnummer und ein 8 Bit großes Feld, welches das Protokoll der Transportebene spezifiziert (z.B. TCP). Den Aufbau des IP-Headers der Version 4 zeigt Abbildung 1.1.

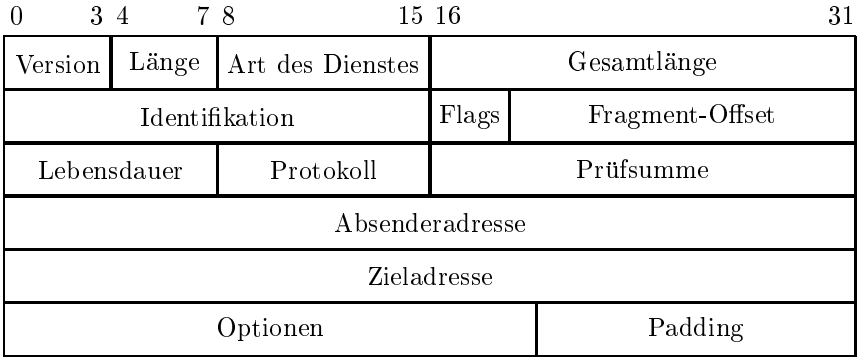

Abbildung 1.1: Aufbau des IPv4-Headers

Auf die Details des Internet-Protokolls wird in hier nur insoweit eingegangen, als es das Verständnis der Sicherheitsprobleme erfordert. Die Spezifikation des IP findet sich in [Pos81].

[3]Der Begriff "verbindungslos" bezeichnet eine Kommunikation ohne vorherige Etablierung eines festen Kommunikationskanals. Im Gegensatz dazu erfolgt zum Beispiel ein Telefongespräch "verbindungsorientiert".

1.2 Der Sicherheitsbegriff

Bevor auf die Sicherheitslücken des Internet-Protokolls näher eingegangen wird, muß zunächst der Begriff Sicherheit selbst näher erläutert werden. In [Pfl97] sind drei prinzipielle Bedrohungen der Sicherheit von Computern im allgemeinen aufgeführt, die im konkreteren Kontext der Datenübertragung unverändert bestehen bleiben: Unterbrechung, Mißbrauch und Fälschung. Daraus ergebenen sich die drei Zielsetzungen, die zum Erzielen einer sicheren Datenübertragung erreicht werden müssen:

* **Vertraulichkeit**: Eine Nachricht kann nur von ihrem rechtmäßigen Empfänger gelesen werden. Ein Beobachter der Nachricht kann aus ihrer Form keinerlei Schlüsse auf ihren Inhalt ziehen.
* **Integrität**: Die Nachricht kann nicht unbemerkt modifiziert werden. Das beinhaltet Veränderung, Löschen und Fälschen von Nachrichten inklusive der zu ihrer Übertragung notwendigen Informationen Absender oder Empfänger.
* **Verfügbarkeit**: Die zur Übermittlung von Nachrichten eingesetzten Systeme können nicht derartig gestört werden, daß die Übermittlung von Nachrichten nicht mehr möglich ist.

Wie im nächsten Abschnitt gezeigt werden wird, bietet das Internet-Protokoll in seiner jetzigen Form keinerlei Mechanismen, um das Erreichen auch nur eines dieser Sicherheitskriterien zu ermöglichen.

1.3 Angriffsmethoden auf IP-Ebene

Die Protokolle TCP[4] und IP sehen keine Möglichkeit zur Verschlüsselung der übertragenen Daten vor. Jede Person, die einen Login-Zugang mit Administratorrechten zu einem der an einer TCP/IP-Verbindung beteiligten Endsysteme oder einem dazwischenliegenden Router hat, kann den gesamten IP-Verkehr mitprotokollieren und auswerten. Zum Schutz der Verbindung vor unbefugtem Zugriff ist bisher die Implementierung von Verschlüsselungsalgorithmen in allen Anwendungen nötig, die Daten mittels TCP/IP übertragen. Zahlreiche weit verbreitete Netzwerkanwendungen tragen dem jedoch keine Rechnung. So werden beispielsweise beim Terminal-Login mittels Telnet oder der Datenübertragung per FTP alle

[4] Das *Transport Control Protocol* sorgt für die Etablierung einer echten Verbindung zwischen den Kommunikationspartnern und bildet zusammen mit IP die Grundlage des bei weitem größten Teils des gesamten Datentransfers über das Internet.

Daten, inklusive der Login-Paßwörter, im Klartext übertragen.

Auch eine Überprüfung der Authenzität des Absenders ist anhand des Internet-Protokoll nicht möglich. Viele Angriffe auf vernetzte Rechner, insbesondere zur Blockierung des Netzwerkverkehrs oder einzelner Dienste (sogenannte Denial-of-Service-Attacken), bedienen sich dieses Mangels an Authentifizierung im IP-Protokoll, um die Herkunft des Angriffs zu verschleiern und eine Rückverfolgug unmöglich zu machen. Dabei wird die Absenderadresse in den IP-Paketen des Angreifers durch eine willkürliche Adressen ersetzt (IP-Spoofing). Die gängigen Implementierungen der IP-Ebene gehen davon aus, daß die Absenderadresse eines eingehenden IP-Pakets tatsächlich mit dem System übereinstimmt, von dem das Paket abgesandt wurde. Die fehlende Authentifizierung des Absenders kann dazu ausgenutzt werden, auf IP-Adressen basierende Authentifizierungsmechanismen von Applikationen auf höheren Ebenen zu täuschen. So können die Filterfunktionen ungenügend konfigurierter Firewall-Systeme umgangen werden, indem als Absenderadresse eine vertrauenswürdige IP-Adresse des geschützten Netzwerks eingesetzt wird.

Ein Angriff mittels IP-Spoofing kann auch Auswirkungen auf den Rechner haben, dessen Adresse als Absenderadresse mißbraucht wird. Wird ein Angriff auf ein System als solcher erkannt, wird möglicherweise das angegriffene System so umkonfiguriert, daß IP-Pakete mit der Absenderadresse des scheinbaren Angreifers generell abgelehnt werden. Dadurch kann legitimierten Systemen der Zugang zu dem getäuschten System effektiv verwehrt werden.

1.4 Beispiel: Überflutung mit TCP-SYN-Paketen und IP-Spoofing

Als Beispiel eines Denial-of-Service-Angriffs, bei dem IP-Spoofing einsetzt wird, soll hier kurz eine Angriffsmethode vorgestellt werden, die unter dem Namen "TCP SYN Flooding"[5] am 19. September 1996 erstmals vom Computer Emergency Response Team (CERT) der Carnegie-Mellon-Universität bekanntgegeben wurde.

Zur Herstellung einer TCP-Verbindung zwischen einem Client und einem Server werden drei TCP-Nachrichten versandt: Der Client schickt eine SYN-Anfrage ("SYNchronize") an den Server als Aufforderung, eine Verbindung zu etablieren. Der Server antwortet mit SYN-ACK ("SYNchronize - ACKnowledge"). Schließlich bestätigt der Client mit ACK den Erhalt

[5] Siehe [CA9621]

der Serverantwort, und die Verbindung ist etabliert.

Nach Absenden des SYN-ACK-Pakets seitens des Servers besteht auf dessen Seite eine sogenannte halboffene Verbindung. Falls nach einem festgelegten Zeitraum keine Bestätigung dieser Verbindung erfolgt, wird sie wieder gelöscht. Der Server kann nur eine begrenzte Anzahl von TCP-Verbindungen gleichzeitig offenhalten, zu denen auch diese halboffenen Verbindungen gehören. Ist die Obergrenze erreicht, werden weitere Verbindungsanforderungen zurückgewiesen.

Solche halboffenen Verbindungen können leicht mittels IP-Spoofing, dem Fälschen der Absenderadresse eines IP-Pakets, erzeugt werden. Der Angreifer schickt eine TCP-SYN-Nachricht in einem Paket mit der Absenderadresse eines Systems, welches zu diesem Zeitpunkt nicht erreichbar ist, an den Server; die als Antwort gesendeten Bestätigungspakete erreichen ihren Adressaten nicht, was vom Server aufgrund der Art der Fehlerbehandlung in IP und TCP nicht bemerkt wird[6]. Mit einer großen Anzahl solcher Pakete können die TCP-basierten Dienste des angegriffenen Systems auf einfache Weise blockiert werden, in Einzelfällen kann dieser Angriff aufgrund fehlerhafter TCP/IP-Implementierungen auch das komplette System des Opfers zum Absturz bringen. Zudem ist aufgrund der gefälschten Absenderadresse die Herkunft des Angriffes nicht nachvollziehbar. Durch die Filterfunktionen von Firewall-Systemen ist kein vollständiger Schutz gegen Angriffe dieser Art gewährleistet. Der Einsatz der IPSec-Protokolle hingegen kann bei entsprechend rigoroser Konfiguration solche Angriffe unterbinden, wenngleich nur auf Kosten der Erreichbarkeit aller nicht mit IPSec arbeitenden Systeme.

[6] Für eine genauere Erklärung dieses Angriffs und seiner technischen Hintergründe auch im Zusamenhang mit IP-Spoofing siehe [Phr99].)

2

Kryptographische Grundlagen

Um die Arbeitsweise der IPSec-Protokolle verstehen und den durch ihren Einsatz erreichbaren Grad an Sicherheit beurteilen zu können, ist eine Kenntnis der gängigen Konzepte moderner Kryptographie unerläßlich. Dieses Kapitel gibt einen Überblick über die IPSec-relevanten Verschlüsselungskonzepte und die in der Praxis eingesetzten Algorithmen. Als Vertiefung sei [Sta99] empfohlen.

2.1 Symmetrische Verschlüsselung

Symmetrische Verschlüsselungsverfahren basieren auf einem gemeinsamen geheimen Schlüssel für beide Kommunikationspartner, der sowohl zur Ver- als auch zur Entschlüsselung eingesetzt wird. Die gängigen Algorithmen sind dabei schnell und effizient sowohl in Soft- als auch in Hardware implementierbar. Symmetrische Verschlüsselung kann entweder auf einen Datenstrom oder auf eine Folge von Blöcken angewandt. Bei der letzteren Methode hängt die Größe der Blöcke von dem eingesetzten Algorithmus ab. IPSec setzt ausschließlich Blockverschlüsselungsalgorithmen ein.

Da der Schlüssel bei diesem Verfahren geheim ist, wird als Nebeneffekt auch die Authenzität des Absenders gewährleistet. Das gilt allerdings nur insoweit, als daß der Empfänger einer verschlüsselten Nachricht weiß, daß nur der andere Besitzer des geheimen Schlüssels diese Nachricht verschickt haben kann. Dritten gegenüber kann er hingegen nicht beweisen, daß die Nachricht nicht von ihm selbst erzeugt wurde.

Bei der Verschlüsselung einer Menge von Datenblöcken mit demselben Schlüssel muß beachtet werden, daß der chiffrierte Text nicht durch statistische Angriffe zu entschlüsseln ist; das bedeutet, daß charakteristische Merkmale des Klartextes, wie die relative Häufigkeit bestimmter Buchstabenkombinationen oder die Frequenz der Buchstaben des Alphabets, im chiffrierten Text nicht mehr zu Tage treten dürfen. Fast alle blockorientierten Algorithmen bedienen sich zum Erreichen dieses Ziels zweier Konzepte aus Shannons Theorie der

Informationssicherheit [1]: Die *Konfusion* verändert einen Block von Information derart, daß die Ausgabe in keiner ersichtlichen Beziehung zur Eingabe steht; mittels der *Diffusion* wird erreicht, daß ein einzelnes Bit des Klartextes mehrere Bits des Chiffriertextes beeinflußt. Eine wirkungsvolle Verschlüsselung wird dabei durch das wiederholte abwechselnde Anwenden dieser beiden Transformationen in Kombination mit einem geheimen Schlüssel erreicht. Derartige Verfahren basieren meist auf einfachen binären Verknüpfungen und können relativ einfach sowohl in Soft- als auch in Hardware realisiert werden.

Dem Vorteil der Effizienz der symmetrischen Verschlüsselung stehen diverse gravierende Nachteile gegenüber:

* Wenn der Schlüssel unbemerkt in die falschen Hände gerät, kann nicht nur der gesamte Datenverkehr entschlüsselt werden, sondern es können auch unbemerkt falsche Nachrichten konstruiert werden, wenn die Authenzität des Absenders nicht durch zusätzliche Maßnahmen überprüft wird.
* Die Anzahl der benötigten Schlüssel wächst mit dem Quadrat der Anzahl der Benutzer eines Systems, die alle untereinander Nachrichten austauschen wollen.
* Die Sicherheit der symmetrischen Algorithmen ist hauptsächlich abhängig von der Länge des eingesetzten Schlüssels. Mit der kontinuierlichen Vervielfachung der Rechenleistung moderner Computersysteme können einst als sicher erachtete Schlüssellängen nicht mehr ohne Risiko eingesetzt werden.
* Die Verteilung der Schlüssel selbst stellt sowohl ein sicherheitstechnisches als auch ein logistisches Problem dar.

2.1.1 DES

Der zur Zeit am häufgsten eingesetzte symmetrische Verschlüsselungsalgorithmus ist der Data Encryption Standard (DES), der im Auftrag des US-amerikanischen Verteidigungsministeriums in den siebziger Jahren entwickelt wurde. Er operiert auf Datenblöcken von 64 Bit Größe und einem ebensolangen Schlüssel, von denen allerdings effektiv nur 56 Bit zur Verschlüsselung beitragen. Der Klartextblock wird in 16 identischen Runden jeweils in zwei Hälften aufgeteilt, deren eine Hälfte nach einem komplexen Verfahren mit dem Schlüssel kombiniert und vertauscht und dann mit der anderen Hälfte kombiniert wird. Zwischen den einzelnen Runden wird auch der Schlüssel nach einem festen Schema verändert[2].

[1] Entnommen aus [Pfl97]
[2] Für eine detaillierte Beschreibung des DES-Algorithmus siehe [Pfl97], S. 103ff

Um zu vermeiden, daß derselbe Klartext immer in identischem Chiffretext resultiert, und um einen Angriff durch Ersetzen von Teilen einer verschlüsselten Nachricht mit früher abgefangenen Teilstücken (*cut-and-paste-Angriff*) auszuschließen, wird DES meistens in einem Rückkopplungs-Modus (*feedback mode*) eingesetzt. Der populärste davon ist der CBC-Modus (*cipher block chaining*). Dabei wird jeder Klartextblock vor der Verschlüsselung mit dem Ergebnis der vorangegangenen Verschlüsselung kombiniert; der erste Block mit einem zufällig gewählten Initialisierungsvektor (IV). Diese zufällige Komponente macht es einem Angreifer unmöglich, mittels eines Angriffs mit bekanntem Klartext alle möglichen Chiffretexte eines Nachrichtenabschnitts vorauszuberechen und durch einfaches und vor allem verhältnismäßig schnelles Vergleichen an den geheimen Schlüssel zu gelangen.

Zur Entschlüsselung wird derselbe Algorithmus mit dem Schlüssel in umgekehrter Bitreihenfolge eingesetzt.

2.1.2 Triple DES

Obwohl trotz anfänglicher Zweifel keine Schwächen im Design des Algorithmus entdeckt wurden, gilt DES inzwischen nicht mehr als sicher, da die effektive Schlüssellänge von 56 Bit einem Brute-Force-Angriff mit moderner Hardware nicht standhält. So wird in [Smi98] eine Maschine diskutiert, die zu einem heutigen Preis von einer Million Dollar realisierbar und in der Lage wäre, einen 56-Bit DES-Schlüssel in durchschnittlich dreieinhalb Stunden zu finden[3]. In der Praxis wurde ein solcher Schlüssel durch den Einsatz zahlreicher vernetzter Rechner, die sich jeweils einem Teilabschnitt des Schlüssels widmeten, in unter 23 Stunden geknackt[4].

In der Praxis kommt daher zunehmend Triple-DES zum Einsatz. Diese Variante erhöht die Schlüssellänge von DES, ohne den Algorithmus selbst zu ändern, was den Umstieg besonders bei der Implementierung in Hardware erleichtert.

Bei Triple-DES kommen drei Schlüssel K_1, K_2 und K_3 zum Einsatz. Die Verschlüsselung erfolgt durch Verschlüsselung mit K_1, Entschlüsselung mit K_2 und erneute Verschlüsselung mit K_3, also durch die Funktion $E_{K_3}(D_{K_2}(E_{K_1}(P)))$ mit einer Gesamtschlüssellänge von 156 Bit. Die Entschlüsselung erfolgt in umgekehrter Reihenfolge. In der Variante Triple-DES mit zwei Schlüsseln wird K_1 gleich K_3 gesetzt. Mittels einer spezieller Angriffsmethode, die eine

[3] [Smi98], S. 66f
[4] Siehe [Zdn99a]

hohe Anforderung an die eingesetzte Hardware stellt, reduziert dich die effektive Stärke des Triple-DES-Schlüssels auf derzeit noch als sicher geltende 112 Bit.

Während zur Zeit noch die Implementierung von DES im CBC-Modus als Standard in IPSec vorgeschrieben ist, geht die Diskussion innerhalb der IPSec-Arbeitsgruppe der IETF dahin, diesen in der nächsten Version durch Triple-DES zu ersetzen.

2.1.3 Der Standard der Zukunft: AES

Das amerikanische *National Institute of Standards and Technology* (NIST) initiierte 1998 die Entwicklung eines Nachfolges des bis heute weit verbreiteten DES. Diese Entwicklung wird als notwendig erachtet, da die Sicherheit von DES schon zum jetzigen Zeitpunkt nicht mehr für alle Einsatzgebiete hoch genug ist und die Performance des ungleich sichereren Triple-DES zu wünschen übrigläßt. In Anerkennung der teilweise hohen Investitionen, die in die Kryptogrphie-Infrastruktur mit DES getätigt wurden, soll ein neuer Algorithmus standardisiert werden, der durch seine Zukunftssicherheit und Effizienz ein Umsteigen von den bisher eingesetzten Algorithmen auch wirtschaftlich rechtfertigt. Der ausgewählte Standard wird in seinen Details offengelegt und frei von Patenten sein, so daß seiner weltweiten unbeschränkten Nutzung nichts im Weg steht.

Der letztendlich im Rahmen von AES eingesetzte Algorithmus wird aus einem oder Teilen von mehreren Algorithmen bestehen, die im Rahmen eines öffentlichen Ausschreibungs- und Evaluierungsprozesses von Einzelpersonen und Organisationen beim NIST eingereicht werden. Alle eingereichten Algorithmen müssen (mindestens) eine Blockgröße von 128 Bit und Schlüssellängen von 128, 192 und 256 Bit unterstützen. Die genauen Anforderungen an die Kandidaten umfassen unter anderem die Spezifikation des Algorithmus selbst, Beispielimplementierungen in ANSI-C und eine Abschätzung der Effizienz in Hard- und Software[5].

Das Auswahlverfahren befindet sich derzeit in der zweiten Runde, bei der von den ursprünglich eingereichten 15 Algorithmen fünf zur weiteren Analyse ausgewählt wurden[6]. Die Konferenz zur Auswahl des oder der endgültigen Algorithmen wird im April 2000 stattfinden, die Fertigstellung des Standards wird für Mitte 2001 anvisiert.

In den IPSec-RFCs findet sich kein Bezug zu AES, da dessen Entwicklung erst kurz vor

[5] Die Ausschreibung mit allen einzureichenden Details findet sich unter http://csrc.nist.gov/encryption/aes.

[6] MARS, RC6, Rijndael, Serpent und Twofish.

der letzten Version dieser Dokumente angestossen wurde. Nach Auskunft von Steve Bellovin und anderen Mitgliedern der IPSec-Arbeitsgruppe der IETF wird die Übernahme von AES erst nach dessen Standardisierung diskutiert werden. Es ist jedoch zu erwarten, das AES DES als obligatorischen Algorithmus ablösen wird.

2.2 Public-Key-Verschlüsselung

Das Prinzip der Public-Key-Kryptographie wurde Mitte der siebziger Jahr fast gleichzeitig von drei Gruppen entwickelt, die unabhängig voneinander unterschiedliche Verfahren entwickelten, denen dieselben Prinzipien zugrundeliegen: Die Verschlüsselung erfolgt mittels einer mathematischen Funktion anstelle der Kombination von Substitution und Permutation, wie sie in der konventionellen Kryptographie zum Einsatz kommt, und es existiert jeweils ein Schlüssel für die Ver- beziehungsweise Entschlüsselung.

Public-Key-Verfahren lösen zwei schwerwiegende Probleme konventioneller Kryptographie mittels eines gemeinsamen Schlüssels: sie reduzieren die Komplexität der Schlüsselverteilung erheblich, und sie leisten unter bestimmten Voraussetzungen eine eindeutige Authentifizierung des Absenders einer Nachricht. Andererseits werfen sie aber ihrerseits neue Probleme auf, die insbesondere auf die Rechenintensität der zugrundeliegenden mathematischen Funktionen zurückzuführen sind. Daher sind Pulic-Key-Verfahren nicht nur wenig geeignet für die Verschlüsselung von großen Datenmengen, sondern bieten allein durch ihren Einsatz eine potentielle Angriffsfläche durch Denial-of-Service-Angriffe. Aus diesen Gründen ist "die Beschränkung der Public-Key-Kryptographie auf Schlüsselmanagement und Unterschriftsanwendungen [...] fast universell akzeptiert"[7].

Bei Public-Key-Verschlüsselungsverfahren kommen zwei verschiedene Schlüssel zum Einsatz: ein öffentlicher für die Verschlüsselung und ein geheimer (privater) für die Entschlüsselung. Das Verfahren wird daher auch als asymmetrische oder Private-Key-Verschlüsselung bezeichnet. Einige Algorithmen wie der bei IPSec vorgeschrieben RSA-Algorithmus weisen die zusätzliche Eigenschaft auf, daß die Schlüssel austauschbar sind; was mit dem einen verschlüsselt wird, kann (ausschließlich) mit dem anderen entschlüsselt werden. Dies ermöglicht den Einsatz des Verfahrens für die Erstellung einer digitalen Signatur.

[7] [Dif88]

2.2.1 RSA

Das Haupmerkmal asymmetrischer Verschlüsselungsalgorithmen besteht darin, daß es rechnerisch nahezu unmöglich ist, den geheimen Schlüssel aus dem öffentlichen zu bestimmen[8]. Der RSA-Algorithmus (benannt nach seinen Erfindern Rivest, Shamir und Adleman) basiert dabei auf der Schwierigkeit der Faktorisierung großer Zahlen, einem wohlbekannten Problem der Zahlentheorie. Es ist eine triviale Aufgabe, das Produkt zweier großer Primzahlen zu berechnen. Anderseits ist es unmöglich, aus dem resultierenden Produkt die beiden Faktoren zu berechnen; die einzige Möglichkeit, diese Faktoren zu finden, besteht im systematischen Durchprobieren aller Möglichkeiten[9].

Die Verschlüsselung des Klartextes erfolgt bei RSA durch die Potenzierung des Zahlenwertes eines Klartextblocks mit dem öffentlichen Wert e modulo eines ebenfalls öffentlichen Wertes n. e und n bilden zusammen den öffentlichen Schlüssel. Die Entschlüsselung erfolgt durch Potenzierung des Chiffretextes mit einem Wert d ebenfalls modulo n; d und n bilden den geheimen Schlüssel. Dabei ist der Modulus n das Produkt zweier im Zuge der Schlüsselgenerierung gewählter großer Primzahlen p und q, aus denen in Verbindung mit dem ebenfalls gewählten Wert e der Wert von d ermittelt wird, der die gewünschte Eigenschaft aufweist, daß $((P)^e)^d = P \bmod n$ ist[10]. Die Primzahlen p und q werden dabei so groß gewählt (gewöhnlich zwischen 100 und 200 dezimale Stellen), daß die Faktorisierung des Produkts n (und damit die Berechnung des privaten Schlüssels) mit mathematischen Verfahren unmöglich ist.

Die Sicherheit des RSA-Algorithmus basiert in erster Linie auf der Länge von n und damit auf der Länge des eingesetzten Schlüssels, der ja aus dem Tupel (e, n) besteht. Der Aufwand zum Auffinden des privaten Schlüssels (bzw. des Wertes von d) entspricht also dem Aufwand zur Faktorisierung von n. Zum heutigen Zeitpunkt wird RSA mit einer Schlüssellänge von mindestens 1024 Bit als sicher angesehen. Die kontinuierlichen Fortschritte in der Rechenleistung moderner Computersysteme und die ebenfalls fortschreitende Verbesserung der Algorithmen zur Faktorisierung großer Zahlen lassen jedoch Zweifel hinsichtlich der Zukunftssicherheit von RSA aufkommen. Zusätzliche Sicherheitsmaßnahmen bei der Auswahl der Werte p und q werden in [Sta99] angeführt.

[8] Die Formulierung in der englischsprachigen Literatur lautet "computationally infeasible". Gemeint ist die Unmöglichkeit, eine Lösung mit mathematischen Verfahren zu finden, im Gegensatz zum systematischen Durchprobieren aller möglichen Schlüssel.

[9] Das heißt, daß das Problem der Faktorisierung großer Zahlen noch nicht einmal als NP-vollständig bekannt ist. Der schnellste bekannte Algorithmus benötigt exponentielle Zeit.

[10] Das Verfahren zur Generierung der Schlüssel ist ausführlich in [Pfl97] beschrieben.

2.2.2 Diffie-Hellmann

Der nach seinen Erfindern benannte Diffie-Hellmann-Algorithmus ist ein Public-Key-Verfahren zur Generierung eines gemeinsamen geheimen Schlüssels für den Einsatz in symmetrischen Verschlüsselungsverfahren. Es basiert auf dem mathematischen Problem der Berechnung diskreter Logarithmen in endlichen Körpern.

Die beiden Kommunikationspartner generieren jeweils eine (Pseudo-) Zufallszahl x als privaten Schlüssel, mit dem ein vorher vereinbarter Wert q modulus eines ebenfalls bekannten Wertes n potenziert wird. Die Ergebnisse werden als öffentliche Schlüssel ausgetauscht und vom jeweiligen Empfänger mit dessen privaten Schlüssel erneut modulus n potenziert. Als Ergebnis erhalten beide Partner den identischen Wert $K = (q^{x_A})^{x_B} \bmod n = (q^{x_B})^{x_A} \bmod n$. Dieser ist bei entsprechender Wahl der Werte für q und n nur anhand der öffentlich bekannten Informationen nicht zu berechnen und kann als Basis für die Generierung weiterer geheimer Schlüssel oder direkt als Schlüssel für symmetrische Kryptographie eingesetzt werden.

Ein Vorteil des Diffie-Hellmann-Algorithmus besteht in der perfekten vorwärtsgerichteten Vertraulichkeit (*perfect forward secrecy*, PFS). Durch die Möglichkeit, für jede Kommunikationssitzung und ohne Einschaltung einer weiteren Instanz einen neuen Schlüssel zu erzeugen, kann gewährleistet werden, daß selbst im Fall des Bekanntwerdens dieses Schlüssels nur die Nachrichten dieser einen Sitzung kompromittiert werden, aber weder aufgezeichnete noch zukünftige Daten gefährdet sind. Diese Eigenschaft ist bei Verfahren wie ANSI X9.17, die zum Schlüsselaustausch langlebige Key-Encryption-Schlüssel verwenden, nicht gegeben. Diffie-Hellmann weist allerdings auch eine Reihe von Schwächen auf, die aber durch die in IPSec eingesetzte Variante, das in Abschnitt 3.4.1 beschriebene Oakley-Protokoll, behoben werden.

2.3 Kryptographische Hashfunktionen

Die Integrität von Nachrichten wird bei IPSec durch Nachrichten-Authentifizierungscodes (*message authentication codes*, MAC) sichergestellt. Ein MAC wird durch die Anwendung einer Hashfunktion auf eine Nachricht und einen geheimen Schlüssel erzeugt.

Eine Hashfunktion berechnet aus einer Eingabe beliebiger Größe einen Wert fester Länge, den sogenannten Hashwert. Um für den Integritätsschutz von Nachrichten brauchbar zu sein,

muß eine Hashfunktion eine Reihe von Eigenschaften besitzen:

* Die Funktion muß unumkehrbar sein (Einwegfunktion); es darf keine Möglichkeit geben, aus einem Hashwert den Eingabetext zu ermitteln (der im Falle von kryptographischen Hashfunktionen ja auch den geheimen Schlüssel enthält).
* Der Hashwert muß sowohl auf die Veränderung als auch auf die Vertauschung einzelner Bits der Eingabe reagieren.
* Es muß unmöglich sein, eine Nachricht mit dem Hashwert einer anderen Nachricht zu berechnen (schwache Kollisionssicherheit).
* Es muß unmöglich sein, zwei beliebige Eingaben zu berechnen, die denselben Hashwert produzieren[11] (starke Kollisionssicherheit).

In der Praxis sind zwei Hashalgorithmen zur Generierung von MACs weit verbreitet: MD5 (Message Digest) und SHA-1 (Secure Hash Algorithm)[12]. Beide kommen in IPSec zum Einsatz. Während SHA-1 mit seiner Hashwertlänge von 160 Bit allgemein als sicher angesehen wird, wird die Sicherheit von MD5 (128-Bit-Hashwert) in letzter Zeit zunehmend in Zweifel gezogen, da erfolgreiche Angriffe auf Teile des Algorithmus demonstriert werden konnten[13]. In IPSec kommen beide Algorithmen aber in einer Form zum Einsatz, in der auch MD5 nicht mehr für diese Art von Angriffen anfällig ist. Diese HMAC genannte Variante kombiniert die zugrundeliegenden Hashfunktionen mit einem geheimen Schlüsssel zur Generierung kryptographischer Prüfsummen. Die Spezifikation des HMAC-Verfahrens findet sich in [rfc2104].

2.4 Digitale Signaturen

Einweg-Hashfunktionen bilden in Verbindung mit Public-Key-Kryptographie auch die Grundlage für das Prinzip der digitalen Signatur. Durch die Chiffrierung eines Hashwertes mit dem privaten Schlüssel wird die Nachricht eindeutig und unfälschbar mit dem Absender verbunden, was mittels des korrespondierenden öffentlichen Schlüssels für jeden überprüfbar ist. Mit dieser Methode kann beispielsweise RSA sowohl zur Verschlüsselung (mit dem öffentlichen Schlüssel des Empfängers) als auch zur Signatur (mit dem privaten Schlüssel des Absenders) eingesetzt werden. Digitale Signaturen sind in IPSec als eine von mehreren

[11] Diese auf den ersten Blick überflüssig erscheinende Eigenschaft ist nötig zur Verhinderung eines sogenannten "Geburtstagsangriffs", einer statistischen Angriffsmethode von erstaunlicher Effizienz. Für Details siehe [Sta99], S. 256.
[12] Zur Funktionsweise der Algorithmen siehe [Sta99].
[13] ebd.

Möglichkeiten vorgesehen, um die initialen Nachrichten im Rahmen des Internet Key Exchange zu authentifizieren.

3

Internet Protocol Security

1994 veröffentlichte das Internet Architecture Board (IAB) einen Workshop-Report über "Sicherheit in der Internet-Architektur", in dem ein generelles Bedürfnis nach mehr und besserer Sicherheit im Internet festgestellt wird[1]. Sowohl für die Ende-zu-Ende-Kommunikation über TCP/IP als auch für die eigentliche Netzwerk-Infrastruktur wird darin ein Sicherheitsdefizit konstatiert, das mit bereits existierenden Mitteln zu beheben sei. Dies wird zur Zeit durch die IETF mit der Entwicklung der IPSec-Protokolle umgesetzt. Deren herausragendes Merkmal besteht darin, daß mit der Verschlüsselung und Authentifizierung des Datenverkehrs auf IP-Ebene *alle* darauf aufsetzenden Protokolle entsprechend abgesichert werden. Das betrifft Protokolle der Transportschicht, zum Beispiel TCP, ebenso wie verbindungslose Protokolle wie UDP oder ICMP. IPSec ist damit transparent für Applikationen und erfordert keine Änderungen bestehender Software außerhalb des TCP/IP-Stacks.

Obwohl IPSec integraler Bestandteil der nächsten Version des Internet-Protokolls (IPv6) sein wird, war es ein explizites Entwurfskriterium der IETF, die Protokolle auch für bestehende Systeme mit IPv4 nutzbar zu machen. So kann IPSec sowohl durch Modifizierung der IP-Implementierungen der Betriebssysteme als auch durch Hinzufügen von Software zwischen den IP-Stack und die Netzwerktreiber ("Bump in the Stack", BITS) oder Hardware ("Bump in the Wire", BITW) implementiert werden.

IPSec gewährleistet sichere Kommunikation, indem es Systemen die Auswahl von Sicherheitsprotokollen und entsprechenden Algorithmen sowie die Generierung dazu benötigter Schlüssel ermöglicht. Für die Absicherung der Datenübertragung stehen zwei Protokolle zur Verfügung: *Authentication Header* (AH) zur Authentifizierung der Daten und des Absenders und *Encapsulating Security Payload* (ESP) zur Verschlüsselung der Daten (mit optionaler Authentifizierung). AH und ESP werden in den Abschnitten 3.2 und 3.3 im Detail beschrieben. Mit diesen beiden Protokollen soll IPSec folgende Dienste leisten:

* Zugangskontrolle

[1] [rfc1636]

* verbindungslose Integrität
* Herkunftsauthenzität
* Zurückweisung von wiederholt gesendeten Paketen
* Vertraulichkeit der Daten
* teilweise Vertraulichkeit des Datenflusses

3.1 Sicherheitsassoziationen

Das Konzept der Sicherheitsassoziation (*Security Association*, SA) ist von grundlegender Bedeutung zum Verständnis von IPSec. Sowohl der Einsatz von AH als auch von ESP findet im Rahmen einer Sicherheitsassoziation statt, und der Schlüsselaustauschs mittels IKE erfordert zunächst die die Etablierung einer eigenen Sicherheitsassoziation, unter derem Schutz die Daten für die IPSec-SA ausgetauscht werden. Das Konzept der SA wird im RFC 2401 in der Sektion 4 beschrieben.

Eine IPSec-Sicherheitsassoziation ist eine Vereinbarung über die Verfahren, die zur sicheren Kommunikation zwischen zwei Systemen angewandt werden sollen, bezüglich Protokoll, Algorithmen und eingesetzter Schlüssel. Sie ist dabei immer unidirektional, so daß für eine Ende-zu-Ende-Verbindung zwei Assoziationen benötigt werden. Die Details einer SA sind in einer nominalen Datenbank[2] (*Security Association Database*, SAD) gespeichert. Eine SA enthält genau *ein* IPSec-Protokoll (AH oder ESP), so daß zur Anwendung beider Protokolle zwei SA je Richtung benötigt werden. Sicherheitsassoziationen können darüber hinaus auf verschiedene Weise zu einem SA-Bündel kombiniert werden.

3.1.1 Transportmodus und Tunnelmodus

Sowohl AH als auch ESP können in zwei verschiedenen Modi eingesetzt werden: Transportmodus und Tunnelmodus. Im Transportmodus erstreckt sich der durch den IPSec-Header geschützte Bereich lediglich auf die Nutzlast eines IP-Pakets, also auf die Daten der Transportschicht (und im Fall von IPv6 zusätzlich auf die optionalen Header). Die einzige Modifikation der IP-Pakete besteht im Einfügen der IPSec-Header nach dem IP-Header (bei IPv6 zwischen IP-Basisheader und den optionalen Headern). Alle zum Routing benötigten

[2] Nominal bedeutet in diesem Zusammenhang, daß es sich nicht notwendigerweise um eine Datenbank im Wortsinne handeln muß, sondern daß der Zugriff auf die benötigten Daten lediglich in einer Form möglich sein muß, die zu den Zugriffsmöglichkeiten einer Datenbank äquivalent ist.

Informationen des IP-Headers liegen während der Übertragung offen.

Im Gegensatz dazu wird das zu schützende IP-Paket im Tunnelmodus vollständig durch ein weiteres IP-Paket gekapselt, welches seinerseits durch einen IPSec-Header geschützt wird. Ein solcher IP-Tunnel kann sowohl zwischen zwei Sicherheits-Gateways[3] (Proxys) als auch zwischen einem Host und einem Sicherheits-Gateway bestehen. Der Schutz der IPSec-Header erstreckt sich in diesem Fall auf das gesamte originäre IP-Paket, solange sich dieses im Tunnel befindet. Der Tunnelmodus ermöglicht Hostrechnern zweier getrennter vertrauenswürdiger Netzwerke auf effiziente Weise die sichere Kommunikation über unsichere Verbindungen wie das Internet, da die Sicherheitsfunktionen von IPSec auf einem einzelnen Rechner pro Teilnetzwerk konzentriert werden. Er reduziert zudem die Anzahl der benötigten Schlüssel und erschwert im Fall von ESP zudem eine Datenflußanalyse, indem die endgültigen Zieladressen innerhalb der Teilnetzwerke verborgen bleiben.

3.1.2 Zuordnung von Sicherheitsassoziationen

Eine Sicherheitsassoziation ist eindeutig bestimmt durch die Kombination aus IP-Zieladresse[4], dem eingesetzten Sicherheitsprotokoll (AH oder ESP) und einem numerischen Wert im IPSec-Header, dem *Security Parameter Index* (SPI). Die Datenbank, die durch dieses Tripel indiziert wird, enthält eine Reihe von Parametern, die zur Anwendung von IPSec auf ein- und ausgehende Pakete notwendig sind:

* Die **Sequenznummer** ist ein 32 Bit großes Feld, das vom Sender gesetzt werden *muß* und vom Empfänger zur Erkennung von Replay-Attacken ausgewertet werden *kann*. Die Sequenznummer wird bei der Generierung der SA mit null initialisiert und bei jeder Anwendung der SA auf ein IP-Paket inkrementiert.
* Das **Sequenznummernüberlauf-Flag** wird entsprechend dem Eintrag in der SPD behandelt (siehe 3.1.3).
* Das **Anti-Replay-Fenster** besteht aus einem 32-Bit-Zähler und einer Bitmap (oder einer äquivalenten Datenstruktur) und dient ebenfalls zur Erkennung von Replay-Attacken (Die Beschreibung des Anti-Replay-Algorithmus findet sich im Anhang 6.1).
* Die **Lebensdauer** der SA kann sowohl in übertragenen Bytes als auch in Zeiteinhei-

[3] Als Sicherheitsgateway werden Rechner bezeichnet, die den einzigen Zugang zwischen einem lokalen Netzwerk und einem öffentlichen Netz darstellen und keine andere Funktion haben als die Überwachung und Filterung des Verkehrs zwischen diesen Netzen. Der Begriff der Firewall umfaßt dagegen auch die zu diesem Zweck auf dem Sicherheitsgateway installierte Software.

[4] Zur Zeit werden nur Unicast-Adressen unterstützt.

ten oder einer Kombination von beiden angegeben werden. Zusätzlich muß festgelegt sein, ob bei Überschreiten der Lebensdauer die SA verworfen oder erneuert wird. Im letzteren Fall kann eine Untergrenze (soft limit) definiert werden, bei deren Erreichen eine neue SA verhandelt wird, die beim Erreichen der absoluten Lebensdauer (hard limit) die bisherige SA ohne Unterbrechung des Datenflusses ersetzt.

* **IPSec-Protokollmodus** (Transport- oder Tunnelmodus)
* **Path MTU** (Maximum Transfer Unit, die maximale Paketgröße): Da das Anwenden von IPSec auf IP-Pakete deren Länge erhöht, muß die an die Transportebene weitergeleitete MTU entsprechend reduziert werden. Ein Alterungs-Feld sorgt für regelmäßige Aktualisierung.

Zusätzlich müssen protokollspezifische Parameter existieren, die die konkreten Algorithmen, Schlüssel und gegebenenfalls Initialisierungswerte für die Verschlüsselung (ESP) bzw. Authentifizierung (ESP und AH) enthalten.

3.1.3 Sicherheitsrichtlinien

Die in der Security Policy Database (SPD) gespeicherten Sicherheitsrichtlinien dienen sowohl der automatischen Generierung von Sicherheitsassoziationen (nur für ausgehende IP-Pakete) als auch zur Zuordnung von IP-Paketen zu bestehenden SAs. In beiden Fällen erfolgt die Auswahl anhand von Selektoren, die teilweise der Netzwerk- und teilweise der Transportebene angehören. Die Form der Implementierung der SPD (die wie die SAD als *nominale* Datenbank bezeichnet wird) ist in den Standards nicht festgelegt, wohl aber die minimalen administrativen Funktionalitäten, die dem Systemadministrator zur Festlegung von Sicherheitsrichtlinien zur Verfügung stehen müssen. Dazu gehören die Unterscheidung zwischen IPSec-relevantem Verkehr, solchem, der unbearbeitet weitergeleitet werden soll, und Verkehr, der grundsätzlich nicht weitergeleitet wird. Um eine eindeutige Zuordnung zu gewährleisten, muß die SPD eine totale Ordnung ihrer Einträge erlauben. Zudem muß sie die Erfassung der nachfolgend beschriebenen Selektoren ermöglichen, die zur paketweisen Zuordnung von SAs notwendig sind. Laut RFC 2401 wird davon ausgegangen, daß aufgrund der wahrscheinlichen Zugehörigkeit aller Pakete einer einzelnen Verbindung der Transportebene zu einem einzigen SPD-Eintrag sowie durch den Gebrauch von Wildcards die Überschaubarkeit der SPD gewährleistet ist[5].

Die Implementierung folgender Selektoren ist zwingend vorgeschrieben:

[5] [rfc2401], S.14f

* Die **IP-Zieladresse** kann aus einer konkreten IP-Adresse, einem Intervall, einer Netzadresse und -maske oder einer Wildcard bestehen. Dadurch ist es möglich, einer Reihe von Zieladressen, zum Beispiel einem Teilnetzwerk, das über ein Security Gateway erreichbar ist, eine gemeinsame SA zuzuordnen. Im Falle eines IP-Tunnels zu einem Security Gateway unterscheidet sich die IP-Zieladresse konzeptionell von derjenigen, die das Security Gateway in Kombination mit IPSec-Protokoll und SPI als Index in die Security Association Database verwendet.
* Für die **IP-Quelladresse** gilt Entsprechendes.
* Der **Name** bindet eine Policy an einen gültigen Benutzer- oder Systemnamen. Mögliche Einträge sind zum Beispiel ein voll qualifizierter Username (DNS) oder ein Namenstupel nach X.500. Dieses Feld wird nur eingeschränkt eingesetzt und muß eine Wildcard als Eintrag erlauben[6].
* Die **Sicherheitsstufe** muß nur für Systeme implementiert werden, die ein hierarchisches Sicherheitsmodell einsetzen. Für Details siehe [Dor99], Sektion 8.
* Das **Protokoll** wird dem Protokollfeld (IPv4) bzw. dem Next-Header-Feld (IPv6) des IP-Pakets entnommen. Dieses muß *nicht* notwendigerweise das Transportprotokoll enthalten[7]. Da dieses Feld im Falle eines ESP-geschützten Pakets verschlüsselt ist, muß auch hier eine Wildcard als Eintrag möglich sein.
* Für die **Quell- und Zielports** der Transportebene gilt Entsprechendes.

3.1.4 Generierung und Auswahl von Sicherheitsassoziationen

Sicherheitsassoziationen können auf zwei Wegen etabliert werden: durch manuelle Generierung oder automatisch mit Hilfe eines standardisierten Protokolls wie dem Internet Key Exchange (IKE, siehe 3.4). Die Probleme der manuellen SA-Generierung - welche natürlich auch den Austausch von Schlüsseln beinhaltet - liegen vor allem in der Komplexität bei steigender Anzahl von beteiligten Systemen, aber auch in der möglichen Kompromittierung eines Schlüssels durch unsachgemäße Handhabung der Datenträger. Zudem ist die Lebensdauer einer manuell generierten Sicherheitsassoziation gemäß RFC 2401 unbegrenzt. Die Probleme, die sich hieraus ergeben, hängen vor allem mit dem Konzept der perfekten vorwärtsgerichteten Vertraulichkeit (*perfect forward secrecy*) zusammen, das im RFC 2409 ([rfc2409]) ausführlich diskutiert wird.

[6] [Dor99], S.75

[7] Z.B. im Fall von IPv6-Erweiterungsheadern oder Paketen, die bereits mit IPSec-Headern versehen wurden. Der RFC 2402 spricht an dieser Stelle dennoch vom „Transport Layer Protocol“.

Die automatische Etablierung von SAs mittels IKE orientiert sich in ihren Details an den Einträgen in der Security Policy Database. Dort werden auch die Lebensspannen festgelegt, nach deren Ablauf die entsprechende SA gelöscht und eventuell neu verhandelt werden muß. Auf die Etablierung von Sicherheitsassoziationen mittels IKE wird in Abschnitt 3.4 eingegangen.

Die Zuordnung einer konkreten SA zu einem IP-Paket ist abhängig von dessen Richtung. Für ausgehenden Datenverkehr wird die erste SA eingesetzt, auf die der erste SPD-Eintrag zeigt, der mit den Selektoren des IP-Pakets übereinstimmt. Zeigt dieser SPD-Eintrag auf keine SA, muß diese mittels des Internet Key Exchange generiert und der SPD-Eintrag entsprechend aktualisiert werden. Eingehenden Paketen wird die richtige SA anhand des SPI-Feldes zugeordnet, dessen Inhalt in Kombination mit der Zieladresse und dem IPSec-Protokoll als Index in die SAD dient.

3.2 Authentifizierung: Authentication Header

Der Authentication Header[8] (AH) gewährleistet die Integrität der übertragenen Daten und die Authentifizierung des Absenders eines IP-Pakets mittels einer kryptographischen Prüfsumme (*Integrity Check Value*, ICV). AH-geschützte Pakete können nicht unbemerkt zwischen Sender und Empfänger modifiziert werden. Zusätzlich ermöglicht AH dem Empfänger einen begrenzten Schutz gegen Angriffe durch wiederholtes Senden abgehörter Pakete, sogenannte Replay-Attacken.

AH ist selbst nichts anderes als ein weiterer Header innerhalb des IP-Protokolls. Daher ist ein AH-geschütztes IP-Paket selbst nur ein weiteres IP-Paket, so daß AH unabhängig vom Einsatz anderer Protokollerweiterungen wie ESP eingesetzt werden kann (z.B. geschachtelt innerhalb eines SA-Bündels, siehe 3.1). AH kann im Tunnel- wie im Transportmodus angewandt werden, und es kann andere Tunnelprotokolle wie L2TP schützen.

3.2.1 Format des Headers

Der IP-Header, hinter dem der AH eingefügt wird, erhält in seinem Next-Header-Feld den neuen Wert 51, der den folgenden Header als AH identifiziert. Den Aufbau des Headers zeigt Abbildung 3.1. Die einzelnen Felder haben folgende Bedeutung:

[8] Spezifikation in [rfc2402]

* Das Feld **Nächster Header** enthält den ursprünglichen Wert des entsprechenden Feldes im vorausgehenden IP-Header, also z.B. das Protokoll der Transportschicht oder einen IPv6-Erweiterungsheader.
* Die **Nutzlastlänge** dient zur Bestimmung der Gesamtlänge des Headers. Aus Kompatibilitätsgründen mit der Headerspezifikation von IPv6[9] enthält dieses Feld die Länge des Headers in 32-Bit-Worten minus 2.
* Das **reservierte** Feld muß auf null gesetzt sein, da es in die Berechnung der Prüfsumme einbezogen wird. Darüber hinaus hat es zur Zeit keine Bedeutung.
* Der **Security Parameter Index** ist ein willkürlicher 32-Bit-Wert, der bei der Etablierung der SA erzeugt wird und der in Verbindung mit der Zieladresse und dem IPSec-Protokoll diese SA eindeutig bestimmt (siehe 3.1.2). Der Wert null darf grundsätzlich nicht als SPI verwendet werden, die Werte von 1 bis 255 sind reserviert.
* Die **Sequenznummer** wird bei der Etablierung der SA mit null initialisiert und *kann* vom Empfänger zur Erkennung von Replay-Attacken genutzt werden. Der Sender hat darauf keinen Einfluß und *muß* dieses Feld bei jeder Anwendung der entsprechenden SA um eins inkrementieren[10].
* Das in der Länge variable Feld **Authentifzierungsdaten** enthält den Integritäts-Prüfwert, der durch die in Abschnitt 3.2.2 beschriebenen Verfahren berechnet wird. Die Feldlänge muß eine ganzzahlige Anzahl von 32-Bit-Worten sein.

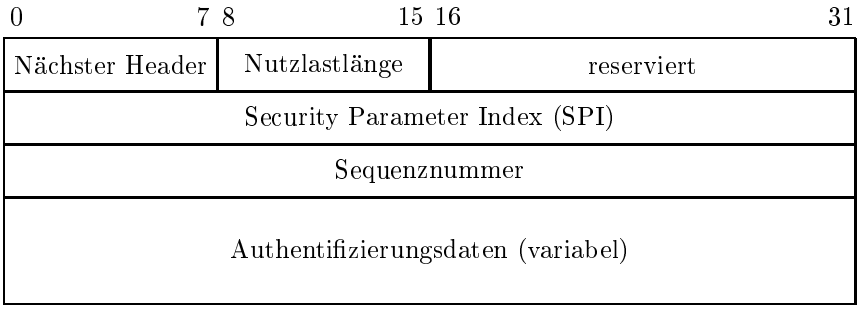

Abbildung 3.1: Der AH-Header

3.2.2 Berechnung der Authentifizierungsdaten

Der Schutz von AH erstreckt sich, im Gegensatz zum Authentifizierungsmechanismus von ESP, auch auf den äußeren IP-Header. Dabei werden die Felder, deren Inhalt auf dem

[9] Bei IPv6 wird die Länge des Headers in 64-Bit-Worten angegeben. Für Details siehe [rfc2402], Sektion 2.2

[10] Zur Erläuterung des Anti-Replay-Algorithmus siehe 6.1

Weg vom Sender zum Empfänger auf nicht berechenbare Weise verändert werden kann, für die Berechnung der Authentifizierungsdaten auf null gesetzt. Ebenso wird mit dem Authentifizierungsdaten-Feld des AH verfahren, der selbst in die Berechnung der Prüfsumme einbezogen wird. Die Einbeziehung sowohl der Absender- als auch der Zieladresse in die Prüfsumme bewirkt einen effektiven Schutz gegen Angriffe mittels Adreßfälschung (*IP spoofing*, siehe 4.2.1).

Die Berechnung der Prüfsumme erfolgt mittels eines kryptographischen Hash-Algorithmus (MAC-Algorithmus, siehe 2.3). Der RFC 2402 schreibt die Implementierung der Algorithmen HMAC-MD-96 und HMAC-SHA-1-96 vor, die beide eine Prüfsumme generieren, deren erste 96 Bit zur Authentifizierung in AH eingesetzt werden. Der durch die jeweilige Sicherheitsassoziation spezifizierte Algorithmus wird in Kombination mit dem ebenfalls in der SA festgelegten Schlüssel auf das gesamte IP-Paket (inklusive des Authentication Headers) angewandt, dessen veränderliche Felder zuvor genullt wurden. Der daraus resultierende Code wird in das Authentifizierungsdaten-Feld des AH kopiert, und mit der Wiederherstellung des Inhalts der zuvor genullten Felder ist die IPSec-Behandlung des Pakets abgeschlossen.

Der Empfänger geht prinzipiell genauso vor, um die Authentifizierungsdaten zu berechenen und den ermittelten Wert mit dem übertragenen Wert zu vergleichen. Ein mögliche Fragmentierung des IP-Pakets muß *vor* der Anwendung von IPSec erkannt und aufgelöst werden, da die IPSec-Protokolle nur auf ganze IP-Pakete angewandt werden dürfen[11]. Aus Effizienzgründen findet die (optionale) Sequenznummernprüfung ebenfalls vor der Verifizierung der Prüfsumme statt.

3.3 Verschlüsselung: Encapsulation Security Payload

Die Encapsulation Security Payload, spezifiziert im RFC 2406[12], gewährleistet die Vertraulichkeit der IP-Nutzlast durch Verschlüsselung und die Integrität durch Authentifizierung. Im Gegensatz zu AH erstreckt sich der durch ESP authentifizierte Bereich nur auf die durch ESP gekapselten Daten, *nicht* hingegen auf den äußeren IP-Header.

Der Einsatz der beiden ESP-Dienste Verschlüsselung und Authentifizierung ist jeweils optional; jedoch muß *mindestens einer* der beiden Dienste eingesetzt werden[13]. Diese Be-

[11] Im Falle von bestimmten Implementationsmodellen ist eine spezielle Behandlung der IP-Fragmentierung notwendig; siehe [rfc2402] Sektion 3.3.4 und [rfc2401] Anhang B.2

[12] [rfc2406]

[13] Siehe [rfc2406], Sektion 3.2

sonderheit in der Spezifikation von ESP stellt ein gewisses Sicherheitsrisiko dar und wird im Rahmen der Sicherheitsanalyse im Abschnitt 4.3 erörtert.

3.3.1 Format des Headers

Der IP-Header, hinter dem der Header-Teil von ESP[14] eingefügt wird, erhält in seinem Next-Header-Feld den neuen Wert 50, der den folgenden Header als ESP identifiziert. Den Aufbau des Headers zeigt Abbildung 3.1. Die einzelnen Felder haben folgende Bedeutung:

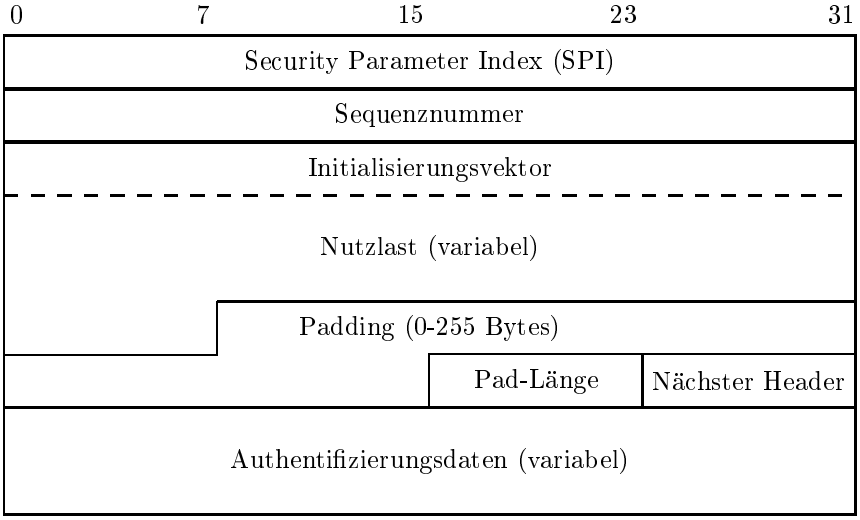

Abbildung 3.2: Der ESP-Header

* Die Felder **SPI**, **Sequenznummer** und **Nächster Header** entsprechen den gleichnamigen Feldern im AH.
* Die **Nutzlastdaten** bestehen aus dem Paket der Transportschicht (Transportmodus) oder einem vollständigen IP-Paket (Tunnelmodus). Für den Fall, daß der Verschlüsselungsalgorithmus einen Initialisierungsvektor (IV) erfordert, findet sich dieser ebenfalls im Nutzlastfeld. Die genaue Position und Länge des IV wird durch den RFC spezifiziert, in dem der Einsatz des entsprechenden Algorithmus in ESP geregelt wird. Für den für jede Implementation vorgeschriebenen Algorithmus DES-CBC[15] sind die ersten acht Byte der IV.
* Das **Padding** dient mehreren Zwecken gleichzeitig: Zum einen stellen viele blockorientierte Verschlüsselungsalgorithmen die Anforderung, daß die Länge der Eingabe ein

[14] Im Gegensatz zu AH besteht ESP aus Header und Trailer. Wenn in Zukunft vom ESP-Header die Rede ist, sind damit grundsätzlich Header *und* Trailer gemeint.
[15] Siehe [rfc2405]

ganzzahliges Vielfaches der Blockgröße ist. Dies kann durch Padding erreicht werden. Zum anderen müssen die beiden folgenden Felder, Pad-Länge und Nächster Header, rechtsbündig in einem 32-Bit-Wort ausgerichtet sein[16].

* Die **Pad-Länge** enthält die Anzahl der vorangegangenen Padding-Bytes.
* Das Feld **Authentifizierungsdaten** entspricht in seine Bedeutung dem gleichnamigen Feld in AH. Im Gegensatz zu AH ist seine Verwendung in ESP jedoch optional.

3.3.2 Verschlüsselung und Authentifizierung

ESP leistet Verschlüsselung, Authentifizierung oder beides. Zur Authentifizierung allein ist AH aufgrund des nur dort gegebenen Schutzes des äußeren IP-Headers besser geeignet, und die Verschlüsselung ohne anschließende Authentifizierung birgt Probleme, die im Abschnitt 4.3 diskutiert werden. Daher wird an dieser Stelle die Vorgehensweise bei Verschlüsselung mit anschließender Authentifizierung beschrieben.

Die Verschlüsselung der Nutzlast erfolgt mittels des Algorithmus, der bei der Verhandlung der SA festgelegt wurde. Zwingend vorgeschrieben ist die Implementierung von CBC-DES. Die Sicherheit dieses Algorithmus darf zum heutigen Zeitpunkt allerdings als nicht mehr ausreichend angesehen werden (wie im Abschnitt 2.1.1 beschrieben), da er aufgrund der unzureichenden Schlüssellänge von effektiv 56 Bit mittels Brute-Force-Attacken binnen Stundenfrist zu knacken ist. Der fortdauernden Diskussion innerhalb der IETF ist zu entnehmen, daß zukünftige Versionen der Standards die Implementation von Triple-DES vorschreiben werden. Zur Authentifizierung des Chiffriertextes sind dieselben Algorithmen wie bei AH als Standard vorgegeben (HMAC-MD5-96 und HMAC-SHA-1-96).

Die Vorgehensweise beim Einsatz von ESP ist wieder abhängig vom gewählten Modus. Im Transportmodus wird der ESP-Header hinter den ursprünglichen IP-Header eingefügt, dessen Next-Header-Feld entsprechend aktualisiert wird. Die darauffolgende Nutzlast wird durch ihre Verschlüsselung ersetzt; darauf erfolgt die Berechnung des Authentifizierungscodes, der am Ende des Pakets eingefügt wird. Diese Vorgehensweise ist effizient, verhindert aber nicht eine mögliche Verkehrsflußanalyse, da Absender- und Empfängeradresse im Klartext vorliegen.

Mit ESP im Tunnelmodus kann dagegen eine Verkehrsflußanalyse zumindest erschwert werden, indem das ursprüngliche IP-Paket komplett verschlüsselt und von einem neuen IP-

[16] Für eine detaillierte Beschreibung des Padding in ESP siehe [rfc2406], Sektion 2.4

Paket gekapselt wird. Dessen Absender- bzw. Zieladresse ist dann das Sicherheits-Gateway, das den jeweiligen Endpunkt des Tunnels bildet. Diese Vorgehensweise realisiert auf effiziente Weise eine geschützte Verbindung zweier vertrauenswürdiger Netzwerke über ein ungeschütztes Teilnetzwerk und damit ein sogenannte virtuelles privates Netzwerk (*Virtual Private Network*, VPN).

3.4 Schlüsselgenerierung: Internet Key Exchange

Bevor der Schutz von AH oder ESP auf ein IP-Paket angewandt werden kann, muß eine Sicherheitsassoziation zwischen den kommunizierenden Rechensystemen existieren. Diese kann manuell oder automatisch erzeugt werden. Mindestens die manuelle Generierung ist laut RFC 2401 als Standard zu implementieren.

Zur automatischen Erzeugung dient der Internet Key Exchange[17] (IKE). IKE ist ein hybrides Protokoll, das in seinen wesentlichen Teilen auf dem Internet Security Association Key Management Protocol (ISAKMP) einerseits und dem Oakley Key Determination Protocol anderseits basiert [18]. Das Oakley-Protokoll beschreibt Verfahren zur Generierung sogenannter "shared secrets", gemeinsamer Geheimnisse, die zur Generierung gemeinsamer Schlüssel für symmetrische Verschlüsselungsverfahren dienen, während ISAKMP die Paketformate spezifiziert, mittels derer (unter anderem) Oakley-Nachrichten ausgetauscht werden.

3.4.1 Das Oakley-Verfahren zur Schlüsselgenerierung

IKE implementiert *nicht* das vollständige Oakley-Protokoll, sondern nur eine Teilmenge davon. Wenn in den weiteren Ausführungen von Oakley die Rede ist, ist damit immer nur der von IKE übernommene Teil gemeint.

Das Oakley-Verfahren ist eine Erweiterung des in Abschnitt 2.2.2 beschriebenen Diffie-Hellmann-Algorithmus zur Generierung gemeinsamer Geheimnisse für die Erzeugung von geheimen Schlüsseln. Das Verfahren nach Diffie-Hellmann weist eine Reihe von bekannten Schwachstellen auf, die durch zusätzliche Mechanismen behoben werden.

[17] [rfc2409]

[18] In der Literatur ist daher auch inkonsistent sowohl von "ISAKMP/Oakley" als auch von "IKE" die Rede. Im folgenden wird zwischen IKE als Standard zum Schlüsselaustausch und ISAKMP als reiner Formatbeschreibung unterschieden.

Eine dieser Schwächen besteht in der Rechenintensität der Modulus-Potenzierung einer großen Zahl. Durch das Erzeugen einer großen Anzahl von Anfragen nach Diffie-Hellmann-Schlüsseln, die in Paketen mit gefälschter IP-Adresse gesendet werden, kann der angegriffene Rechner mit deren Berechnung ausgelastet und faktisch lahmgelegt werden[19]. Zur Verhinderung dieser Art von Denial-of-Service-Angriffen setzt Oakley zufällig generierte Zahlenfolgen, sogenannte Cookies ein. Der Initiator sendet eine Zufallszahl in seiner ersten Nachricht, der Empfänger bestätigt diese durch Wiederholung und schließt in seiner Antwort ein eigenes Cookie ein. Dieses muß wiederum vom Initiator bestätigt werden. Erst nach Erhalt dieser zweiten Bestätigung findet auf Seiten des Empfängers die Berechnung des Schlüssels statt. Die eingesetzten Cookies müssen dabei zwei Kriterien erfüllen: sie müssen von den beteiligten Adressen abhängig sein, um einen Angriff mit einem abgehörten gültigen Cookie und einer gefälschten Absenderadresse auszuschließen, und sie müssen durch Einbeziehung eines lokalen Geheimnisses bei ihrer Generierung fälschungssicher sein. Zudem muß die Berechnung der Cookies so effizient sein, daß sie nicht selbst für einen Denial-of-Service-Angriff mißbraucht werden kann.

Eine weitere Schwäche von Diffie-Hellmann ist die Anfälligkeit für einen Man-in-the-Middle-Angriff. Dabei täuscht der Angreifer beiden am Schlüsselaustausch beteiligten Seiten die Identität des anderen vor, was dazu führt, das beide Seiten einen sicheren Kanal zum Angreifer aufbauen, den sie für ihren legitimen Kommunikationspartner halten. Dieser kann nun alle Nachrichten entschlüsseln, neu verschlüsseln und weiterleiten, ohne daß sich die Opfer eines Angriffes bewußt sind. Dieser sehr gefährlichen Angriffsmethode wird bei Oakley mit starken Authentifizierungsmaßnahmen wie digitalen Signaturen oder asymmetrischer Verschlüsselung begegnet[20]. Ein Beispiel folgt im nächsten Abschnitt bei der Behandlung der Phase 1 des IKE-Protokolls.

3.4.2 ISAKMP und IKE

Das Internet Security Association Key Management Protocol[21] definiert kein spezielles Verfahren zum Austausch von Schlüsseln. Vielmehr ist ISAKMP ein generisches Protokoll, daß den Rahmen für den Einsatz eines beliebigen Schlüsseltausch-Algorithmus bildet. Zu diesem Zweck definiert ISAKMP eine Reihe von Nutzlast-Paketen, aus denen zusammen mit dem ISAKMP-Header die einzelnen Nachrichten eines solchen Austauschs zusammengesetzt

[19] Die Literatur spricht hier von einem "Clogging"- Angriff, was am besten noch mit "Verstopfungsangriff" zu übersetzen ist.
[20] Zur Unterscheidung zwischen starker und schwacher Authentifizierung siehe [Smi98], S.69ff
[21] [rfc2407]

werden. Den Aufbau des ISAKMP-Headers zeigt Abbildung 3.3.

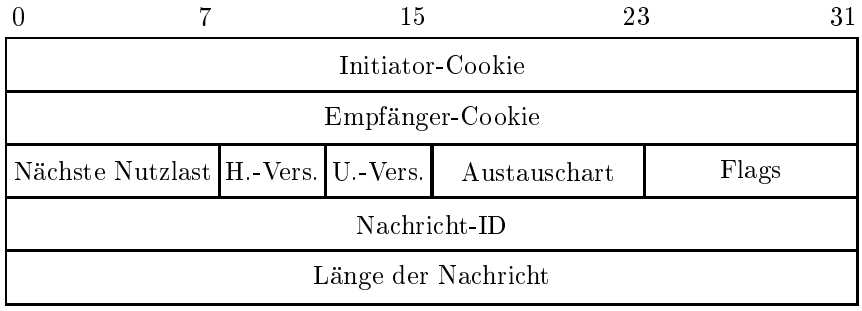

Abbildung 3.3: Der ISAKMP-Header

IKE definiert zwei verschieden Modi für den authentifizierten Austausch von Schlüsseln: Hauptmodus (*Main Mode*) und aggressiver Modus (*Aggressive Mode*). Diese entsprechen den ISAKMP-Austauscharten "Identitätsschutz" und "aggressiv". Da sich die beiden Modi im Hinblick auf den Grad ihrer Sicherheit nicht unterscheiden, sei für die Details auf [rfc2409] Abschnitt 5 verwiesen. Der generelle Unterschied besteht in der geringeren Anzahl von Nachrichten im aggressiven Modus, die auf Kosten der Flexibilität und des Identitätsschutzes des Hauptmodus erreicht wird.

Der Austausch von Schlüsseln und Algorithmen zur Etablierung von IPSec-Sicherheitsassoziationen erfolgt bei IKE in zwei Phasen. Die Phase 1 dient zur Etablierung einer IKE-Sicherheitsassoziation, in deren Schutz in Phase 2 Sicherheitsassoziationen anderer Protokolle, wie z.B. AH und ESP im Falle von IPSec, etabliert werden können. Da hier die erstmalige Kontaktaufnahme zweier einander bisher nicht bekannter Systeme eintritt und die Nachrichten noch nicht durch IPSec-Header geschützt sein können, ist an dieser Stelle die Anfälligkeit für Angriffe besonders hoch. Daher werden in Phase 1 alle oben erwähnten Mechanismen eingesetzt, um die Integrität und Authenzität der einzelnen Nachrichten und der darauf aufbauenden Sicherheitsassoziationen zu gewährleisten. Wie das im Einzelnen erreicht wird, ist am anschaulichsten an einem Beispiel nachzuvollziehen, das dem RFC 2409 entnommen wurde:

Die erste Nachricht des Initiators enthält eine Menge von Vorschlägen (*proposals*) für die IKE-SA, aus der der Empfänger eine konkrete SA auswählt und in seiner Antwort bekanntgibt. Die Vorschläge enthalten unter anderem jeweils einen Algorithmus für Verschlüsselung und Authentifizierung sowie eine von fünf in IKE definierten Diffie-Hellman-Gruppen. Der Aufbau eines solchen SA-Vorschlags kann sehr schnell komplex werden, für die Details sei auf [Sta99] verwiesen. Als Teil der Header (HDR) werden mit diesen beiden Initialnachrichten

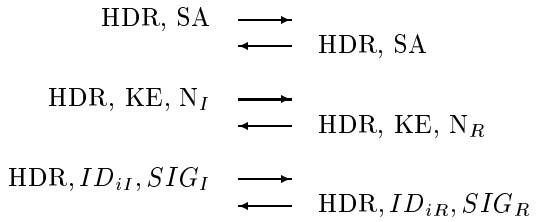

Abbildung 3.4: Nachrichten der IKE-Phase 1

Cookies ausgetauscht.

Die nächsten beiden Nachrichten dienen dem Austausch von Material zur Generierung von Schlüsseln (KE), im Kontext von IKE also den von den Teilnehmern gewählte öffentlichen Diffie-Hellmann-Werten. Zudem werden Pseudozufallszahlen ausgetauscht, sogenannte Noncen (N_I bzw. N_R), die zusammen mit dem gemeinsamen Diffie-Hellman-Geheimnis zur Generierung des gemeinsamen Schlüssels eingesetzt werden. Sie dienen außerdem zur Verhinderung von Wiederholungsangriffen. Durch Überprüfung des eigenen Cookies, das ab der dritten Nachricht (also HDR, KE, N_i) stets Teil des Headers ist, ist ein Clogging-Angriff mittels Adreßfälschung ausgeschlossen, da dieses nur dem Empfänger der zweiten Nachricht zur Verfügung steht und direkt von dessen IP-Adresse abhängig ist.

Zu diesem Zeitpunkt steht beiden Seiten alle Informationen zur Verfügung, um einen gemeinsamen Schlüssel nach Diffie-Hellman berechnen zu können. Allerdings kann bis zu diesem Zeitpunkt ein Man-in-the-Middle-Angriff nicht ausgeschlossen werden, da bisher weder die Identitäten der Teilnehmer noch die Integrität der bisher ausgetauschten Daten gewährleistet sind. Beides erfolgt mit dem Austausch der letzten beiden Nachrichten, im konkreten Fall durch das Senden eines signierten Hashwertes (SIG_I bzw. SIG_E), der über alle bisher ausgetauschten Daten unter Einbeziehung des gemeinsamen Schlüssels berechnet wird. Andere Möglichkeiten der Absender-Authentifizierung mittels vorher vereinbarter Schlüssel (preshared key), asymmetrischer Verschlüsselung oder durch Zertifikate stehen ebenfalls zur Verfügung[22].

Durch die Verkettung der einzelnen Nachrichten mittels Cookies, Noncen und Hashcodes wird auch ein Angriff mittels Verbindungsübernahme (*Session Hijacking*) unterbunden. Zudem kann die Nutzlast des dritten Nachrichtenpaares bereits mit dem gemeinsamen Schlüssel chiffriert werden, wodurch die Identität der Teilnehmer (ID_I bzw. ID_E) nach außen abge-

[22]Zu den Details der Berechnung des gemeinsamen Schlüssels und der Hashwerte für die verschiedenen Authentifizierungsmethoden siehe [rfc2409], Abschnitt 5.

schirmt wird.

Die Phase 2 des Internet Key Exchange dient zur Etablierung der eigentlichen IPSec-Sicherheitsassoziationen. Der einzige dafür vorgesehene Modus ist Quick Mode, der keinen ISAKMP-Austausch als Entsprechung hat. Die Nachrichten, die im schnellen Modus ausgetauscht werden, müssen komplett von der in Phase 1 erzeugten SA[23] geschützt werden. Die Reihenfolge der einzelnen ISAKMP-Pakete innerhalb einer Nachricht im schnellen Modus ist - im Gegensatz zu den Nachrichten der Phase 1 - insofern vorgegeben, als daß unmittelbar auf den Header ein Hash-Paket folgen muß, das den Rest der Nachricht authentifiziert, und im Anschluß daran ein SA-Paket.

Eine einzelne IKE-SA kann zur Erzeugung vieler Sicherheitsassoziationen der Phase 2 eingesetzt werden. Im schnellen Modus dient dabei das ID-Feld des ISAKMP-Headers zur Unterscheidung von Nachrichten verschiedener Phase-2-SAs innerhalb einer Phase-1-SA, die wiederum durch die Kombination der beiden Cookies eindeutig bestimmt ist.

[23] nachfolgend als IKE-SA bezeichnet

4

Sicherheitsanalyse

4.1 Sinnvolle Einsatzbereiche von IPSec

Für eine Diskussion der Sicherheit der IPSec-Protokolle bedarf es zuerst einer Abgrenzung ihrer Einsatzgebiete und einer Betrachtung der Ziele von IPSec. Es sollte deutlich sein, daß der Einsatz der IPSec-Protokolle nicht in jedem Szenario der Datenübertragung über das Internet sinnvoll ist. Einen Webserver, der ein möglichst großes Publikum erreichen soll, mit IPSec auszustatten, ist sicherlich nicht im Sinne der Erfinder. Einen ungeschützen Webserver hingegen in einem Netz zu betreiben, daß ansonsten nur über IPSec-geschützte Sicherheitsgateways zu erreichen ist, macht alle darin enthaltenen Sicherheitsmaßnahmen gegen unbefugtes Eindringen wirkungslos.

Wir beschränken uns daher in der Untersuchung der Sicherheit der IPSec-Protokolle auf den Einsatz dieser Protokolle zum Schutz der Verbindung zweier vertrauenswürdiger Netzwerke beziehungsweise Rechner über ein öffentliches TCP/IP-Netzwerk. Dabei wird davon ausgegangen, daß die Sicherheitsrichtlinien der Gateways so konfiguriert sind, daß *ausschließlich* IPSec-geschützter Datenverkehr zugelassen ist. Anders ausgedrückt muß jedes IP-Paket, daß keinen gültigen IPSec-Header aufweist, ohne weitere Bearbeitung fallengelassen werden. Ist hingegen das System so konfiguriert, daß anhand der SPD-Selektoren (siehe 3.1.3) ausgewählte IP-Pakete die Bearbeitung durch IPSec umgehen können, ist das gesamte System nicht mehr gegen die unten aufgeführten Angriffe geschützt, was den Einsatz von IPSec zumindestens fragwürdig macht.

4.2 Schutz gegen klassische Internetangriffe

Seit der explosionsartigen Ausbreitung des Internet und besonders des World Wide Web sind zahlreiche Angriffe auf an das Netz angeschlossene Rechner dokumentiert, die alle auf den inhärenten Schwächen der Protokolle TCP und IP basieren. Gegen einige dieser An-

griffsmethoden gibt es bereits Schutzmaßnahmen, die in den verbreiteten Firewall-Systemen eingesetzt werden. Gegen andere Angriffe, insbesondere solche, die auf IP-Adreßfälschung basieren, gibt es bis heute keinen wirkungsvollen Schutz.

Im folgenden wird auf die am häufigsten anzutreffenden Angriffsmethoden eingegangen und die Wirksamkeit des Schutzes vor ihnen durch den Einsatz der IPSec-Protokolle untersucht. Dabei wird weiterhin vorausgesetzt, daß diese Angriffe auf ein IPSec-geschütztes virtuelles Netz von außerhalb dieses Netzes erfolgen. Auf die Problematik eines Insider-Angriffs wird in Abschnitt 4.3 eingegangen.

4.2.1 Adreßfälschung (IP Spoofing)

Das Fälschen der Absenderadresse dient nicht nur der Verschleierung des Herkunftsortes eines Angriffs, sondern kann auch dazu eingesetzt werden, um einem Rechner innerhalb eines Netzwerkes einen vertrauenswürdigen Absender vorzutäuschen[1]. Während der Versuch, einem Gateway eine Absenderadresse des internen Netzwerks vorzutäuschen, von jeder modernen Firewall mittels Input-Filterung unterbunden werden kann, ist eine beliebig gefälschte Absenderadresse nicht ohne weiteres als solche erkennbar. Dies kann als Grundlage für eine ganze Reihe von Angriffsarten ausgenutzt werden. Nach Angabe des CERT der Carnegie-Mellon-Universität ist eine steigende Zahl von Denial-of-Service-Angriffen zu beobachten, die IP-Spoofing zur Verschleierung ihrer Herkunft einsetzen[2].

Akzeptiert ein IPSec-geschützter Rechner nur Pakete, die durch AH authentifiziert werden, wird eine gefälschte Absenderadresse sofort erkannt, da es einem Angreifer von außerhalb des Netzes nicht möglich st, einen gültigen Integritäts-Prüfwert zu generieren.

Gefälschte oder wiederholt gesendete ESP-Pakete hingegen werden bei gültigem SPI erst *nach* der Entschlüsselung ihrer Nutzlast als solche erkannt, da sich der Authentifizierungsmechanismus von ESP nicht auf den äußeren IP-Header erstreckt. Dadurch kann es ohne den Einsatz von AH zumindest zu einer Performanceverminderung durch unnötige Entschlüselungsoperationen kommen.

[1] In der Literatur wird der Begriff "IP sppoofing" häufig auch nur in dieser eingeschränkten Bedeutung verwendet. Da "spoofing" sowohl mit "Manipulation" als auch mit "Verschleierung" übersetzt werden kann, bezeichnet der Begriff hier das generelle Fälschen von IP-Adressen.

[2] Siehe [Cer97]

4.2.2 ICMP-Broadcast-Angriff (Smurfing)

Obwohl IP-Spoofing in den meisten Fällen den oben beschriebenen Zwecken dient, kann das gezielte Fälschen einer Absenderadresse auch unmittelbar für einen Angriff instrumentalisiert weden. Bei einem Angriff mittels "Smurfing"[3] wird ein ICMP-Echo-Request-Paket ("ping") mit der (gefälschten) Absenderadresse des ultimativen Opfers an eine Broadcast-Adresse eines dritten Systems gesendet. Dies hat zur Folge, daß alle mit der Broadcast-Adresse erreichbaren Rechner Antworten an den vermeintlichen Sender des Pakets verschicken, was eine Überlastung von dessen Netzwerkanbindung zur Folge haben kann.

Angriffe dieser Art können durch den Einsatz von IPSec grundsätzlich *nicht* verhindert werden. Das liegt daran, daß der Effekt dieser Angriffe bereits in der Überlastung der Datenleitungen selbst besteht, so daß auch das Ignorieren aller ICMP-Echo-Reply-Nachrichten ohne IPSec-Authentifizierung nur das interne Netzwerk schützen kann, nicht aber dessen Verbindung zu anderen Teilnetzen.

4.2.3 Abhören von Paßwörtern (Paßword Sniffing)

Das Abhören von Paßwörtern (*password sniffing*) ist eine spezielle Ausprägung des generellen Abhörens von IP-Verkehr (*packet sniffing*), für das es eine ganze Reihe von Programmen gibt. Es beruht auf der Tatsache, daß in vielen Protokollen der Transportschicht Loginnamen und Passwörter unzureichend oder gar nicht verschlüsselt übertragen werden. Da sich diese Daten zudem meist an leicht vorhersagbarer Stelle am Anfang des Datenstroms befinden, beispielsweise zu Beginn einer Telnet-Sitzung, ist es mit einem Sniffing-Programm auf einem Router ein Leichtes, in relativ kurzer Zeit eine große Menge an Name-Paßwort-Kombinationen auszuspähen und sich so unberechtigt und unbemerkt Zugang zu den betroffenen Systemen zu verschaffen.

Die Verschlüsselung mittels ESP beugt einem solchen Angriff wirkungsvoll vor. Einer der expliziten Einsatzziele von ESP ist die Verschlüsselung des gesamten Netzwerkverkehrs, unabhängig vom eingesetzten Transportschichtprotokoll, so daß auch wiederverwendbare Paßwörter ohne großes Risiko über ein öffentliches Netz übertragen werden können.

[3] Benannt nach einem der Programme, mit denen ein solcher Angriff ausgeführt wurde; siehe [CA9801].

4.2.4 Überflutung mit TCP-SYN-Paketen

Die Funktionsweise dieses Denial-of-Service-Angriffs wurde bereits in Abschnitt 1.4 beschrieben. Da auch dieser Angriff auf IP-Spoofing beruht, ist er durch die Authentifizierung aller IP-Pakete mit AH zu verhindern.

4.2.5 Verbindungsübernahme (Session Highjacking)

Die Technik der "Entführung" einer etablierten TCP-Verbindung (*session hijacking*) dient vor allem der Umgehung von Authentifizierungsmaßnahmen von Verbindungen höherer Schichten, zum Beispiel eines durch Paßwort-Überprüfung geschützten Logins mittels Telnet. Nachdem die Authentifizierung auf dieser Ebene abgeschlossen ist, die der Angreifer mittels eines Sniffers beobachtet hat, wird der Client von der Verbindung abgeschnitten, so daß der Angreifer nachfolgend dessen Identität vortäuschen kann. Dies kann zum Beispiel dadurch erreicht werden, daß dem Server TCP-Pakete mit hoher Sequenznummer in IP-Paketen mit der Adresse des Clients gesendet werden, was bei diesem zu einem nicht synchronen Zustand und anschließendem Abbruch der Verbindung führt. Der Server bemerkt diesen Abbruch nicht, und der Angreifer kann mit gefälschten IP-Adressen den abgeschnittenen Teilnehmer simulieren. Damit dieser nicht einen erneuten Verbindungsaufbau initiiert, kann er in einem vorbereitenden Angriff mittels TCP-SYN-Überflutung blockiert werden[4].

Ein Angriff dieser Art ermöglicht den Einbruch in Systeme, die mit Mechanismen wie Einmal-Paßwörtern oder Challenge-Response-Systemen geschützt werden und somit durch simples Paßwort-Sniffing nicht kompromittierbar sind.

Ist eine solche derartige Verbindung durch IPSec (AH oder ESP) authentifiziert, ist der Angreifer nicht in der Lage, Pakete mit gespoofter Adresses und gültigem IPSec-Header zu generieren. Somit ist keiner der drei Teilangriffe Verbindungsabbruch, Blockierung des Client und Täuschung des Servers durchführbar. Allein die Verschlüsselung durch ESP verhindert eine erfolgreiche Übernahme, da das Ausspähen der bestehenden Verbindung hinsichtlich der Sequenznummern unerläßliche Voraussetzung für das Gelingen des Angriff ist.

[4] Für eine ausführliche Betrachtung dieses Angriffs und der dazu benötigten Techniken siehe [Ips96].

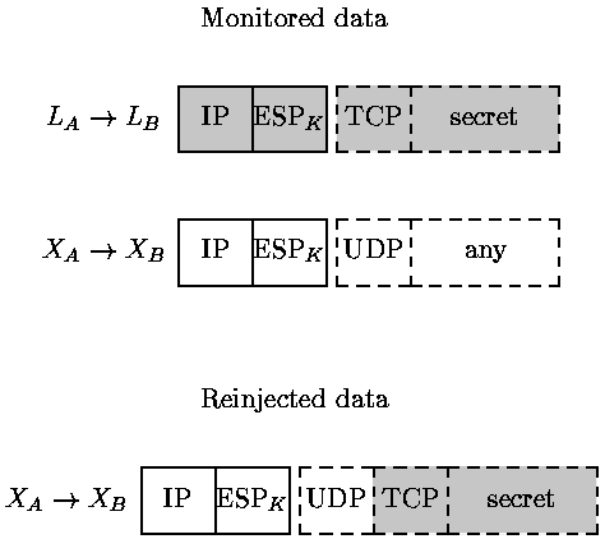

Abbildung 4.1: Cut-and-Paste-Angriff (aus [Bel96])

4.3 Insiderangriffe auf ESP

Bei den oben beschriebenen Angriffen handelt es sich ausnahmslos um Angriffe von außerhalb auf ein mit IPSec geschütztes Teilnetzwerk. Es kann jedoch auch Szenarien geben, bei denen ein Angreifer Zugang zu einer oder beiden Maschinen hat, zwischen denen legitime IPSec-Nachrichten ausgetauscht werden. Steve Bellovin beschreibt in [Bel96] ein solches Szenario. Dabei fängt der Angreifer eine ESP-geschützte Nachricht von einem Benutzer auf einer Maschine A an einen Benutzer auf einer Maschine B ab. Die verschlüsselte Nutzlast kapselt er in ein UDP-Paket mit gültigem ESP-Header, das er auf seinem Account auf Maschine A erzeugt und an seinen eigenen Account auf Maschine B addressiert (siehe Abbildung 4.1). Die IPSec-Implementierung auf B wird dieses Paket ohne Beanstandung entschlüsseln. Daß dabei möglicherweise der erste Block des TCP-Headers der kopierten Nachricht verloren geht, spielt keine Rolle, solange der eigentliche Inhalt aufgrund der selbstheilenden Eigenschaften des Cipher Block Chaining lesbar bleibt.

Vorraussetzung für diesen Angriff ist die Anwendung von ESP sowohl ohne die (laut RFC 2401 optionale) Authentifizierung als auch ohne den (ebenfalls optionalen) Einsatz des Anti-Replay-Mechanismus.

Bellovin beschreibt in seiner Analyse noch weitere mögliche Insiderangriffe auf die IPSec-Protokolle. Dabei handelt es sich aber ausschließlich um Angriffsmethoden, die entweder auf ungenügender Umsetzung der Standards (zum Beispiel der Fragmentierungsangriff) oder auf

inkonsequentem Einsatz der vorhandenen Schutzmechanismen (ESP ohne Authentifizierung) beruhen. Bellovin kommt zu dem Schluß, daß die Verschlüsselung auf Netzwerkebene sich auf wenige, wohldefinierte Bereiche beschränken sollte. Das sind im wesentlichen die im nächsten Abschnitt aufgeführten.

4.4 Fazit

IPSec stellt einen entscheidenden Schritt hin zu mehr Sicherheit für die Kommunikation über das Internet dar. Die in jüngster Zeit immer wieder beobachteten Angriffe mittels IP-Spoofing können durch den Einsatz der Protokolle AH und ESP ebenso verhindert werden wie Angriffe durch das Ausnutzen von Sicherheitslücken aller Protokolle, die auf das Internet-Protokoll aufsetzen. Die Implementierung von IPSec als Bestandteil des Internet-Protokolls der nächsten Generation wird letztendlich zu einer universell verfügbaren Sicherheits-Infrastrukur führen, die nicht auf proprietäre Lösung einzelner Hersteller angewiesen und somit zukunftssicher ist. Während die technische Entwicklung des IPv6-Standards weitgehend abgeschlossen ist, läßt sich die Frage nach dem Zeitpunkt, zu dem die universelle Verfügbarkeit erreicht sein wird, derzeit nicht beantworten[5]. Bis zu diesem Zeitpunkt wird der sinnvolle Einsatz von IPSec auf die Realisierung von virtuellen privaten Netzwerken und den Anschluß mobiler Rechner mit starker Authentifizierung an Firmennetzwerke beschränkt bleiben müssen.

Zu diesem Zweck allerdings ist IPSec bestens geeignet. Die Vertraulichkeit von sensitiven Daten ist durch die Stärke der Verschlüsselung, die durch den Einsatz von Algorithmen wie Triple-DES und zukünftig AES erreicht wird, hinreichend gewährleistet. Auch die Risiken, die durch eine Anbindung von lokalen Netzwerken an das Internet enstehen, werden durch den überlegten Einsatz der IPSec-Protokolle erheblich reduziert. Die Standardisierung der Protokolle und ihrer Inkorporation in IPv6 garantiert die Investitionssicherheit in entsprechende Hard- und Softwarelösungen, die durch bisherige proprietäre VPN-Lösungen nicht gegeben war.

[5] Unter http://www.ipv6.org finden sich Listen der derzeit über IPv6 erreichbaren Websites sowie von Software, die dieses Protokoll unterstützen. Solange mangels Masse eine solche Liste überhaupt aufstellbar ist, kann von einer Umsetzung von IPv6 als dem Standardprotokoll im Internet nicht die Rede sein.

5

Eine Demonstration mit VPN+

Zur praktischen Demonstration einer IPSec-Implementierung wurde ein minimales Testnetzwerk angelegt. Bei den beiden miteinander kommunizierenden Rechnern handelt es sich um PCs auf Intel-Basis unter dem Betriebssystem Windows NT 4.0 der Firma Microsoft. Als Angriffsrechner dient ebenfalls ein Intel-PC unter dem freien Betriebssystem Linux. In der minimalen Testkonfiguration dient dieser Rechner sowohl als Ausgangspunkt für die Angriffe als auch als Überwachungsstation für das Netzwerk, um die Funktionsweise der Angriffe und die Auswirkungen des Einsatzes der IPSec-Protokolle zu dokumentieren.

Die Demonstration dient ausdrücklich *nicht* zur Evaluierung oder Performancemessung der eingesetzten kommerziellen IPSec-Software, sondern nur zur Veranschaulichung der in den vorangegangenen Kapiteln diskutierten Mechanismen der IPSec-Protokolle. Eine Bewertung der Software VPN+ findet sich unter anderem in [Zdn99b], für Daten zur Performance siehe auch [Vpn99b].

5.1 Aufbau des Testnetzwerks

Abbildung 5.1 zeigt den Aufbau des Testnetzwerks. Die Rechner NT1 und NT2 mit den IP-Adressen 1.0.0.1 und 1.0.0.2 bezeichnen die beiden zur Kommunikation miteinander berechtigten Systeme. Prinzipiell könnten sie genauso gut als Gateways für dahinter liegende lokale Netzwerke dienen. Beide Rechner werden mit dem Betriebssystem Windows NT 4.0 Workstation betrieben. Zur Vorführung des TCP-SYN-Angriffs wurde dabei bewußt auf die Installation des aktuellen Servicepacks von Microsoft verzichtet.

Ebenfalls auf beiden Rechnern ist die IPSec-Software VPN+ Client von Datafellows installiert. Die Installation erfolgte jeweils lokal von der CD, die Möglichkeit einer zentralen Installation von einem Server aus ist gemäß der Dokumentation jedoch ebenfalls vorgesehen. Im konkreten Fall mußte aufgrund der Einschränkungen der Testumgebung auf diese Methode verzichtet werden.

Der Rechner NT1 dient im Testnetzwerk gleichzeitig als Administrationsrechner für VPN+. Obwohl die Administrationssoftware durch ein Paßwort geschützt ist, sollte sie in einer "echten" Netzwerkumgebung aus Sicherheitsgründen auf einen eigenständigen Rechner ausgelagert werden; die Funktionsweise der Software ist davon jedoch unabhängig.

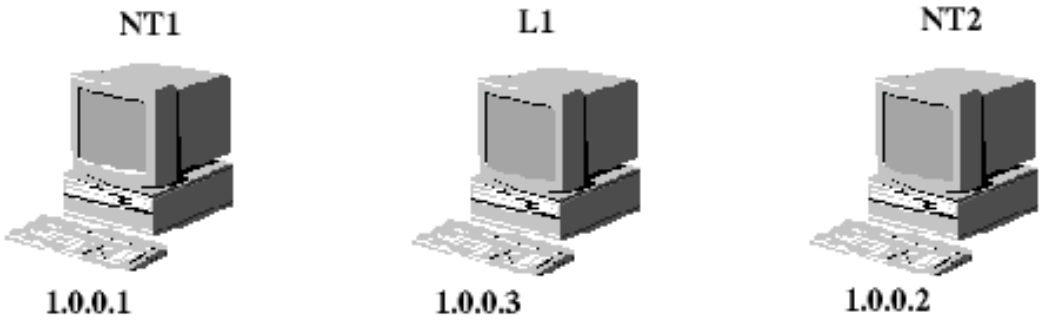

Abbildung 5.1: Beispielnetzwerk mit zwei IPSec-Hosts

Auf beiden NT-Rechnern laufen die Microsoft Peer Web Services, also jeweils ein FTP- und ein HTTP-Server mit den üblichen Ports 21 beziehungsweise 80.

Der Rechner L1 läuft unter dem Betriebssystem Linux, Kernelversion 2.0.36. Zur Beobachtung des Verkehrs auf dem Netzwerk ist darauf die unter der GPL entwickelte Software *ethereal* (Version 0.7.8) installiert, mit der man unter anderem sämtliche TCP-Pakete nach Klartext filtern kann. Dieses Programm ist im Prinzip ein grafisches Frontend zu dem Netzwerk-Analyseprogramm *tcpdump*, das zum Standard-Installationsumfang jeder Unix-Variante gehört.

Darüberhinaus sind auf dem Linux-Rechner Programme zum Durchführen eines TCP-SYN-Angriffs und zum Unterbrechen einer bestehenden TCP-Verbindung installiert. Diese wurden im Quellcode aus dem World Wide Web bezogen und leicht modifiziert auf der Linux-Plattform kompiliert[1].

5.2 Die Software: VPN+

Die kommerzielle Softwaresuite F-Secure VPN+ der Firma Datafellows besteht aus mehreren Komponenten. Zur IPSec-konformen Absicherung der Kommunikation über ein TCP/IP-Netzwerk dient die Clientsoftware VPN+ Client im Zusammenspiel mit dem Verwaltungstool

[1] Auf eine Quellenangabe wird hier aus naheliegenden Gründe verzichtet.

F-Secure Management Agent. Sie kann sowohl auf NT Workstation als auch Server installiert werden und implementiert die komplette IPSec-Protokollfamilie. Zur Absicherung von Windows-NT-Datei- bzw. Applikationsservern dient die Serverversion von VPN+, welche im Testszenario nicht zum Einsatz kam.

Weiterer notwendiger Bestandteil jedes VPN+-Netzwerks ist der F-Secure Administrator. Neben der Verwaltung weiterer F-Secure-Produkte dient er zum Erstellen, Verwalten und Verteilen der Sicherheitsrichtlinien für die VPN+-Clients. Er implementiert somit die nominale Sicherheitsrichtlinien-Datenbank des RFC 2401, die in Abschnitt 3.1.3 erläutert wurde.

5.2.1 Administration der Sicherheitsrichtlinien

Zur Administration der Sicherheitsrichtlinien und deren Distribution an die einzelnen VPN+--Klienten sind mehrere Möglichkeiten vorgesehen:

* Für große Netzwerke empfiehlt Datafellows die Einrichtung eines Management-Servers. Dies ist ein dedizierter Rechner unter der Serverversion von Windows NT, der nur zur Distribution der mit dem F-Secure Administrator erstellten Sicherheitsrichtlinien (*Policies*) dient. Besonders bei der gemeinsamen Verwaltung mehrerer Subnetze mit unterschiedlichen NT-Domänen ist dies die einzig praktikable Vorgehensweise.
* Alternativ dazu besteht die Möglichkeit, ein gemeinsam genutztes Verzeichnis innerhalb der NT-Domäne zu nutzen. Für kleinere lokale Netzwerke hat diese Methode den Vorteil, mit weniger Ressourcen auszukommen, da auf einen eigenständige Management-Servers verzichtet werden kann.
* Als dritte Option besteht die Möglichkeit der manuellen Installation der Sicherheitsrichtlinien über Datenträger auf jedem Client. Diese Methode wird mit zunehmender Größe des Netzwerks offensichtlich unpraktikabel, hat aber den Vorteil der erhöhten Sicherheit, da keinerlei sicherheitsrelevante Informationen über das Netzwerk übertragen werden. Zudem ist die Implementierung der manuellen Verteilung von Schlüsseln und Zertifikaten im Rahmen des Internet Key Exchange (die im Falle von VPN+ an eine bestimmte Sicherheitsrichtlinie gebunden sind) ein notwendiges Kriterium für die Konformität mit dem RFC 2401.

Aufgrund des geringen Umfangs des Testnetzwerks, insbesonder mangels eines Rechners mit der Serverversion von Windows NT, erfolgte die Verteilung der Sicherheitsrichtlinien

bei dieser Demonstration ausschließlich mittels der manuellen Methode[2]. Dabei wurden die verschieden Sicherheitsrichtlinien mittels des F-Secure Administrators auf dem Rechner NT1 lokal erzeugt und manuell in die entsprechenden Verzeichnisse der beiden Rechner kopiert. Während dieses Verfahren reibungslos funktionierte, schlug der Versuch fehl, diese Sicherheitsrichtlinien mittels des Management Agent während des Betriebs der Clientsoftware zu importieren. Nur durch das Anhalten der Software, manuelles Kopieren der Sicherheitsrichtlinien-Dateien und anschließenden Neustart von VPN+ war ein Wechsel der Sicherheitsrichtlinien zu erreichen. Die Sicherheit der Systeme wird von diesem Verfahren jedoch nicht beeinträchtigt, da das Herunterfahren eines einmal installierten VPN+-Klienten dazu führt, daß keine Verbindung mehr über das Netzwerk aufgebaut werden kann. Dieses Verhalten verhindert auch eine versehentliche oder bewußte Deaktivierung der Software durch einen Benutzer ohne Administrationsrechte.

5.2.2 Algorithmus- und Schlüsselmanagement

VPN+ unterstützt alle im RFC 2409 beschriebenen Varianten des Schlüsselaustauschs. In der Testkonstellation wurde für die IKE-Phase 1 der Hauptmodus mit Authentifizierung mittels Zertifikat gewählt. Die Generierung der Zertifikate erfolgt mittels des F-Secure Certificate Wizard, der ein Bestandteil der Administrationssoftware ist. Bei der Erstinstallation dieser Software wird ein Master-Zertifikat generiert, von dem Zertifikate für die einzelnen Client-Rechner abgeleitet werden. Diese Zertifikate sind Bestandteil der manuell verteilten Sicherheitsrichtlinien.

VPN+ unterstützt die folgenden Algorithmen und Authentifizierungsverfahren:

* Authentifizierung: HMAC-MD5-96, HMAC-SHA-1-96
* Verschlüsselung: Triple-DES (168 Bit), DES (56), Blowfish (128), CAST128 (128)
* Schlüsselaustausch: RSA-Signatur mit und ohne X.509v3 Zertifikat, DSA-Signatur, PKCS #7 und #10, Preshared Key

Folgende Algorithmen wurden in der Demonstration ausgewählt:

* IKE: IKE-Gruppe 5, Triple-DES im CBC-Modus und HMAC-SHA-1-96, Authentifi-

[2]Der Versuch, die zweite Methode zu nutzen, scheiterte an der Weigerung des F-Secure Administrators, die Rechner innerhalb der lokalen NT-Workgroup zu erkennen. Vermutlich funktioniert diese Methode nur in einem "vollwertigen" NT-Netzwerk, d.h. unter Einbindung mindestens eines Domain-Servers. Die Dokumentation zu diesem Aspekt ist bestenfalls verbesserungsfähig.

zierung mit X.509, Oakley-Gruppe 5[3]

* ESP: Triple-DES im CBC-Modus, HMAC-SHA-1-96
* AH: HMAC-SHA-1-96

5.3 Ausgewählte Angriffe und ihre Auswirkungen

Zur Demonstration der Auswirkungen ausgewählter Netzwerkangriffe ohne den Einsatz von IPSec wurde als Sicherheitsrichtlinie zunächst der Modus "Bypass all" für die VPN+-Software konfiguriert. Das bedeutet nichts anderes, als daß ausnahmslos alle IP-Pakete ohne Bearbeitung durch VPN+ durchgelassen werden[4].

5.3.1 Abhören des Netzwerkverkehrs

Zum Überwachen der Netzwerkverbindung wird auf dem Rechner L1 das oben erwähnte Programm ethereal eingesetzt. Damit kann der gesamte Datenverkehr auf dem überwachten Netz in Echtzeit unter verschiedenen Gesichtspunkten überwacht werden. Einzelne Pakete können ebenso analysiert werden wie Datenströme, die nach verschiedenen Kriterien gefiltert werden. Zum Beispiel kann mittels eines simplen Filters die Anzeige nur auf die Pakete beschränkt werden, die mit dem File Transfer Protocol (FTP) übertragen werdem. Abbildung 5.2 zeigt das Protokoll aller während eines FTP-Verbindungsaufbaus übertragenen TCP-Pakete. Wie im mittleren Fenster zu sehen ist, können so auf sehr bequeme Weise die im Klartext übertragenen Daten wie Loginname und Passwort beim Aufbau einer FTP-Verbindung abgefangen werden. Gleiches gilt für alle anderen Protokolle (Telnet, rlogin etc), die ihre Nutzlast unverschlüsselt übertragen.

Zusätzlich erhält ein potentieller Angreifer eine Reihe von Informationen, die für weiterführende Angriffe instrumentalisiert werden können, wie etwa die TCP-Sequenznummer oder die Absender-Portnummer. Beide werden zum Beipiel für den Angriff in Abschnitt 5.3.3 benötigt.

[3] In RFC 2409 sind nur vier Oakley-Gruppen definiert. Gruppe 5 entstammt der derzeitigen Arbeitsversion (Internet-Draft) [Har99], die den RFC ersetzen wird.

[4] Nach der Erstinstallation der VPN+-Clients ist dies die voreingestellte Richtlinie. Das ist zur verteilten Konfiguration über ein Netzwerk zwar notwendig, unter Sicherheitsaspekten jedoch zumindest fragwürdig.

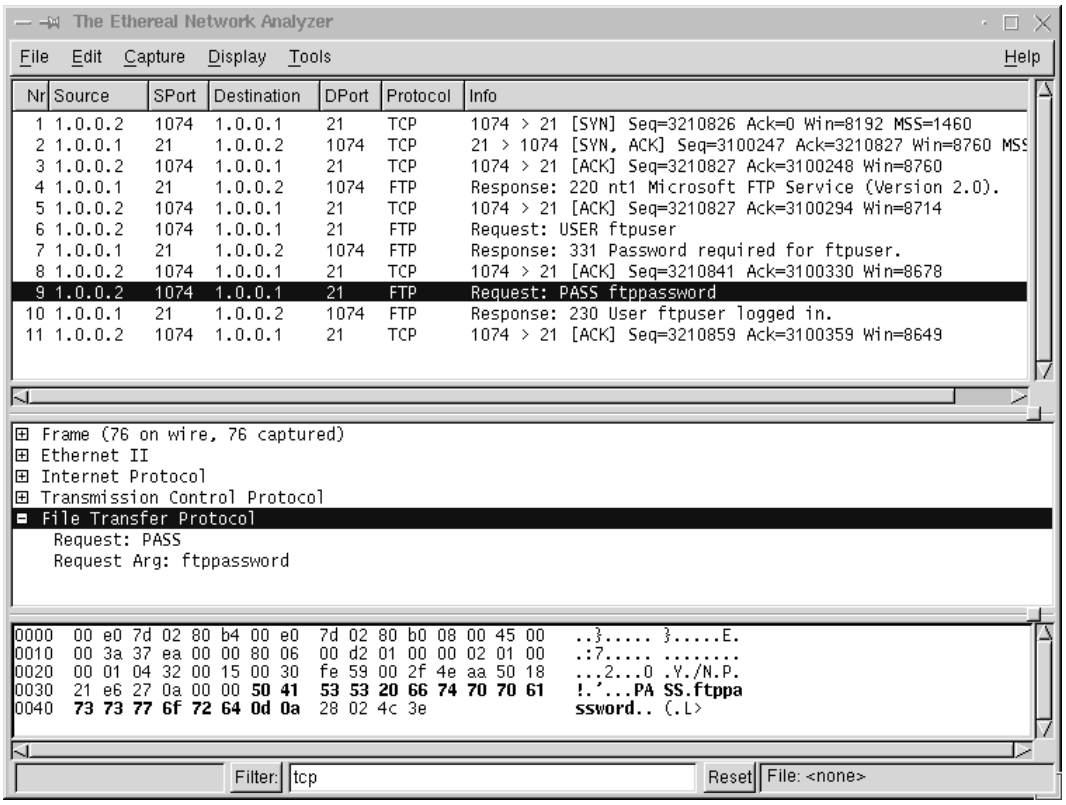

Abbildung 5.2: Mitlesen eines FTP-Verbindungsaufbaus

5.3.2 TCP-SYN-Überflutung

Dieser klassische Internetangriff wird mittels des Programms `synflooder` durchgeführt, daß in verschiedenen Variationen im World Wide Web frei verfügbar und über Suchmaschinen leicht auffindbar ist. Das in C geschriebene Programm umfaßt im Quellcode gerade 164 Programmzeilen.

Zur Demonstration wurden von L1 mittels dieses Programms 15 TCP-SYN-Request--Pakete mit der gefälschten Absenderadresse 1.0.0.4 an der FTP-Port (21) des Rechners NT1 geschickt. Abbildung 5.3 zeigt die Ausgabe des Befehls `netstat -n -p tcp` auf NT1, mit dem alle bestehenden TCP-Verbindungen angezeigt werden. Die Auswirkungen des Angriffs sind offensichtlich: Alle TCP-Verbindungen sind blockiert, eine FTP-Anfrage von 1.0.0.2 an 1.0.0.1 wird zurückgewiesen. Darüberhinaus wurde im Testbetrieb der Rechner NT2 durch diesen Angriff in schöner Regelmäßigkeit zum vollständigen "Absturz" gebracht, dem nur durch einen Kaltstart beizukommen war[5].

[5] Nach Auskunft von Microsoft ist dieser Fehler mit dem letzten Servicepack behoben worden. Siehe dazu http://support.microsoft.com/support/kb/articles/Q142/6/41.asp.

```
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    1.0.0.1:21             1.0.0.4:52480          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:52736          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:52992          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:53248          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:53504          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:53760          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:54016          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:54272          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:54528          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:54784          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:55040          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:55296          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:55552          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:55808          SYN_RECEIVED
  TCP    1.0.0.1:21             1.0.0.4:56064          SYN_RECEIVED
  TCP    127.0.0.1:1025         127.0.0.1:1028         ESTABLISHED
  TCP    127.0.0.1:1028         127.0.0.1:1025         ESTABLISHED
```

Abbildung 5.3: Auswirkungen eines TCP-SYN-Angriffs

5.3.3 Verbindungsabbruch einer FTP-Verbindung mit TCP-Reset–Paketen

Dieser Angriff bedient sich der Methode des Sequenznummern-Ratens, um eine bestehende TCP-Verbindung mittels eines Reset-Pakets zu unterbrechen. Mit Hilfe eines Paketsniffers wird die Sequenznummer eines Pakets dieser Verbindung ausgelesen. Daraufhin werden TCP-Pakete mit gesetzten Reset-Flag (RST) und der (gefälschten) Absenderadresse eines der beteiligten Rechner an den anderen Host geschickt. Stimmt die Sequenznummer mit dem internen Zähler dieses Rechners überein, wird das Paket als gültig akzeptiert, und die Verbindung wird unterbrochen, da Reset-Pakete keinen ACK-Wert zur Bestätigung des vorangegangen Pakets enthalten.

Die Schwierigkeit bei diesem Angriff besteht im Erraten der korrekten Sequenznummer. Durch das Senden einer größeren Anzahl von RST-Paketen in schneller Folge mit sukzessive erhöhter Sequenznummer wird in der Praxis eine hohe Trefferquote erreicht. Abbildung 5.4 zeigt einen erfolgreich durchgeführten Angriff auf eine FTP-Verbindung zwischen den Hosts NT2 (Client) und NT1 (Server). Die Pakete 1 bis 5 zeigen den Ablauf der Anfrage “CWD” (Change Working Directory) von NT2 an NT1 und die Antwort von NT1. Anhand der Portnummern dieser beobachteten Kommunikation wurde ein Angriff von L1 auf diese Verbindung durchgeführt. Das dabei eingesetzte Programm wartet auf eine weiteres TCP-Paket

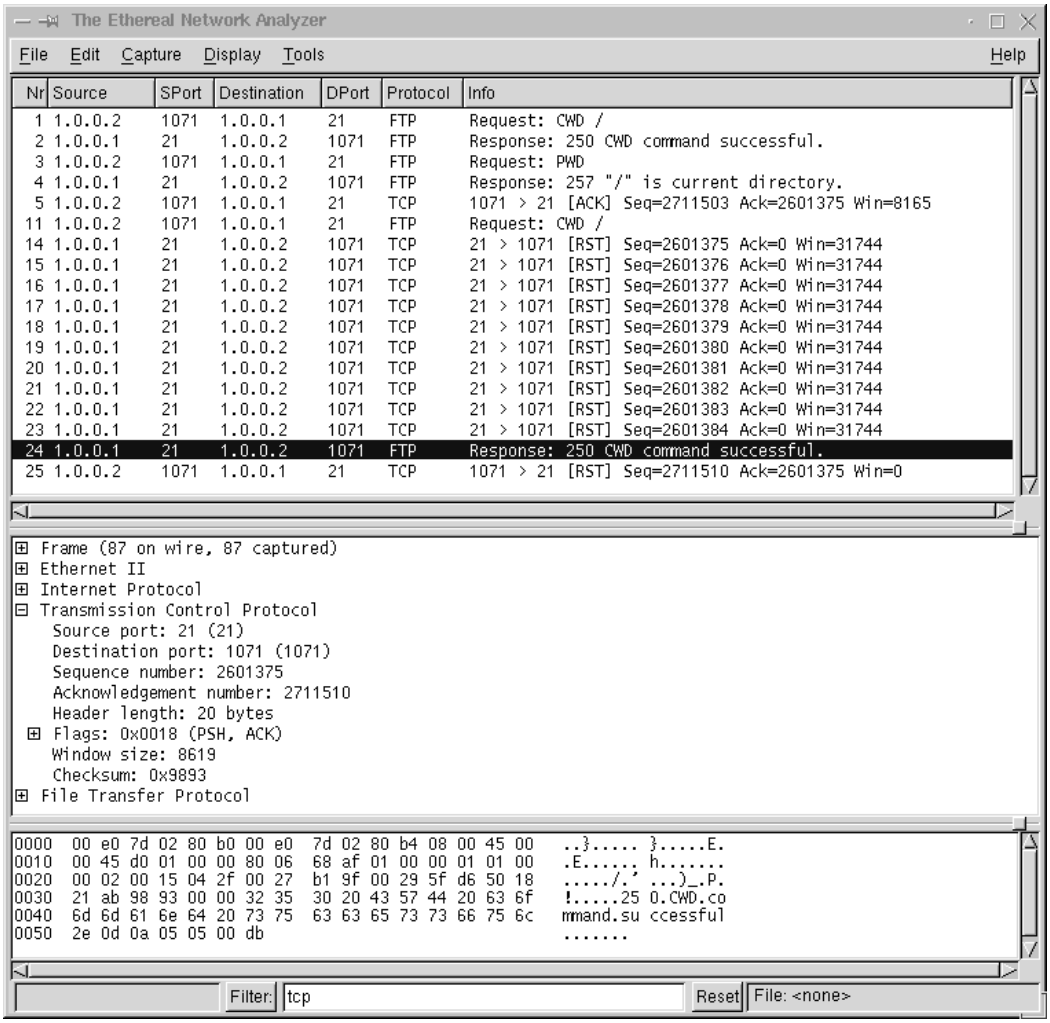

Abbildung 5.4: Abbruch einer FTP-Verbindung mit Reset-Paketen

mit den entsprechenden Adressen und Portnummern, um eine aktuelle Seqzuenznummer zu erhalten (Paket 11 in der Abbildung)[6]. Anschließend werden zehn Reset-Pakete mit den gefälschten Absenderdaten von NT1 an NT2 gesendet (Pakete 14 bis 23). Wie in der Abbildung zu sehen ist, ist die Sequenznummer des ersten gefälschten Pakets (14) mit der des legitimen Antwortpakets von NT1 (24, erkennbar im mittleren Fenster) identisch. Das zuerst bei NT2 ankommende Reset-Paket wird als gültig akzpetiert und mit einem weiteren Reset-Paket bestätigt (25). Die Verbindung ist damit unterbrochen, das gültige Paket 24 geht bereits verloren.

Im vorgeführten Beispiel sorgt bereits das erste gefälschte Reset-Paket für einen Abbruch

[6]Die in der Abbildung nicht angezeigten Pakete gehören zu anderen Netzwerkdiensten und wurden aus der Anzeige herausgefiltert.

der Verbindung. Für den Fall, daß dem Paket, aus dem die aktuelle Sequenznummer ausgelesen wird, weitere Datenpakete folgen und die Sequenznummer dementsprechend erhöht wird, würde das Senden mehrerer Pakete mit aufsteigender Sequenznummer mit hoher Wahrscheinlichkeit zu einem Treffer führen.

Ein Ausschnitt aus dem Logfile des FTP-Clients auf NT2 zeigt die Auswirkungen auf den Initiator der beiden ausgeführten CWD-Befehle:

```
CWD /
250 CWD command successful.
PWD
257 "/" is current directory.
CWD /
! Receive error: connection reset
```

5.4 Auswirkungen des Einsatzes von VPN+

Der Einsatz der IPSec-Software VPN+ auf den beiden Rechnern NT1 und NT2 kann bei entsprechender Konfiguration alle demonstrierten Angriffe wirkungsvoll unterbinden. Dabei muß allerdings beachtet werden, daß zur Verhinderung der Denial-of-Service-Angriffe die Sicherheitsrichtlinien so gewählt sein müssen, daß keine Verbindungen zugelassen sind, die IPSec umgehen können. VPN+ sieht gemäß der Spezifikation des RFC 2401 drei mögliche Modi zur Behandlung von IP-Paketen vor: IPSec, Bypass (Passieren) und Discard (Verwerfen). Die genannte Einschränkung bedeutet in diesem Fall, daß der Modus Bypass unter keinen Umständen zum Einsatz kommen darf.

Im Testnetzwerk wurden verschieden Sicherheitsrichtlinien zum Schutz der Verbindung zwischen NT1 und NT2 erstellt, um den Einsatz der Protokolle AH und ESP zu demonstrieren. Alle diese Sicherheitsrichtlinien wurden so gewählt, daß sämtliche IP-Pakete zwischen genau diesen beiden Rechnern durch IPSec geschützt und alle anderen verworfen werden. Die Unterscheidung erfolgt also nur anhand des Selektors IP-Quelladresse aus Abschnitt 3.1.3 und unabhängig von allen anderen Selektoren wie Ports oder Protokolle. Diese an sich recht simple Sicherheitsrichtlinie gewährleistet neben einer einfachen Konfigurierbarkeit ein hohes Maß an Schutz und entspricht darüberhinaus genau dem Szenario, in dem IPSec voraussichtlich am häufigsten zur Anwendung kommen wird: dem Schutz einer einzelnen, wohl-

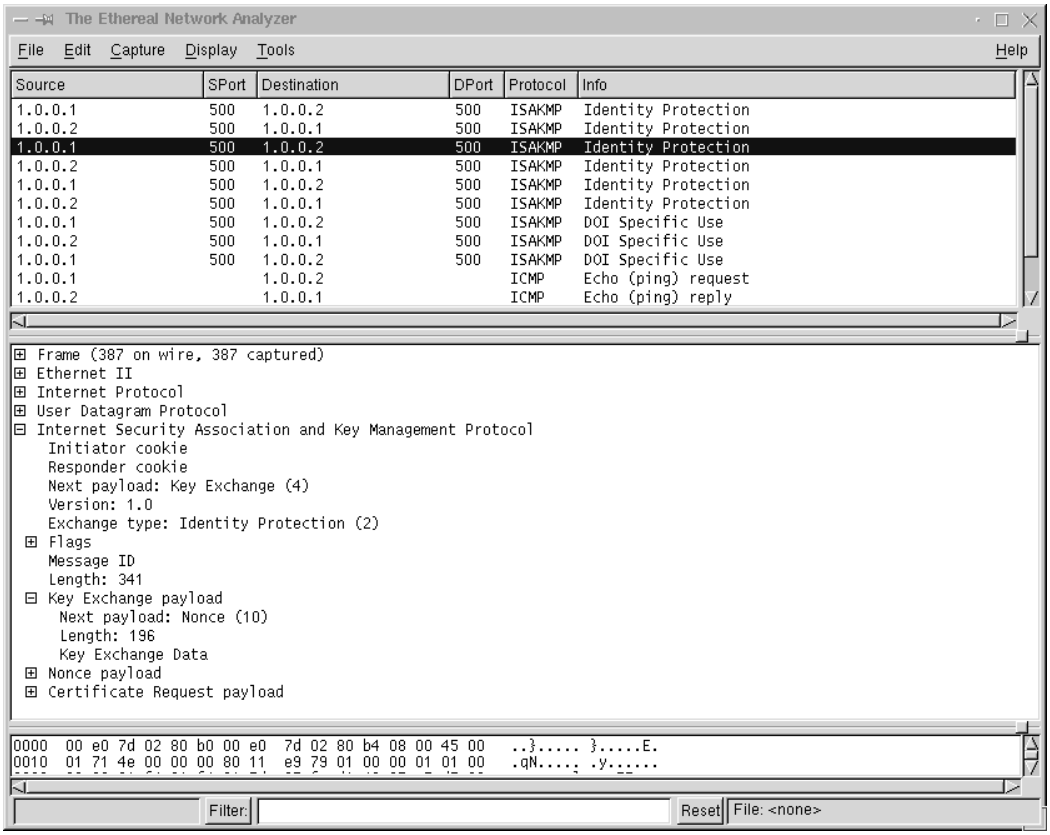

Abbildung 5.5: IKE im Einsatz

definierten Verbindung zwischen zwei Rechnern. Ob diese Rechner wie in der Demonstration die Endpunkte der Kommunikation oder wie im Falle eines virtuellen privaten Netzwerks zwei Sicherheits-Gateways sind, spielt für die Betrachtung der Sicherheit keine Rolle. Der Unterschied zwischen beiden Variationen besteht lediglich in der Verwendung des Transport- beziehungsweise des Tunnelmodus bei der Anwendung der IPSec-Protokolle.

5.4.1 Aufbau einer IPSec-Sicherheitsassoziation

Abbildung 5.5 zeigt den Aufbau einer Sicherheitsassoziation zwischen NT1 und NT2 mittels IKE, um den praktischen Einsatz der in Abschnitt 3.4 beschriebenen Verfahren zu veranschaulichen. In diesem Fall bestand zwischen beiden Rechnern zuvor keine Verbindung, so daß durch das Senden eines ICMP-Echo-Request-Pakets (“ping”) von NT1 an NT2 der Schlüsselaustausch gemäß der eingesetzten Sicherheitsrichtlinie angestoßen wurde. Wie im mittleren Fenster der Abbildung 5.5 zu erkennen ist, erfolgt die Authentifizierung der Teilnehmer mittels der zuvor generierten und verteilten Zertifikate. Die ersten sechs

Nachrichten bilden zusammen die Phase 1 zur Etablierung der IKE-Sicherheitsassoziation unter Verwendung des Hauptmodus. Die weiteren drei Nachrichten etablieren die eigentliche IPSec-Sicherheitsassoziation.

In allen im Rahmen der Demonstration eingesetzten Sicherheitsrichtlinien wurden die im Abschnitt 5.2.2 aufgelisteten Protokolle im Transportmodus explizit gewählt. Die Software enthält darüberhinaus eine ganze Reihe an vorkonfigurierten Proposals, die im Anhang der mitgelieferten Dokumentation aufgelistet werden[7]. Durch Einschränkung der Proposals mittels zusätzlicher Flags kann eine sehr detaillierte Auswahl der gewünschten Algorithmen erfolgen.

5.4.2 AH und ESP im Einsatz

Beide im Rahmen der Demonstration durchgeführten Denial-of-Service-Angriffe konnten schon durch den Einsatz von AH im Rahmen der oben ausgeführten Sicherheitsrichtlinien verhindert werden. Zwar sind die Verbindungsdaten des TCP-Protokolls sowie der darauf aufsetzenden Protokolle weiterhin sichtbar; sie können aber für keinen der genannten Angriffsmethoden mißbraucht werden, da das Fälschen der IP-Absenderadresse als ihre gemeinsame Grundlage durch den Einsatz von AH wirkungsvoll verhindert wird. Gefälschte SYN-Pakete im Rahmen des Angriffs aus Abschnitt 5.3.2 werden auch bei gespoofter Absenderadresse eines der Rechner NT1 oder NT2 ohne weitere Bearbeitung oder Beantwortung fallengelassen, da sie keinen Authentication Header besitzen. Selbst wenn die IP-Pakete mit einem solchen Header ausgestattet wären, der zum Beispiel aus einem legitimen IP-Paket zwischen NT1 und NT2 kopiert werden könnte, so würde dieses durch den Anti-Replay-Algorithmus erkannt und verworfen werden [8].

Der Angriff auf eine TCP-Verbindung mittels Reset-Paketen schlug in der Praxis schon dadurch fehl, daß das eingesetzte Programm nicht in der Lage war, den TCP-Header in einem AH-geschützten IP-Paket zu erkennen. Dies ist natürlich keine zulässige Demonstration der Sicherheit von AH; da aber die Abwehr von IP-Spoofing anhand des TCP-SYN-Angriffs demonstriert werden kann, ist die Wirksamkeit dieses Protokolls gegenüber allen darauf basierenden Angriffen hinreichend veranschaulicht.

[7] Die Dokumentation ist im WWW verfügbar unter [Vpn99a].

[8] Diese Diskussion muß zwangsläufig akademisch bleiben, da (noch) kein Angriffsprogramm in Umlauf zu sein scheint, welches zur Durchführung von Replay-Angriffen auf eine IPSec-Implementierung in der Lage wäre.

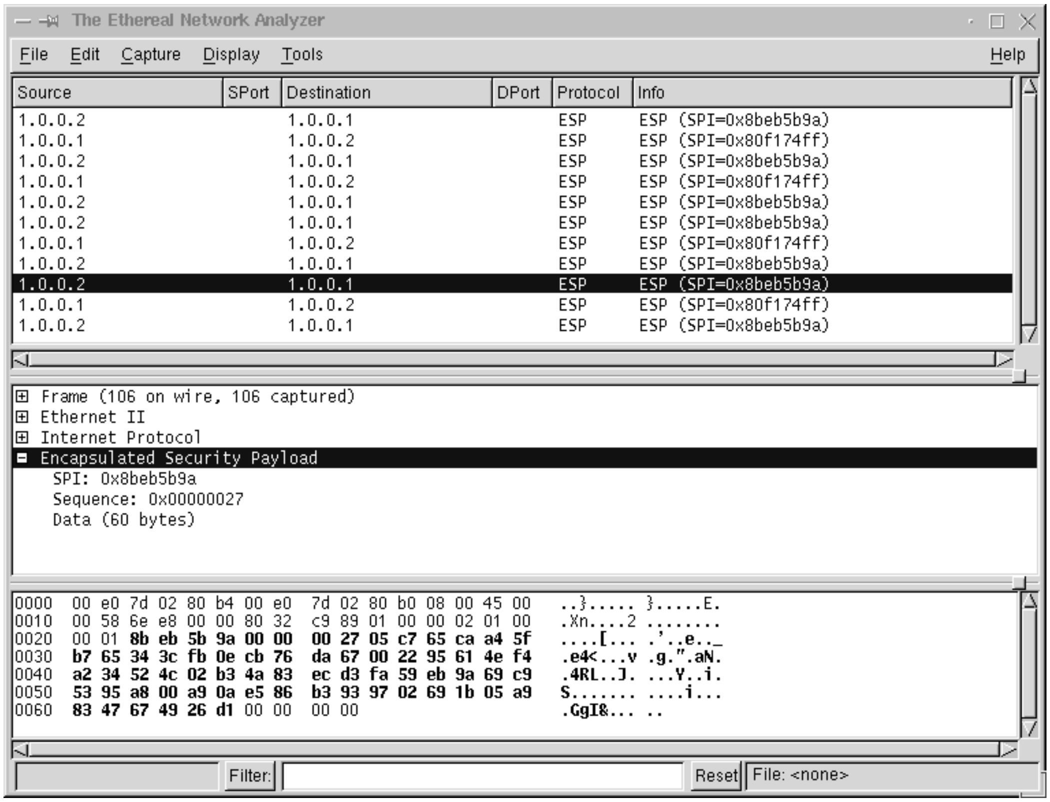

Abbildung 5.6: ESP-geschützer FTP-Verbindungsaufbau

Der Einsatz von ESP mit VPN+ erzielt in der Demonstration denselben Effekt. Zusätzlich wird das Auslesen der Verbindungsdaten aus den TCP-Paketen durch die Verschlüsselung unmöglich gemacht. In VPN+ sind neben dem zur Zeit noch obligatorischen einfachen DES die Verschlüsselungsalgorithmen 3DES (168 Bit), Blowfish (128 Bit) und CAST128 (128 Bit) implementiert, mit denen die Verschlüsselung für heutige Maßstäbe als sicher angesehen werden kann. Abbildung 5.6 zeigt die während eines FTP-Verbindungsaufbaus übertragenen Pakete einer mit ESP verschlüsselten Verbindung zwischen 1.0.0.1 und 1.0.0.2 analog zu Abbildung 5.2. Offensichtlich sind über die Verwendung des Protokolls ESP hinaus keine Information über die übertragenen Daten ersichtlich. Anschaulicher kann man den durch ESP gewährten Schutz leider nicht machen, ohne einen Brute-Force-Angriff auf den mitgelesenen Chiffretext durchzuführen. Klar zu erkennen ist jedoch, daß der Typ der IP-Nutzlast nicht mehr offenliegt. Eine komplett mit ESP geschützte Verbindung wird keine Details der auf ihr übertragenen Daten preisgeben, die über die in der Abbildung gezeigten hinausgehen.

5.4.3 Auswirkungen eines unvollständigen Schutzes

Um die Unzulänglichkeit der IPSec-Protokolle außerhalb des Einsatzes als Schutz dedizierter Verbindungen zu veranschaulichen, wurde der Rechner NT1 mit einer Sicherheitsrichtlinie ausgestattet, die einen ungeschützten Zugang auf den TCP-Port 21 zuläßt (Bypass-Modus für Verbindungen zwischen 0.0.0.0/0 und 1.0.0.1:21) und IP-Pakete an alle anderen Adressen verwirft. Die Online-Dokumentation von Datafellows führt eine analoge Konfiguration für den Port 80 ausdrücklich als Beispiel für die IPSec-Absicherung eines Webservers an[9]. Wörtlich heißt es dort "You can make a web server far more secure [...]". [10]. Dieser Aussage kann nicht wiedersprochen werden, da mit dem Zugang zum Rechner NT1 über das Netzwerk auch die Möglichkeiten eingeschränkt werden, vorhandene und potentielle Sicherheitslöcher auszunutzen. Andererseits wird unterschlagen, daß mit dem einen ungesicherten Port der Schutz vor allem vor Denial-of-Service-Angriffe praktisch zunichte gemacht wird. So ist der Angriff mittels TCP-SYN-Überflutung auf den FTP-Port weiterhin möglich. Da die IPSec-Protokolle AH und ESP in diesem Szenario überhaupt nicht zum Einsatz kommen, ist es darüberhinaus auch nicht möglich, Angriffe mit gefälschten IP-Adressen überhaupt als solche zu erkennen. RFC 2402 erklärt beispielweise die Entdeckung eines IP-Pakets mit ungültigem AH oder ungültiger Sequenznummer zu einem "vernehmbaren Ereignis" (*audible event*), dessen Auftreten in einer Logdatei protokolliert werden sollte. Derartige Mechanismen zur Entdeckung von möglichen Angriffen werden durch den Einsatz von IPSec in der hier vorgestellten Konfiguration wirkungslos gemacht.

5.5 Fazit

Die Demonstration der Software VPN+ bestätigt sowohl die Wirksamkeit der IPSec-Protokolle bei verantwortungsvollem Einsatz als auch die Beschränkung ihrer Wirksamkeit auf eng umrissene Einsatzgebiete. Während eine Absicherung fester Verbindungen mit ESP und AH ein hohes Maß an Sicherheit sowohl für die über diese Verbindung übertragenen Daten als auch für die an das Internet angeschlossenen lokalen Netzwerke gewährleistet, kann der unsachgemäße Einsatz von IPSec nicht alle Sicherheitsrisiken beseitigen. Es besteht dabei sogar die Gefahr, daß trotz der Installation einer entsprechenden Software immer noch vorhandene Angriffsmöglichkeiten nicht mehr als solche wahrgenommen werden, was zu einem Nachlassen der Kontrolle und damit einhergehend zu einem faktisch größeren Risiko

[9] Im konkreten Fall wurde ein FTP-Server gewählt, da der mit Windows NT 4.0 mitgelieferte Webserver ohne Servicepack beim ersten Zugriff zu einem Totalabsturz des Rechners führte.
[10] Siehe http://www.Europe.DataFellows.com/support/vpn+/sec-web.html

führen kann.

Die eingesetzte Software selbst weist einige Implementierungslücken auf, die sich direkt auf das Maß an erreichbarer Sicherheit auswirken. Am augenfälligsten ist das Fehlen der Möglichkeit, SA-Bündel gemäß dem RFC 2401 zu generieren (Siehe Abschnitt 3.1). Obwohl der Einsatz von ESP mit Authentifizierung (der bei VPN+ immer erfolgt) ein hohes Maß an Schutz gewährt, kann er doch nicht dasselbe leisten wie in Kombination mit AH. Abgesehen von der Möglichkeit von Insiderangriffen, wie sie in [Bel96] dargelegt wurde, kann eine Absenderauthentifizierung wegen der Tatsache, daß die Authentifizierung sich bei ESP nicht auf den äußeren IP-Header erstreckt, erst nach erfolgter Entschlüsselung greifen. Im Falle von Denial-of-Service-Angriffen mit gültigem SPI (der notwendigerweise nicht verschlüsselt ist) kann so zumindest ein unnötiger Rechenaufwand nicht vermieden werden.

Bis auf diese Schwächen implementiert VPN+ alle Vorgaben der IPSec-RFCs. Insbesondere der Schlüsselaustausch mittels IKE funkioniert problemlos. Die Demonstration bestätigt, daß zumindest die herkömmlichen Angriffe mittels IP-Spoofing und dem Ausnutzen inhärenter Schwächen höherer Protokolle wie TCP durch den konsequenten Einsatz von IPSec abgewehrt werden können.

6

Anhang

6.1 Ein Anti-Replay-Algorithmus

Replay-Angriffe erfolgen durch das erneute Senden von zuvor abgefangenen Paketen. Wenn ein Angreifer zum Beispiel weiß, daß ein bestimmtes Paket die Daten einer Überweisung an ihn selbst enthält, so kann er durch wiederholtes Einspielen dieses Pakets eine Wiederholung der Überweisung veranlassen.

Sind die IP-Pakete durch AH oder ESP geschützt, so werden sie vom Empfänger aufgrund des SPI und der erfolgreichen Authentifizierung zunächst als gültig angesehen. Es muß also eine Überprüfung der Sequenznummer von AH bzw. ESP erfolgen. Durch den verbindungslosen Charakter des Inernet-Protokolls ist allerdings nicht gewährleistet, daß die Pakete den Empfänger in derselben Reihenfolge erreichen, in der sie verschickt wurden, so daß ein einfacher Zähler zu diesem Zweck nicht ausreicht. Aus diesem Grund wird ein sogenanntes verschiebbares Empfänger-Fenster eingesetzt, das in einer geeigneten Datenstruktur wie zum Beispiel einer Bitmap implementiert ist (siehe 3.1.2, SA-Parameter). Abbildung 6.1 zeigt die Funktionsweise des Algorithmus.

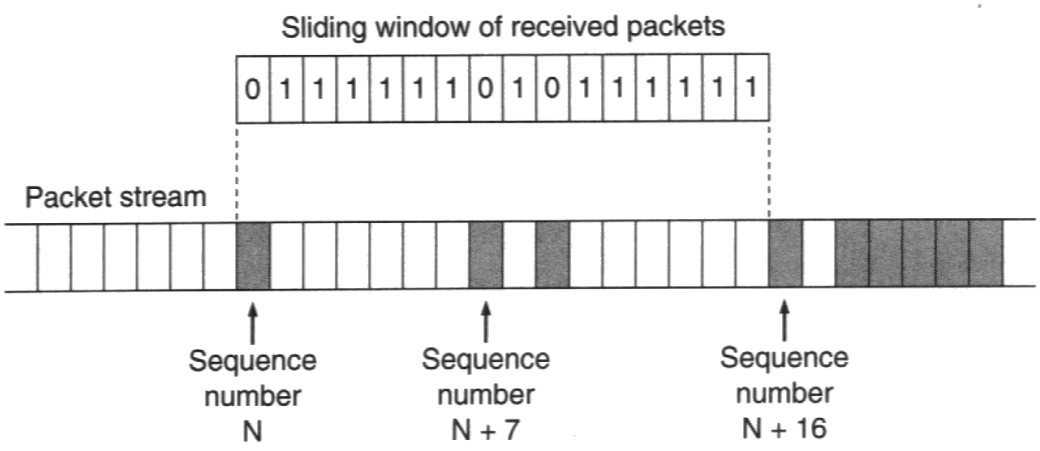

Abbildung 6.1: Ein 16-Bit-Empfangsfenster (aus [Dor99]

Die rechte Seite des Fensters repräsentiert den höchsten Wert aller bisher empfangenen Sequenznummern, der in der SAD gespeichert und in der Datenstruktur entsprechend markiert ist. Trifft ein neues Paket ein, so werden die folgenden drei Fälle unterschieden:

* Die Sequenznummer des eintreffenden Pakets ist größer als die aktuelle Sequenznummer (größer als N+15 in der Abbildung): Das Fenster wird entsprechend der Differenz zwischen neuer und alter höchster Sequenznummer nach rechts verschoben und markiert.
* Die Sequenznummer ist kleiner als die aktuell höchste Sequenznummer, aber noch innerhalb des Fensters (größer oder gleich N): Der entsprechende Eintrag des Fensters wird markiert.
* Die Sequenznummer ist kleiner als die am linken Rand des Fensters (kleiner N): Das Paket wird verworfen und geht verloren.

Durch diesen Mechanismus wird erreicht, daß jede Sequenznummer nur einmal akzeptiert wird, die Reihenfolge aber innerhalb der Fenstergröße nicht eingehalten werden muß[1]. Zu beachten ist dabei, daß die Authentifizierung des Pakets *nach* der Überprüfung der Sequenznummer erfolgt, um unnötige Berechnungen zu vermeiden, aber *vor* dem eventuellen Verschieben des Fenster, da sonst eine nicht authentifizierte hohe Sequenznummer zu einer Denial-of-Service-Attacke eingesetzt werden könnte.

[1] Die Mindestgröße des Fensters beträgt laut RFC 2401 32 Bit. In der Abbildung wurde aus Gründen der Überschaubarketi ein nur 16 Bit großes Fenster gezeigt.

Literatur

Bücher

[Dif88] W. Diffie. *The First Ten Years of Public-Key Cryptography*. Proceedings of the IEEE, Mai 1988

[Dor99] Naganand Doraswamy, Dan Harkins. *IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks*. Prentice Hall, 1999

[Feg99] Jalal Feghhi et al. *Digital Certificates: Applied Internet Security*. Addison-Wesley, 1999

[Fis96] Stephan Fischer et al. *Open Security: Von den Grundlagen zu den Anwendungen*. Springer, 1996

[Hui96] Christian Huitema. *IPv6: The New Internet Protocol*. Prentice Hall, 1996

[Pap96] Uday O. Pabrai, Vijay K. Gurbani. *Internet and TCP/IP Network Security: Securing Protocols and Applications*. McGraw-Hill, 1996

[Pfl97] Charles P. Pfleeger. *Security in Computing*. Prentice-Hall International, Second Edition, 1997

[Smi98] Richard E. Smith. *Internet-Kryptographie*. Addison-Wesley-Longmann, 1998

[Sta99] William Stallings. *Cryptography And Network Security: Principles And Practice*. Prentice Hall, Second Edition, 1999

Request for Comments

Alle RFC stehen auf dem FTP-Server des Fachbereichs Informatik der Universität Hamburg zur Verfügung: ftp://ftp.informatik.uni-hamburg.de/pub/doc/rfc

[Pos81] Jon Postel (Editor). *Internet Protocol*. RFC 791, September 1981

[rfc1636] R. Braden et al. *RFC 1636: Report of IAB Workshop on Security in the Internet Architecture*. RFC 1636, Juni 1994

[rfc2401] Stephen Kent, Randall Atkinson. *RFC 2401: Security Architecture for the Internet Protocol*. RFC 2401, November 1998

[rfc2402] Stephen Kent, Randall Atkinson. *IP Authentication Header*. RFC 2402, November 1998

[rfc2104] Hugo Krawczyk et al. *HMAC: Keyed-Hashing for Message Authentication*. RFC 2104, Februar 1997

[rfc2405] Cheryl Madson, Naganand Doraswamy. *The ESP DES-CBC Cipher Algorithm With Explicit IV*. RFC 2405, November 1998

[rfc2406] Stephen Kent, Randall Atkinson. *IP Encapsulating Security Payload*. RFC 2402, November 1998

[rfc2407] Derrell Piper. *The Internet Security Domain of Interpretation for ISAKMP*. RFC 2407, November 1998

[rfc2408] Douglas Maughan et al. *Internet Security Association and Key Management Protocol (ISAKMP)*. RFC 2408, November 1998

[rfc2409] Dan Harkins, Dave Carrel. *The Internet Key Exchange (IKE)*. RFC 2409, November 1998

[rfc2411] Rodney Thayer et al. *IP Security Document Roadmap*. RFC 2411, November1998

Internetquellen

Die hier angegebenen URLs existieren mit Stand vom 14. Dezember 1999. Alle Dokumente liegen dem Author vor.

[Bel96] Steven M. Bellovin. *Problem Areas for the IP Security Protocols*. Proceedings of the Sixth Usenix Unix Security Symposium, Juli 1996
ftp://ftp.cert.dfn.de/pub/docs/net/ipext.ps.gz

[Cer97] *CERT Coordination Center 1997 Annual Report (Summary)*
http://www.cert.org/annual_rpts/cert_rpt_97.html

[Cer98] *CERT Coordination Center 1998 Annual Report (Summary)*
http://www.cert.org/annual_rpts/cert_rpt_98.html

[CA9801] *CERT Advisory 98.01*
ftp://info.cert.org/pub/cert_advisories/CA-98.01.smurf

[CA9621] *CERT Advisory 86.21*
fttp:///info.cert.org/pub/cert_advisories/CA-96.21.tcp_syn_flooding

[Ips96] *A short overview of IP spoofing PART I*
http://eve.speakeasy.org/ dittrich/talks/seaslug/IP-spoof.1

[Har99] Dan Harkins, Dave Carrel. *The Internet Key Exchange*. Internet-Draft
http://www.ietf.org/internet-drafts/draft-ietf-ipsec-ike-01.txt

[Phr99] *IP-spoofing Demystified*. Phrack-MagazineVolume Seven, Issue Forty-Eight
http://www.fc.net/phrack/files/p48/p48-14.html

[Vpn99a] Dokumentation für VPN+ (PDF)
http://www.datafellows.com/download-purchase/manuals/crypto/vpnplus.pdf

[Vpn99b] Performancedaten für VPN+
http://www.Europe.DataFellows.com/support/vpn+/cases/vpn-tp.html

[Zdn99a] Dietmar Müller. *DES in wenigeer als 23 Stunden geknackt.*
http://www.zdnet.de/news/artikel/1999/01/20012-wc.html

[Zdn99b] Ken Philips. *Software VPN IPSec-friendly.* PC-Week, 22. März 1999
http://www.zdnet.com/products/stories/reviews/0,4161,394399,00.html
(November 1999)