Ruhr-Universität Bochum
Lehrstuhl für Kommunikationssicherheit
Internetsicherheit
Wintersemester 2002/2003

IPSec

Marcel Selhorst, 108 099 210313
Benedikt Gierlichs, 108 000 242347

Betreuer: Dipl.-Ing. Ahmad-Reza Sadeghi
vorgelegt am: 25. April 2003

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Gerade in der heutigen Zeit, in der Computer bereits nicht mehr wegzudenken sind, erfreut sich das Internet einer immer größer werdenden Beliebtheit. Schlagworte wie IT, www und eMail sind in aller Munde und niemand möchte mehr auf den Komfort verzichten, den er durch die steigende Nutzung des Internets gewonnen hat. Der steigende Vernetzungsgrad und die daraus folgende Dezentralisierung von Computersystemen ermöglichen dem Benutzer größere Freiheiten als je zuvor. Der Einsatz von netzwerkbasierten Applikationen steigt und somit auch die Notwendigkeit, die über offene Netzwerke zu übertragenen Informationen zu schützen. Das Internet ist aus seinen Kinderschuhen herausgewachsen und muss nun mit seinen Anfangsschwierigkeiten fertig werden, bei denen die Sicherheit die kleinste Rolle gespielt hat. Netzwerkprotokolle wie TCP/IP haben sich etabliert und sind mittlerweile in fast jedem Computersystem integriert. Die hierbei fehlende Sicherheit wird heutzutage oft kritisiert. Die Internet Engeneering Task Force (IETF) arbeitet auf Hochtouren an neuen, sichereren Protokollen und Kryptologen entwickeln neue Algorithmen, um die zu schützenden Daten für nicht-autorisierte Benutzer unzugänglich zu machen. Mittlerweile stehen der Computerindustrie viele verschiedene kryptographische und technische Werkzeuge zur Verfügung, um Kommunikation zu verschlüsseln oder zu verstecken, es stellt sich jedoch die Frage, wie diese Sicherheit am besten in das bestehende System integriert werden kann. Problematisch beim nachträglichen Hinzufügen von Sicherheit ist allein die Tatsache, dass Sicherheit kein Produkt ist, welches gekauft wird, sondern einen Prozess darstellt, der sich durch sämtliche an der Kommunikation beteiligten Anwendungen und Protokollschichten ziehen muss. Schon ein schwaches Glied kann die Sicherheitskette zerstören und der vermeintliche Schutz wäre dahin.

1.2 Gliederung

Eine Möglichkeit, Sicherheit für auf TCP/IP basierende IT-Systeme zu erlangen, soll die Implementierung von IPSec[1] bieten, welches im Rahmen dieser Arbeit erläutert werden soll. Hierzu werden zunächst einige Grundlagen der IP-Sicherheit erläutert. Im Anschluss folgt die Beschreibung der verschiedenen Modi und Arten von IPSec und die Erklärung der Bedingungen und Parameter, die zum reibungslosen Betrieb notwendig sind. Weiterhin soll diese Arbeit

[1] IPSecurity

die unterschiedlichen Implementations- und Anwendungsmöglichkeiten aufzeigen und auch die zukünftige Einsetzbarkeit erläutern. Im letzten Kapitel wird auf die Sicherheit von IPSec eingegangen.

1.3 Zielsetzung

Ziel dieser Arbeit soll in erster Linie sein, dem Leser einen kurzen Überblick über die Entstehung, den Aufbau und die Architektur von IPSec zu vermitteln. Weiterhin soll der Leser die unterschiedlichen Betriebsmodi von IPSec kennenlernen und anhand einer kryptographischen Evaluation die Sicherheit von IPSec einschätzen können. Auf tiefgreifende und detailgetreue Darstellung der einzelnen Verfahren wird ebenso verzichtet wie auf die kryptographischen Grundlagen, die IP-Architektur und die zur Implementierung notwendigen Details von IPSec, um den Rahmen dieser Arbeit einzuhalten.

2 Grundlagen der IP-Sicherheit

Um sichere Kommunikation über IP-basierte Netzwerksysteme erreichen zu können, müssen zuerst die Ziele definiert werden, welche durch sichere Kommunikation erreicht werden sollen, sowie die an der Kommunikation beteiligten Anwendungen und Protokolle. Zusätzlich ist es notwendig, eine geeignete Schnittstelle zu finden, in der Sicherheit am geschicktesten hinzugefügt werden kann, um am effektivsten und mit dem geringsten Aufwand Sicherheitsfunktionen anbieten zu können. Hierzu ist es notwendig zu wissen, dass sämtliche Daten, welche über ein Netzwerk übertragen werden sollen, innerhalb eines Computersystems eine bestimmte Reihenfolge von Schichten durchlaufen müssen, bevor sie über ein Übertragungsmedium, z.B. Kabel- oder Funkverbindungen, übertragen werden können.

2.1 Schutzziele

Um sichere Kommunikationskanäle nutzen zu können, muss man sich erst einmal verdeutlichen, was durch eben diese geschützt werden soll – und vor wem.
Sichere Systeme sollen vor allem vier Dinge gewährleisten:

1. Vertraulichkeit: Nur berechtigte Nutzer dürfen Zugang oder Einsicht zu den zu schützenden Daten haben
2. Integrität: Daten dürfen im Kommunikationskanal nicht unbemerkt verändert werden können

3. Verbindlichkeit: Die stattgefundene Kommunikation darf nicht abstreitbar sein

4. Verfügbarkeit: Das System muss zu einem vorgegebenen Zeitpunkt in einem funktionsfähigen Zustand sein

Im Falle einer sicheren Kommunikation kann nicht jedes dieser Ziele erreicht werden, da man als Nutzer keinen Einfluss auf die Verfügbarkeit von Kommunikationswegen hat. Es muss aber beim Nutzen von IP-Sicherheit die Vertraulichkeit der zu sendenden Daten vor unbefugten Dritten z.B. durch kryptographische Algorithmen sowie die Integrität der Daten z.B. durch Message Authentication Codes (MACs) gewährleistet sein. Sollte die Integrität der Daten nicht mehr gewährleistet sein, so muss das genutzte System dieses bemerken und entsprechend reagieren. Vor wem man die Daten schützen muss und wie hoch dementsprechend die Sicherheitsstufe gesetzt wird, hängt ganz vom Umfeld der Benutzer ab, die sicher kommunizieren wollen.
(nach [DoraHar00, S.57f], [SchlHaPo00, S.3])

2.2 Die TCP/IP-Architektur

Die im heutigen Internet verwendeten Protokolle basieren auf der TCP/IP-Familie, deren Ursprung historisch betrachtet Anfang der 60iger Jahre liegt. Die Defense Advanced Research Project Agency (DARPA) hat damals ein Projekt ins Leben gerufen, welches Universitäts- und Forschungsnetze miteinander verbinden sollte. Dieses Projekt wurde ARPANET genannt und stellt den Vorgänger des heutigen Internets dar. 1983 wurden die damaligen Kommunikationsprotokolle durch die heute bekannte TCP/IP-Familie ersetzt, da diese einfacher zu benutzen, zu implementieren und zu erweitern war. Diese Protokollansammlung setzt eine spezielle Architektur voraus, um Kommunikation zu ermöglichen. Hierzu gibt es einen standardisierten Protokollstapel aus vier Ebenen, wovon jede eine genau festgelegte Funktion und Fähigkeit innerhalb des Kommunikationsprozesses besitzt. Durch diese Schichtenarchitektur ist gewährleistet, dass Daten nur über vordefinierte Schnittstellen ausgetauscht werden können und somit die Kommunikationsabfolge hierarchisch strukturiert ist. Jede Schicht kennt nur die Darunter- und die Darüberliegende und somit sind sie einzeln und unabhängig implementierbar.

Die **Anwendungsschicht** (application layer) hat die Aufgabe, im Auftrag von Anwendungen Daten über das Netzwerk zu versenden oder zu empfangen. Hierzu bedienen sich die Anwendungen, z.B. eMail-Programme oder Browser, speziellen Diensten innerhalb der Anwendungsschicht, um mit ihren Partnern zu

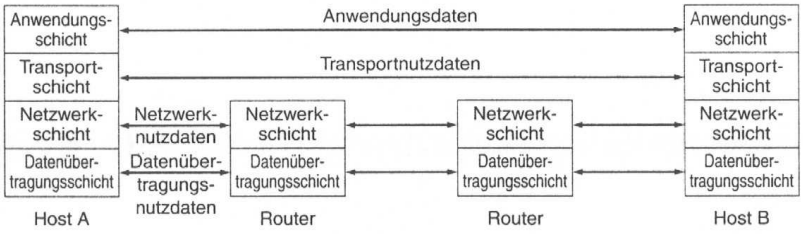

Abbildung 1: Die vier Schichten des TCP/IP-Modells und die Kommunikation zwischen zwei Hosts, [DoraHar00, S.38]

kommunizieren. Die Anwendungsschicht nimmt demnach Aufträge entgegen, verarbeitet diese und leitet im Anschluss daran die nächsten Schritte ein. Hierzu spricht sie die Transportschicht (transport layer) über eine betriebssystemabhängige Schnittstelle an (z.B. Socket-Interface).

Die **Transportschicht** versorgt die Anwendungsschicht mit verschiedenen Diensten, aus der diese wählen muss. So ist die Transportschicht z.B. dafür zuständig, Verbindungen zum Kommunikationspartner aufrecht zu erhalten, sollte ein verbindungsorientierter Transport gewünscht sein. Soll die Kommunikation verbindungslos stattfinden, wird für jedes zu übertragene Paket ein eigener Bestimmungsort angegeben. Weiterhin sorgt die Transportschicht für den zuverlässigen Transport von Datenpaketen zum Kommunikationspartner. Sollte ein Paket unterwegs verloren gehen oder fehlerhaft ankommen, so sendet es die Transportschicht erneut, wenn die Kommunikation zuverlässig erfolgen soll. Soll die Kommunikation unzuverlässig sein, müssen sich die Anwendungen um fehlerhafte Übertragungen kümmern. In neueren TCP/IP-Implementierungen werden zusätzlich noch Sicherheitsmerkmale in die Transportschicht integriert.

Die **Netzwerkschicht** (network layer) hat lediglich die Aufgabe, Pakete zu routen. Hierbei legt sie den Weg fest, den ein Paket über das Netzwerk zurücklegen muss, um zu seinem Ziel zu gelangen. Sie leistet also verbindungslose Dienste, und muss somit in der Lage sein, die zu versendenden Pakete mit einer eindeutigen Adresse zu versehen, welche im Allgemeinen als IP-Adresse bekannt ist.

Die letzte Schicht im TCP/IP-Stapel ist die **Datenübertragungsschicht** (data link layer), die für die physikalische Übertragung zwischen zwei Hosts zuständig. Beispiele für Datenübertragungsschichten sind Ethernet oder Token Ring.
(nach [DoraHar00, S.35-39])

2.3 Einbindung von Sicherheit in den Protokollstapel

Um nun Sicherheitsmechanismen in ein vorgegebenes Stapelsystem zu integrieren, muss man sich überlegen, an welcher Stelle dies besonders sinnvoll ist. Versucht man z.B. Sicherheitsmaßnahmen in der Anwendungsschicht bereitzustellen, so fällt auf, dass diese in die Endgeräte implementiert werden müssen. Der Nachteil liegt auf der Hand: Jede Anwendung muss Sicherheitsmechanismen eigenständig bereitstellen, und die Entwickler müssen jeweils ihren eigenen Sicherheitsmechanismus definieren. Dies führt natürlich bei immer komplexer werdenden Netzwerken und immer mehr Interaktionen zwischen unterschiedlichen Anwendungen zu einem Problem, da ohne geregelte Standards und Qualitätskontrolle eine sichere Kommunikation untereinander nicht gewährleistet ist. Außerdem müssen sich die Anwendungen selbst um das Management wie z.B. Schlüsselaustausch kümmern. Ein funktionierendes Beispiel für Sicherheit in der Anwendungsschicht sind Programme wie z.B. PGP. Allerdings ist nicht davon auszugehen, dass bei der Implementierung von Sicherheit in jede Anwendung die entsprechenden Programmierer das dazu nötige Grundwissen haben, um Sicherheit auch wirklich zu gewährleisten.
Der Hauptvorteil von Sicherheit in der Transportschicht liegt darin, dass Anwendungen nicht explizit erweitert werden müssen. Beispiel für Sicherheit in der Transportschicht ist TLS (Transport Layer Security). Hierbei wird aber von einem Benutzerkontext ausgegangen, der ein System mit einem einzelnen Benutzer identifiziert und somit nicht allgemeingültig ist und damit für generelle IP-Sicherheit unpraktisch ist.
Als vorteilhafteste Schicht für die Implementierung von Sicherheit innerhalb des TCP/IP-Schichtenmodells ist die Netzwerkschicht anzusehen. Sicherheitsimplementierungen an dieser Stelle erfordern nur geringfügige Änderungen im TCP/IP-Protokoll und kaum Änderungen an Anwendungen oder den darüber liegenden Schichten. Somit ist eine explosionsartige Implementation von Sicherheitsprotokollen in den darüber liegenden Schichten eingedämmt und die neu installierten Funktionen können effektiv jeder Anwendung transparent und problemlos zur Verfügung gestellt werden. Mit der Netzwerkschicht können weitere nützliche Funktionen erfüllt werden, wie z.B. die Einrichtung von virtuellen, privaten Netzwerken (VPNs) und Intranets. Jedoch bietet die Netzwerkschicht nicht nur Vorteile, da der Umgang mit Problemfällen in dieser niedrigen Schicht komplexer ist und weiter oben in der Hierarchie-Struktur besser gelöst werden könnte. Mit IPSec wird Sicherheit genau in dieser Schicht gewährleistet, da Sicherheit sowohl pro Datenfluss als auch pro Verbindung möglich ist. Mit IPSec ist eine sehr feine Abstufung der Sicherheitskontrollen möglich.
Die Implementierung von Sicherheit in der Datenübertragungsschicht kommt

nur in ganz speziellen Bereichen in Frage, da spezielle Verschlüsselungshardware sowie physikalische Verbindungen zwischen zwei Hosts notwendig sind, um den Sicherheitskontext vernünftig aufrecht zu erhalten.
Da die Anwendung von IPSec in der Regel entweder in Endsystemen oder in Sicherheitsgateways (wie z.B. Firewalls oder Routern) erfolgt, stellt sich die Frage, wie die Implementierung am besten erfolgen sollte. Typischerweise geschieht dies durch eine direkte Modifikation des IP-Stapels im Betriebssystem. Ist dies nicht möglich, so kann IPSec auch über Bump in the Stack (BITS) dem Stapel übergeordnet werden oder mittels Bump in the Wire (BITW) als Zwischenstück direkt auf den Übertragungskanal implementiert werden. Somit ist die Implementierung in jedem Szenario möglich.
(nach [DoraHar00, S.53-58,60])

3 IPSec

IPSec ist eine Ansammlung von Standards, die 1998 von der Internet Engeneering Task Force (IETF) verabschiedet wurde und eine Vielzahl von Protokollen enthält, die eine auf TCP/IP-basierende Kommunikation über unsichere Netzwerke, wie z.B. das Internet, sichern soll. Diese Protokollsammlung ist in zwölf Request for Comment (RFCs) veröffentlicht worden, welche die verschiedenen Aspekte von IPSec beleuchten, wie z.B. die Architektur, die Schlüsselverwaltung und die Basisprotokolle. IPSec bedient sich einer Implementierung in die dritte Schicht der TCP/IP-Architektur und sichert die einzelnen IP-Pakete auf der Netzwerkschicht ab, indem es die ursprünglichen IP-Pakete mit neuen Eigenschaften einkapselt und somit den ursprünglichen Inhalt entweder verschlüsselt oder authentifiziert oder beides. Hierzu bedient sich IPSec unterschiedlichen Modi und Protokollvarianten, die im Folgenden näher erläutert werden sollen.
(nach [DoraHar00, S.75], [SchlHaPo00, S.4])

3.1 Die Architektur von IPSec

IPSec besteht aus einer genau definierten Basisarchitektur, auf der alle Implementierungen aufgebaut sind. Diese Architektur legt die bereitgestellten Sicherheitsdienste und die Reihenfolge der Abarbeitung von IP-Paketen fest. Es gibt zwei Basisprotokolle, mit denen IPSec betrieben werden kann: Authentication Header (AH) und Encapsulating Security Payload (ESP). Diese Basisprotokolle von IPSec werden in den folgenden Kapiteln näher erläutert. Zusätzlich können diese beiden Protokollvarianten in zwei unterschiedlichen Modi betrieben werden: dem Transport-Modus und dem Tunnel-Modus. Grundlegend lässt sich

sagen, dass der Transport-Modus benutzt wird, um übergeordnete Protokollschichten zu schützen, während der Tunnel-Modus für komplette IP-Pakete zuständig ist. Wie Abbildung 2 zeigt, wird im Transport-Modus ein IPSec-Header zwischen den IP-Header und den TCP-Header eingefügt. Der Tunnel-Modus hingegen umschließt das ursprüngliche Paket und fügt dem neuen Paket einen eigenen IP-Header plus IPSec-Header hinzu.

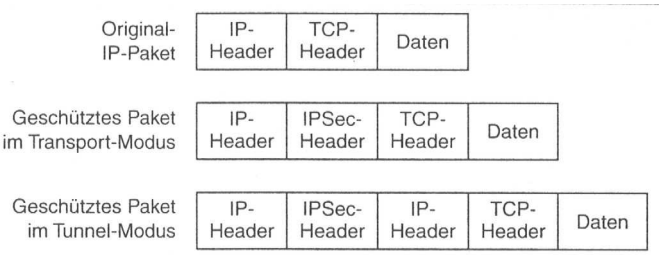

Abbildung 2: Der Unterschied zwischen Transport- und Tunnelmodus, [DoraHar00, S.59]

Wegen der Art der Erzeugung von IP-Paketen kann der Transport-Modus nur zum Schutz von Paketen angewandt werden, bei denen der Kommunikationsendpunkt auch das endgültige Ziel des IP-Paketes ist. Beim Tunnel-Modus kann Sicherheit auch für andere Netzwerk-Entitäten gewährleistet werden, wenn diese nicht direkter Kommunikationsendpunkt sind. Beispiele hierfür sind z.B. Intranets, die über einen Router oder eine Firewall mit dem Internet verbunden sind und sicher mit Rechnern auf der anderen Seite kommunizieren wollen. Hierzu baut das Gateway einen gesicherten Tunnel zum Kommunikationspartner (oder dessen Gateway) auf und tunnelt die zu schützenden Pakete somit durch das Internet.
Einen Gesamtüberblick über die Architektur von IPSec liefert Abbildung 3.
(nach [DoraHar00, S.59ff, 75ff, 81-86], [SchlHaPo00, S.4], [Wichmann99, S.11f])

3.2 Encapsulating Security Payload (ESP)

Das ESP-Protokoll in IPSec ist für Vertraulichkeit, Datenintegrität und Authentifizierung der Datenquellen von IP-Paketen zuständig. Es kann sämtliche an IPSec gestellten Forderungen zur Sicherung von IP-Paketen erfüllen, da es sowohl die Daten vor dem Übertragen verschlüsseln und somit die Informationen vor unbefugten Dritten schützen, als auch dafür sorgen kann, dass die Authentizität und Integrität gewährleistet ist. Weiterhin hindert es durch bestimmte Eigenschaften einen Angreifer daran, eine sog. Replay-Attacke ausführen zu können,

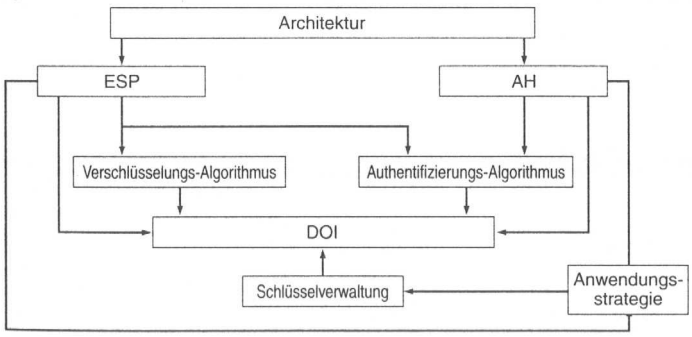

Abbildung 3: Die Gesamtarchitektur von IPSec, [DoraHar00, S.75]

in dem er abgefangene Pakete erneut sendet. ESP erreicht dies, in dem ein spezieller ESP-Header im Anschluss an den IP-Header eingefügt und ein spezieller ESP-Trailer am Ende des Paketes angefügt wird. Header und Trailer enthalten signifikante Informationen über das IP-Paket, verwendete IPSec-Parameter und kryptographisch notwendige Informationen, wie z.B. den Initialisierungsvektor von CBC-verschlüsselnden Algorithmen.[2]
Vertraulichkeit wird im ESP durch das Verschlüsseln des ursprünglichen IP-Paketes gewährleistet. Da IPSec für diesen Fall mehrere Algorithmen vorgesehen hat, müssen die kommunizierenden Partner sich auf einen bestimmten Algorithmus einigen, was über Sicherheitsassoziationen geschieht, die etwas später beschrieben werden. Der verwendete kryptographische Algorithmus zur Verschlüsselung wird Cipher oder Encryptor genannt, der zur Gewährleistung der Authentizität heißt Authentifikator.
Der ESP-Header wird immer vor dem Header mit den Zieloptionen in das Paket eingefügt, um diese Zielinformationen in den Schutz mit einzubeziehen. Grundsätzlich folgt der ESP-Header auf den letzten zur Paket-Übertragung notwendigen IP-Header, aber noch vor dem ursprünglichen IP-Header. Er enthält einen sog. Sicherheitsparameterindex (SPI), welcher zusammen mit der Zieladresse und dem Protokoll des vorangegangenen IP-Headers die Sicherheitsassoziation festlegt, nach der das Paket verarbeitet wird. Einzelheiten über SPI und Parameter folgen in Kapitel 4. Weiterhin enthält der ESP-Header eine für das Paket eindeutige Seriennummer, mit der die Eindeutigkeit des Paketes festgelegt wird und ein wiederholtes Senden ausschließt (anti-replay). Die Daten des ESP-Headers sind nicht verschlüsselt, um eine Identifikation und die Auswahl der richtigen Sicherheitsassoziation für den Empfänger zu ermöglichen. Der

[2] An dieser Stelle wird auf die Erläuterung des Cipher-Block-Chaining-Modes (CBC) für symmetrische Algorithmen verzichtet, da es den Rahmen der Arbeit sprengen würde. Dieser wird als bekannt vorausgesetzt, ebenso wie das grundlegende Verständnis vom standardmäßig in IPSec verwendeten Algorithmus DES (Data Encryption Standard) bzw. Triple-DES.

ESP-Trailer enthält neben für manche kryptographische Algorithmen notwendige Fülldaten weitere Authentifizierungsdaten. Nachfolgend werden die zwei möglichen Modi im ESP erläutert.
(nach [DoraHar00, S.63-66, 101-108], [SchlHaPo00, S.4], [Wichmann99, S.11f])

3.2.1 ESP im Tunnelmodus

Wird IPSec mit ESP im Tunnelmodus angewandt, so wird das gesamte zu schützende IP-Paket von ESP umschlossen und es kommt ein weiterer IP-Header hinzu. Abbildung 4 zeigt den neuen Aufbau des Paketes, wobei die Daten zwischen den beiden IP-Headern ESP-Header genannt werden.

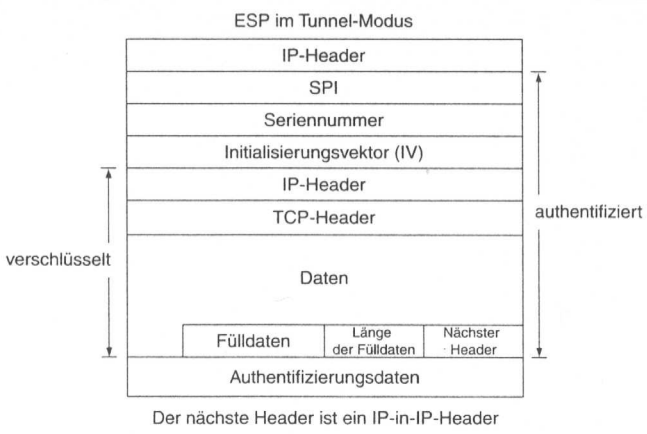

Abbildung 4: Der detaillierte Aufbau des Paketes mit ESP-Header und -Trailer im Tunnel-Modus, [DoraHar00, S.105]

3.2.2 ESP im Transportmodus

Handelt es sich um den Transport-Modus, so wird der ESP-Header zwischen den IP-Header und den der darüber liegenden Schicht (z.B. TCP-Header) eingebettet, sodass sich ein Aufbau nach Abbildung 5 ergibt.

3.3 Authentication Header (AH)

Das AH-Protokoll sorgt in IPSec für Datenintegrität, Authentifizierung von Datenquellen und bietet Anti-replay-Funktionalität. Prinzipiell gesehen ist AH nichts anderes als ESP, es fehlt lediglich die Vertraulichkeit der Dateninhalte, da IPSec im AH-Modus betrieben nicht verschlüsselt. AH verfügt somit nur

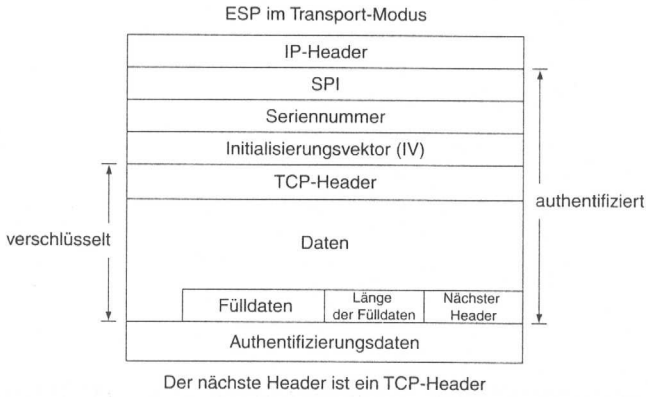

Abbildung 5: Der detaillierte Aufbau des Paketes mit ESP-Header und -Trailer im Transport-Modus, [DoraHar00, S.104]

über einen Authentikator, der Encryptor fehlt. Typischerweise werden als Authentifikator HMAC-SHA-96 oder HMAC-MD5-96 genutzt. Der Unterschied zu ESP liegt vor allem in der Anwendung. Muss keine Vertraulichkeit gewährleistet werden, sondern lediglich dafür gesorgt werden, dass Pakete integer sind und unterwegs nicht verändert wurden, so reicht IPSec mit AH vollkommen aus. Der Vorteil von AH im Vergleich zu ESP liegt in der Simplifizierung. Der AH-Header ist ein einfacher Header, der nicht noch zusätzlich einen Trailer braucht und im Klartext übermittelt werden kann. Somit ist AH ein allgemein anwendbarer Sicherheitsdienst für IP. Ein durch AH-geschütztes IP-Paket ist ebenfalls ein normales IP-Paket. Der AH-Header wird dem ursprünglichen IP-Header angefügt. Die durch AH gewährleistete Datenintegrität unterscheidet sich grundlegend von der Integrität in ESP, da AH auch äußere Teile des IP-Headers authentifizieren kann. AH kommt im Transport-Modus mit nur einem IP-Header aus, was den Overhead deutlich minimiert. Der AH-Header enthält ebenfalls einen Sicherheitsparameterindex (SPI), eine Seriennummer und Authentifizierungsdaten. Im Folgenden werden die einzelnen Modi näher erläutert. (nach [DoraHar00, S.63, 66-68, 109-116], [SchlHaPo00, S.4f], [Wichmann99, S.12])

3.3.1 AH im Tunnelmodus

Wird AH im Tunnelmodus genutzt, so wird das gesamte ursprüngliche IP-Paket geschützt. Es wird ein neuer IP-Header erzeugt, der vor den AH-Header eingefügt wird. Das innere IP-Datagramm enthält die Originaladressen der Kommunikationspartner, das äußere IP-Datagramm die Adressen der IPSec-Endpunkte. Abbildung 6 zeigt den Aufbau von AH im Tunnelmodus.

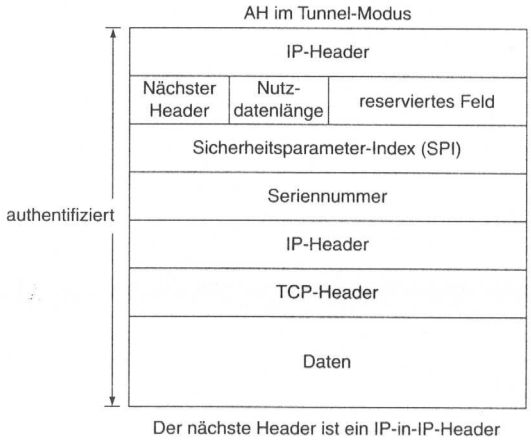

Abbildung 6: Der detaillierte Aufbau des Paketes mit AH-Header im Tunnel-Modus, [DoraHar00, S.112]

3.3.2 AH im Transportmodus

AH im Transportmodus schützt die Kommunikation von Endstelle zu Endstelle, somit ist jeder Kommunikationspartner gleichzeitig IPSec-Endpunkt. Der AH-Header wird nach dem IP-Header und vor dem Header der übergeordneten Schicht eingefügt, wie Abbildung 7 verdeutlicht.

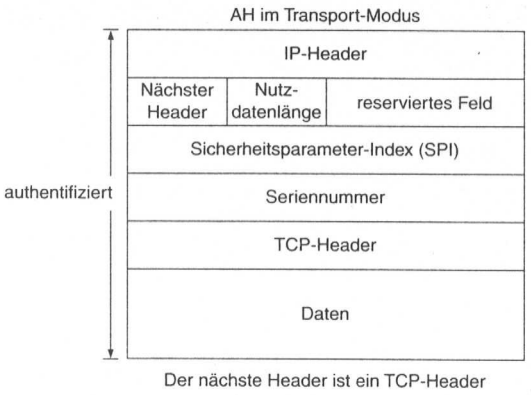

Abbildung 7: Der detaillierte Aufbau des Paketes mit AH-Header im Transport-Modus, [DoraHar00, S.112]

3.4 ESP / AH – Tunnel / Transport?

Um nun zu entscheiden, welches IPSec-Protokoll für welche Anwendung am zutreffendsten ist, muss man sich den Anwendungsrahmen anschauen und die Sicherheit definieren, welche erreicht werden soll. Wird IPSec beispielsweise genutzt, um ein virtuelles, privates Netz aufzubauen, so muss auf jeden Fall ESP verwendet werden, da die Vertraulichkeit der Daten oberste Priorität hat. Authentifikation ist in jedem Fall gewährleistet, es sei denn man wählt im ESP-Protokoll als Authentikator den Null-Authentikator (was in der Praxis nicht gemacht wird, aber durch die Definitionen möglich sein soll, um Implementationen auf Korrektheit zu überprüfen). Soll die Kommunikation lediglich von Host zu Host stattfinden und gewährleistet sein, dass die Daten korrekt und ohne Manipulation ihr Ziel erreichen, reicht der AH-Modus aus, da dieser den geringsten Overhead bietet und mit einem IPSec-Header ohne Trailer auskommt. In den IPSec-Standards ist auch eine Kombination der beiden Protokolle vorgesehen, um geschachtelte oder verkettete Kommunikation zu ermöglichen, z.B. um durch ein bereits existierendes, durch IPSec geschütztes, VPN einen weiteren Tunnel zu einem im Intranet liegenden Server aufzubauen. Ein weiteres Szenario wäre z.B. die sichere Kommunikation eines externen Mitarbeiters mit Firmenservern über das Internet. Hierzu würde er einen ESP-Tunnel aufbauen, der durch verschiedene Internet-Router geschleust wird, welche wiederum untereinander IPSec im AH zur Authentifizerung der Daten nutzen können. In jedem Falle sehen die Standards vor, dass AH und ESP in Kombination immer so geschachtelt werden, dass AH das ESP-Paket authentifiziert, da somit auch äußere Teile des IP-Headers mit gesichert werden, was im umgekehrten Falle nicht gewährleistet ist. Ausserdem macht die Verschlüsselung von authentifizierten Daten in keinem Szenario Sinn. Die Wahl des entsprechenden Modus (Transport- oder Tunnel-Modus) hängt nur davon ab, an welcher Stelle man im Kommunikationsnetz ist und welche Funktion man dabei übernimmt. Ist man z.B. IPSec-Endpunkt, so macht der Transport-Modus Sinn, andernfalls muss der Tunnel-Modus verwendet werden. Abbildungen 8 und 9 verdeutlichen den Aufbau der Pakete in beiden Modi.
(nach [DoraHar00, S.188-195], [SchlHaPo00, S.4f], [Wichmann99, S.11ff])

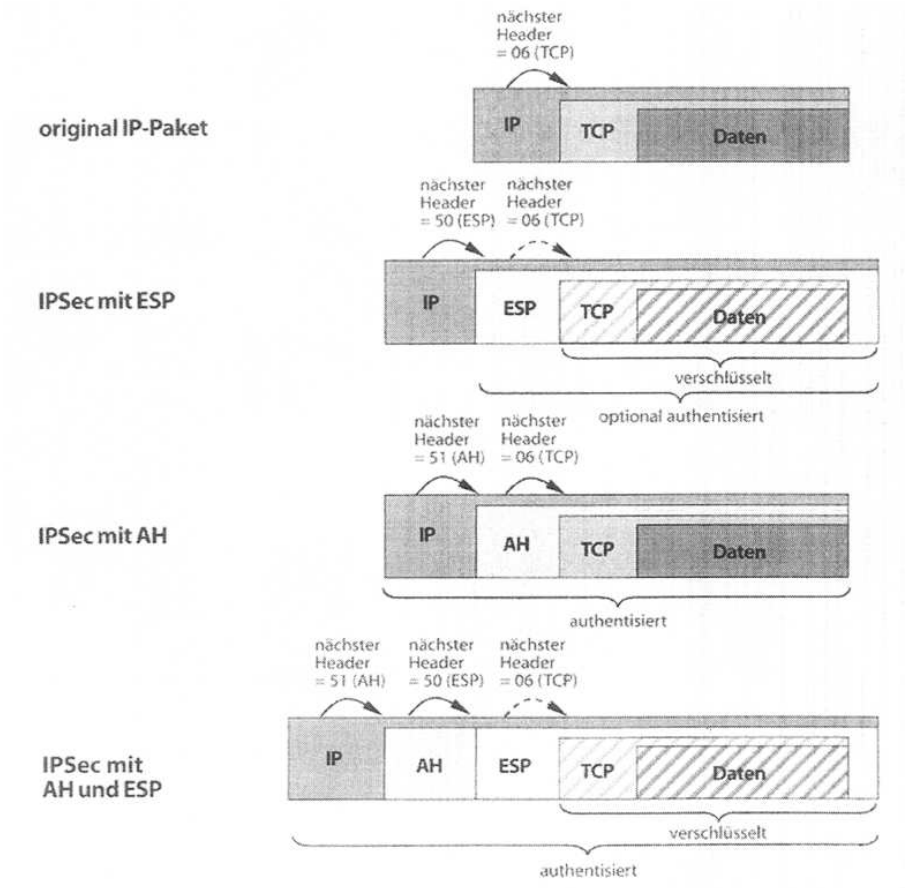

Abbildung 8: Überblick über IPSec im Transport-Modus, [Wichmann99, S.13]

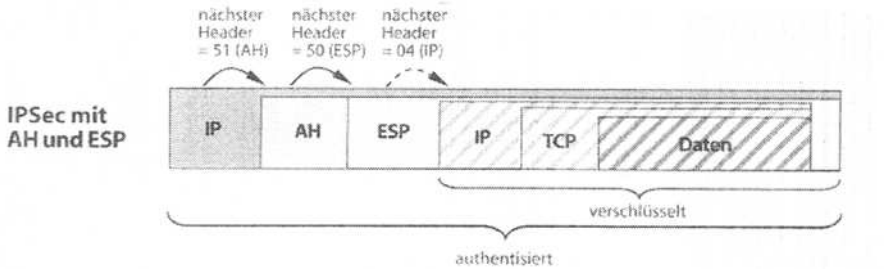

Abbildung 9: Überblick über IPSec im Tunnel-Modus, [Wichmann99, S.14]

4 Sicherheitsassoziationen

Um nun sicher über IPSec kommunizieren zu können, müssen sich die beiden Teilnehmer auf verschiedene Parameter einigen, die für den Aufbau und die Erhaltung der Verbindung notwendig sind. Dazu gehören u.a. die zu verwendenden kryptographischen Algorithmen, deren Initialisierungsvektoren und die zu benutzenden kryptographischen Schlüssel. Um für den Gebrauch von IPSec notwendige Schlüssel zu erzeugen, gibt es spezielle RFCs, welche detaillierte Protokolle für den Schlüsselaustausch definieren. Diese lehnen sich häufig an den Diffie-Hellman-Schlüsselaustausch an. Auf eine detaillierte Erklärung des Schlüsselaustausches (und natürlich der Erzeugung kryptographisch sicherer Schlüssel) wird im Rahmen dieser Arbeit allerdings verzichtet. Es sei lediglich darauf hingewiesen, dass in den IPSec-Standards zwei Protokolle vorgesehen sind: Internet Key Exchange (IKE) und Internet Security Association and Key Management Protocol (ISAKMP).
(nach [DoraHar00, S.68-73, 86], [SchlHaPo00, S.4f], [Wichmann99, S.11])

4.1 SA und SADB

Sicherheitsassoziationen (SAs) definieren exakt, welcher IP-Verkehr durch IPSec geschützt werden soll und vor allem wie. Diese Sicherheitsassoziationen bestehen immer nur in eine Richtung und sind eindeutig durch einen Index identifizierbar (siehe Kapitel 4.2). Ist die Kommunikationsbasis bi-direktional (was eigentlich immer der Fall ist), so muss jeder an der Kommunikation beteiligte Host pro Verbindung zwei Sicherheitsassoziationen speichern, eine für den ausgehenden und eine für den eingehenden Verkehr. SAs stellen somit die Basis für die IPSec-Kommunikation dar und bilden den Vertrag zwischen den Kommunikationspartnern.
Da gerade bei Servern immer mehrere Verbindungen gleichzeitig aufrecht erhalten werden, nimmt die Anzahl der zu verwaltenden SAs linear mit der Anzahl der Verbindungen zu. Um den Verwaltungsaufwand zu minimieren, werden sämtliche SAs in einer speziellen Sicherheitsassoziations-Datenbank (SADB) gespeichert. An diese werden entsprechend hohe Anforderungen gestellt, da die Systemleistung vor allem von der Geschwindigkeit des Datenbankzugriffs abhängt. Dabei ist anzumerken, dass das Auffinden von Parametern für eingehende Verbindungen viel einfacher ist als für ausgehende, da die im empfangenen Paket enthaltenen Informationen exakt mit denen in der Datenbank übereinstimmen. In der Praxis werden für die Implementation von SADBs oft Hash-Listen oder Bäume verwendet.
(nach [DoraHar00, S.60ff, 86-90, 163-167], [SchlHaPo00, S.4f])

4.2 SPI und SPD

Der Security Parameter Index (SPI) ist eines der wichtigsten Elemente innerhalb der SA, da er die Kommunikationspartner eindeutig identifiziert und kennzeichnet. In ihm werden Parameter wie Schlüssel oder Algorithmen übertragen. Jede Kommunikationsrichtung in jeder Verbindung hat einen eigenen SPI, welcher im IPSec-Header mit übertragen wird. Auf die im SPI enthaltenen Informationen kann der Empfänger nur teilweise zugreifen, da Teile der enthaltenen Selektoren zur Transportschicht gehören. Eine weitere Komponente innerhalb IPSec ist die Security Policy Database (SPD), welche eine Datenbank für Sicherheitsstrategien definiert. Diese arbeitet eng mit der SADB zusammen und ist für die Weiterverarbeitung der IPSec-Pakete zuständig. Sie definiert die weitere Vorgehensweise für die einzelnen Pakete, die durch sog. Selektoren festgelegt wird (siehe 4.2.2).
(nach [DoraHar00, S.60ff, 87f, 93], [SchlHaPo00, S.4f], [Wichmann99, S.11])

4.2.1 Parameter

Um nun jedes einzelne Paket richtig zu verarbeiten, hat der IPSec-Header verschiedene Felder, die als Parameter genutzt werden. Diese enthalten eine eindeutige Seriennummer, die pro Paket um 1 erhöht wird, um einen anti-replay-Schutz zu gewährleisten. Kurz vor dem Überlauf der Seriennummer muss die SA neu aufgebaut werden, um Fehler zu vermeiden. Hierzu gibt es ein spezielles Seriennummernüberlauf-Feld, welches vor einem Überlauf gesetzt wird. Zur Überprüfung der Seriennummern und deren Gültigkeit gibt es ein Fenster, in das die Seriennummern passen müssen. Dieses bewegt sich dynamisch beim Empfänger mit der Anzahl der empfangenen Pakete mit. Enthält der Host ein Paket, dessen Seriennummer nicht mehr in das Fenster passt, wird das Paket verworfen. Weitere Parameter sind Lebensdauer, der Modus, in dem IPSec betrieben wird und natürlich der Zielort.
(nach [DoraHar00, S.91f])

4.2.2 Selektoren

Die Art, in der empfangene Pakete verarbeitet werden, hängt von der verwendeten Sicherheitsstrategie ab und sieht drei Möglichkeiten vor: Lass das Paket fallen (discard), Umgehe die Sicherheit (bypass) oder wende Sicherheit an (apply). Um nun entscheiden zu können, welche Möglichkeit für welches Paket passt, gibt es bestimmte Selektoren, die eine Weiterverarbeitung exakt festlegen. Diese Selektoren werden durch die Ursprungsadresse, die Zieladresse, das verwendete

Protokoll und die Ports definiert. Bei der Definition der Sicherheitsstrategien muss anhand der Selektoren exakt festgelegt werden, welche Art von Verkehr durch IPSec geschützt wird. Somit lässt sich beispielsweise festlegen, dass alle Pakete, die nicht IPSec-geschützt sind, fallen gelassen werden. Dies würde dazu führen, daß der Aufbau einer Kommunikation mit einem nicht in der SADB-gespeicherten Kommunikationspartner scheitern, und so der Host für sämtliche nicht autorisierte Hosts unerreichbar würde.
(nach [DoraHar00, S.94-98])

4.3 Erzeugen und Löschen von SAs

Da Sicherheitsassoziationen keine festgelegte Lebensdauer haben, müssen sie explizit erzeugt und gelöscht werden. Hierzu gibt es bei IPSec zwei Möglichkeiten: die dynamische Erzeugung von Regeln durch die Hosts selber und die manuelle Zuweisung durch den Administrator. Werden SAs dynamisch erzeugt, muss sich der Host auch selbst um die Lebensdauer, d.h. die Löschung der SA kümmern, während manuell erzeugte SAs auch nur manuell wieder entfernt werden können. Protokolle wie ISAKMP und IKE unterstützen die dynamische Erzeugung und Löschung von SAs. Gründe für das Löschen von SAs sind z.B. die Kompromittierung eines Sitzungsschlüssels, abgelaufene Lebensdauer oder ein übermäßiger Gebrauch der Parameter, wodurch evtl. eine Aufdeckung des Schlüssels durch Kryptanalyse möglich wäre.
(nach [DoraHar00, S.89f])
Zusammenfassend zeigt Abbildung 10 das Zusammenspiel der IPSec-Module.

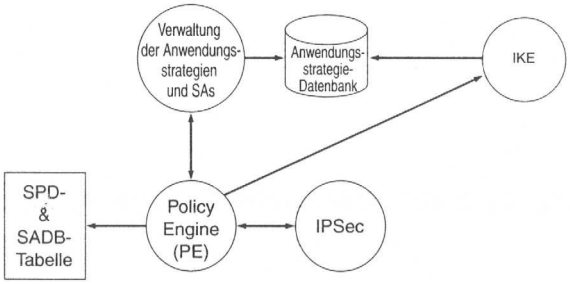

Abbildung 10: Die IPSec-Module und die Zusammenhänge, [DoraHar00, S.162]

5 Die Sicherheit von IPSec

Um die Sicherheit von IPSec zu bewerten, muss man sich erst mal klar machen, dass Sicherheit ein Prozess ist, der sich durch sämtliche an der Kommunikati-

on beteiligten Bereiche zieht. Wenn bereits die Implementierung fehlerhaft ist, nutzt das gesamte System nichts. Weiterhin hängt die Sicherheit vorwiegend von den zur Authentifizierung oder Verschlüsselung verwendeten kryptographischen Algorithmen ab. Wenn diese Algorithmen gebrochen werden ist natürlich auch IPSec nicht mehr sicher. Um diesen Fall aber auszuschließen, bedient sich IPSec einer Modularisierung, sodass Algorithmen jederzeit hinzugefügt werden können. Für die Parameter dazu bietet IPSec genügend Platz. Der standardmäßig implementierte Algorithmus DES ist auf Grund seiner kurzen Schlüssellänge nicht mehr zeitgemäß. Sollen Daten auch für die Zukunft beständig bleiben, so muss man auf andere symmetrische Verschlüsselungsverfahren ausweichen. Alternativen sind z.B. Triple-DES oder AES. Der Einsatz von symmetrischen Verfahren erfolgt hauptsächlich aus Effizienzgründen. Durch die Verkettung der ausgehenden Pakete mittels CBC wird die Sicherheit der Algorithmen noch erhöht. Durch die Authentifizierung, welche über keyed mac-functions realisiert wird, sind die Inhalte sowie die Pakete selber vor Manipulation geschützt, Verzögerungen werden durch das Fenster-System der Seriennummern erkannt und somit besteht auch ein Schutz vor erneutem Senden. IPSec ist momentan der bestmögliche Schutz für IP-basierte Kommunikation, allerdings hat er auch gravierende Nachteile. Wie Niels Ferguson und Bruce Schneier in [SchnFerg00] beschreiben, ist der größte Feind der Sicherheit Komplexität. Durch das Zusammenwirken eines gesamten Komitees wurden viele Features in IPSec eingebaut, was auf Kompromissbereitschaft zurückzuführen ist. Allerdings ist eben diese große Optionalität eine Schwachstelle. Je mehr Optionen man hat und je flexibler man IPSec gestalten kann, umso größer ist der Verwaltungs- und Implementierungsaufwand. In komplexen Systemen treten immer mehr komplexe Fehler auf, das Zusammenspiel untereinander ist sehr schwer zu analysieren und es kann nicht ausgeschlossen werden, dass in Interaktion neue Schwachstellen auftauchen. Nach Schneier und Ferguson ist IPSec zu komplex, um sicher zu sein. Ein weiterer Nachteil in IPSec ist die Bereitstellung verschiedener Modi, obwohl die gleichen Ziele auf verschiedene Weisen erreicht werden können. Authentizität ist sowohl im ESP als auch im AH-Modus möglich, warum also das AH-Protokoll nicht gänzlich weglassen? Der Overhead, der durch das ESP-Protokoll entsteht, lässt sich durch geschickte Kompressionsmethoden, wie das Payload Compression Protocol (PCP), neutralisieren. Die Komplexität würde bereits um 50 % sinken. Des weiteren ist die Unterteilung in Tunnel- und Transport-Modus auch unnötig, da die Funktionen des Transport-Modus vollständig durch den Tunnel-Modus übernommen werden können. Somit würde die Sicherheit erneut erhöht, der Overhead hingegen nur ein bisschen größer werden, da im Tunnel-Modus ein zweiter IP-Header angefügt wird. Zusammenfassend lässt sich also festhalten, dass IPSec bei korrekter Implementierung das

zur Zeit beste Protokoll zur Sicherung von IP-Kommunikation ist, allerdings auf Grund der hohen Komplexität und Flexibilität diese korrekte Implementierung nicht wirklich gelingt. Somit wäre eine Reduzierung der Optionen durch z.B. Elimination des AH und des Transport-Modus eine empfehlenswerte Vereinfachung von IPSec und somit als Steigerung der Sicherheit zu empfehlen.
(nach [DoraHar00, S.93, 197-201], [SchnFerg00, S.1-27])

6 Fazit

Im Rahmen dieser Arbeit wurde die Problematik einer sicheren IP-Kommunikation beleuchtet und darauf aufmerksam gemacht, dass diese nicht von vornherein gegeben ist. Um dennoch sicher über das TCP/IP-Protokoll kommunizieren zu können, muss Sicherheit explizit implementiert werden. Hierzu wurde IPSec mit seinen verschiedenen Möglichkeiten, Modi und Parametern erläutert und anhand von Beispielen die Einsatzmöglichkeiten aufgezeigt. Aus kryptographischer Sicht liegen die Nachteile von IPSec eindeutig in der hohen Komplexität der Protokoll-Familie, welche auf den Entstehungsprozess zurückzuführen ist. Vorteile von IPSec sind die Flexibilität im Austausch von unsicher werdenden kryptographischen Algorithmen und die Einordnung in den TCP/IP-Stapel. Anwendungen müssen nicht gesondert erweitert werden, um Sicherheitsfunktionalität anbieten und nutzen zu können, sondern Sicherheit wird auf einer tieferen Ebene realisiert. Im momentanen Stadium von IPSec ist es jedoch auf Grund der gravierenden Nachteile nicht bedingungslos zu empfehlen, es wird eher darauf hingewiesen, die Komplexität von IPSec zu minimieren. Der Grundgedanke und das Konzept von IPSec sind aber auf jeden Fall schon weit mehr als ein Schritt in die richtige Richtung, Sicherheit nahezu überall verfügbar zu machen, ohne eine explosionsartige Zunahme von Anwendungserweiterungen heraufzubeschwören. Durch die Erweiterbarkeit von IPSec ist es für die Zukunft auch in anderen Szenarien denkbar, momentan werden z.B. Protokolle für Multicast-Umgebungen entwickelt.

Abbildungsverzeichnis

Literatur

[DoraHar00] Doraswamy, Naganand / Harkins, Dan: IPSec, der neue Sicherheitsstandard für das Internet, Intranets und virtuelle private Netze, Addison-Wesley, 2000

[SchlHaPo00] Schlichting, Jochen / Hartmann, Jörg / Pohl, Hartmut: Virtual Private Networks auf IPSec-Basis, Lessing und Partner, 2000

[SchnFerg00] Schneier, Bruce / Ferguson, Niels: A Cryptographic Evaluation of IPSec, http://www.counterpane.com, 2000

[Wichmann99] Wichmann, Michael: IPSec in der Satelitenkommunikation, EIT 99