IPSec Architektur und Protokolle, Internet Key Exchange (IKE)

Wolfgang Thomas
(thomasw@in.tum.de)

Hauptseminar: **Sicherheit in Kommunikationsnetzen**
Technische Universität München

WS 2002/2003 (Version 8. Februar 2003)

Zusammenfassung

Dieses Papier behandelt die grundlegende Architektur der Sicherheitsprotokollsuite IPSec. Erläutert werden die grundlegenden Protokolle *Encapsulating Security Payload (ESP)* und *Authentication Header (AH)*, sowie deren Einsatz im Rahmen von IPSec. Das Prinzip der Sicherheitsassoziation und ihr Zusammenhang mit dem Schlüsselaustauschprotokoll *Internet Key Exchange (IKE)* wird ebenfalls vorgestellt.

1 Einleitung

Wie bereits in mehreren Beiträgen gezeigt wurde, ist der Bedarf nach Sicherheitsmechanismen in Kommunikationsnetzen stetig gewachsen. Das derzeitige Netzwerkstandardprotokoll *Internet Protocol* in der Version 4 (IPv4) bietet jedoch keine inhärente Sicherheit. So hat der Empfänger eines IP-Pakets keine Garantie,

- dass die Quelle der Nachricht der im IP-Header angegebenen Adresse entspricht,
- dass keine andere Person den Dateninhalt gelesen hat,
- dass die Daten auf dem Weg zwischen Sender und Empfänger nicht von Dritten verändert wurden.

Diese drei Bedrohungsszenarien werden allgemein als „IP-spoofing“, „sniffing“ und „data spoofing“ bezeichnet.

Ein Sicherheitsprotokoll auf IP-Ebene muss also ein unbefugtes Mitlesen und Verändern von Daten verhindern und die Kommunikationspartner authentifizieren. Um Vertraulichkeit herzustellen, bietet es sich an, die Daten zu verschlüsseln; bei der großen Anzahl von Paketen, die im IP-Verkehr pro Zeiteinheit anfallen, scheidet ein kryptographisches Verfahren mit asymmetrischen Schlüsseln aus, da es nicht performant genug ist. Die bei einem symmetrischen Verfahren notwendigen Schlüssel müssen jedoch sicher zwischen den Kommunikationspartnern ausgetauscht und verwaltet werden. Daneben sollte es auch möglich sein, weitergehende Parameter zu verhandeln, beispielsweise die zu verwendenden Verschlüsselungsalgorithmen oder die Lebensdauer von Schlüsseln.

IPSec liefert Lösungen für all die eben erwähnten Probleme. Im Folgenden soll auf die einzelnen Aspekte von IPSec detailliert eingegangen werden.

2 IPSec im Überblick

2.1 Die Entwicklung von IPSec

Ursprünglich war IPSec als Teil der Entwicklung von IPv6, dem Nachfolger des heute quasi als Standard verwendeten Netzwerkprotokolls IPv4, vorgesehen. Mitte der 1990er Jahre wurde eine Komission gebildet, der die *Internet Engineering Task Force (IETF)* vorstand; die Entwicklung mündete im Jahr 1998 in mehrere *Requests for Comments (RFC)*, womit IPSec ein offizieller Standard wurde. Im Verlauf der Arbeit fiel der Entschluss, IPSec so generisch zu entwickeln, dass es auch mit IPv4 verwendet werden kann.

2.2 Die Bestandteile von IPSec

2.2.1 Die Protokolle

IPSec ist nicht ein einzelnes umfassendes Protokoll, sondern eine Suite von mehreren Protokollen, die Methoden zum Schutz von IP-Paketen vor den in der Einleitung erwähnten Bedrohungen anbieten. Im Wesentlichen sind dies die folgenden Protokolle:

- **Encapsulating Security Payload (ESP)**. ESP ermöglicht es, den Kommunikationspartner zu authentifizieren, sowie die Integrität der Daten und deren Vertraulichkeit zu sichern.
- **Authentication Header (AH)**. Wie ESP authentifiziert AH den Sender eines Pakets und gewährt Datenintegrität, leistet jedoch keine Verschlüsselung.
- **Internet Key Exchange (IKE)**. Die Verständigung über die zu verwendenden Sicherheitsalgorithmen und der dazu notwendige Austausch von Schlüsselmaterial erfolgt mittels IKE.

ESP und AH, die beiden Protokolle die den tatsächlichen Schutz von IP-Verkehr leisten, definieren jeweils einen eigenen Header, also den *ESP-Header* bzw. den *AH-Header*. Die Protokolle sind parallel einsetzbar, das bedeutet man hat die Wahl, ob man AH, ESP, oder eine Kombination aus beiden verwenden möchte. Aus diesem Grund spricht man bei allgemeinen IPSec-Szenarien vom *IPSec-Header*. ESP hat zusätzlich einen Trailer, der an das ursprüngliche Paket angehängt wird. Aus Gründen der Vereinheitlichung soll dieser im Folgenden als IPSec-Trailer bezeichnet werden, falls im gleichen Kontext von einem IPSec-Header die Rede ist.

2.2.2 Sicherheitsassoziation

Möchten zwei Entitäten über IPSec miteinander kommunizieren, so werden alle für den Verbindungsaufbau benötigten Parameter und Schlüssel in einer so genannten **Sicherheitsassoziation (SA)** festgehalten. Man kann sie sich als einen Vertrag vorstellen, der beschreibt, *welche* Art von Verkehr mit *wem* zu schützen ist, und *wie* dies geschehen soll. Das Prinzip einer Sicherheitsassoziation wird in Abschnitt 6 ausführlich erklärt.

2.3 Einsatzmöglichkeiten von IPSec

2.3.1 Host to Host

Wie der Name *IP Security* schon vermuten lässt, sind die Schutzmechanismen die es bietet auf IP-Ebene angesiedelt. Im einfachsten Anwendungsfall bezieht sich dieser Schutz auf die Kommunikation zwischen zwei Hosts, die beide mit einer Implementierung von IPSec versehen sind. Hier laufen Authentifizierung sowie Ver- und Entschlüsselung unmittelbar auf den beiden an der Kommunikation beteiligten Hosts ab. Anders formuliert kann man sagen, dass der so genannte *kryptographische Endpunkt* auch dem Kommunikationsendpunkt entspricht (siehe Abb. 1).

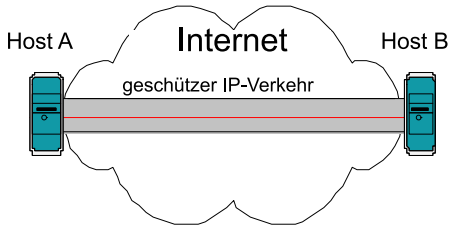

Abbildung 1: Host to Host Szenario

2.3.2 Virtual Private Network

Daneben ermöglicht IPSec auch den Aufbau eines **Virtual Private Networks** (Virtuelles privates Netzwerk - VPN), eine Eigenschaft von hohem praktischen Nutzen. Bei einem VPN werden zwei private Subnetze über ein unsicheres öffentliches Netz wie z. B. dem Internet durch einen so genannten IP-Tunnel verbunden. Dazu wird für jedes der beiden Subnetze ein VPN-Gateway benötigt, das einerseits mit dem eignenen privaten Netz und andererseits mit dem Öffentlichen verbunden ist. Die Datenpakete aus den privaten Netzen, deren Ursprungs- und Zieladressen jeweils nicht im erlaubten IP Adressbereich für Routing liegen, bekommen auf dem Gateway des Quellnetzes gültige IP Adressen für den Transport - man könnte anschaulich sagen, sie werden „eingepackt“. Nach erfolgreichem Routing zum Zielgateway erfolgt das Auspacken und Weiterleiten der Daten an den eigentlichen Zielhost. Durch dieses Kapseln eines IP-Pakets in in weiteres IP-Paket erscheint es dem Empfänger, als ob es ihn direkt und - mit Außnahme der VPN-Gateways - ohne Zwischenhops ereicht habe, quasi in einem Tunnel zwischen den Gateways (siehe Abb. 2).

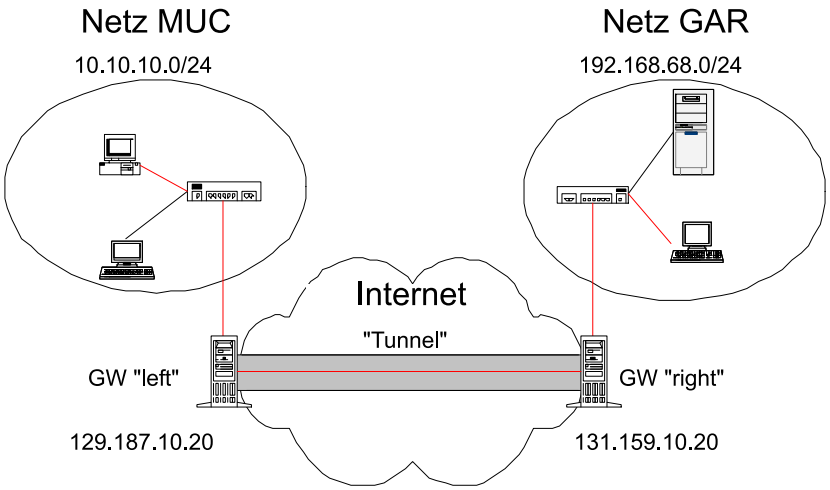

Abbildung 2: VPN Szenario

Ein VPN auf der Basis von IPSec leistet natürlich mehr als bloßes Tunneling. Auf den VPN-Gateways, oder besser gesagt IPSec-Gateways, erfolgt auch die Verschlüsselung des zu versendenden IP Pakets, bzw. die Authentifizierung und Entschlüsselung des empfangenen Pakets. Hier ist das Gateway zwar kryptographischer Endpunkt, nicht aber Endpunkt der Kommunikation; dieser ist ja der Zielrechner im Subnetz.

2.4 Verschiedene Arten der Implementierung

Es gibt mehrere Möglichkeit, wie IPSec auf einem Host oder einem Gateway implementiert weden kann. Das Dokument für IP-Sicherheitsarchitektur ([KA98a]) beschreibt einige

Beispiele:

Zum Einen wird die **Integration in das Betriebssystem** genannt. Hier wird IPSec als Teil der Netzwerkschicht des Rechners implementiert. Dem Vorteil der nahtlosen Zusammenarbeit von IPSec mit den Protokollen der benachbarten Netzwerkschichten steht der Nachteil gegenüber, dass man hierfür Zugang zu den Quellen des Betriebssystems benötigt.

Ein zweite Möglichkeit wird als **Bump-in-the-stack (BITS)** bezeichnet. Wie der Name schon vermuten läßt, wird IPSec hier wie ein Zwischenstück zwischen der Implementierung des IP-Protokollstapels im Betriebssystem und den lokalen Netzwerktreibern eingefügt. Bei einer BITS-Implementierung existiert IPSec als eigene Teilschicht zwischen der Netzwerk- und der Datenübertragungsschicht. Der Vorteil hiervon ist, dass man weitgehend unabhängig von den Besonderheiten des Betriebssystems ist. Allerdings führt BITS zu einer Erhöhung des Aufwands, da einige Eigenschaften der Netzwerkschicht, wie z. B. Routing-Tabellen und Fragmentierung, auch auf der IPSec-Schicht implementiert werden müssen.

Als weitere Variante wird die Implementierung von IPSec auf einem gesonderten Gerät genannt, das einerseits physikalisch mit einem Router, andererseits mit dem öffentlichen Netz verbunden ist. Diese Methode nennt sich **Bump-in-the-wire (BITW)**. Sie bietet sich an, wenn es nicht möglich ist, direkt auf den zu schützenden Host oder das Netz-Gateway zuzugreifen, hat aber den Nachteil, dass für jede Schnittstelle des Routers ein eigenes IPSec-Gerät verwendet werden muss.

2.5 Sicherheit - warum auf IP-Ebene?

2.5.1 Exkurs: Sicherheit auf der Anwendungschicht

Auf der Anwendungsebene (application layer) des Netzwerkprotokollstapels gibt es bereits eine Reihe von Sicherheitslösungen; beispielhaft seien hier *PGP (Pretty Good Privacy)* für den sicheren Mail- und *HTTPS* mit X.509-Zertifikaten für den sicheren Internetverkehr erwähnt. Es stellt sich also die Frage, wozu eine Sicherheitsanwendung auf IP-Ebene überhaupt benötigt wird.

Ein Vorteil, den ein Sicherheitsprogramm einer höheren Protokollebene bietet, liegt darin, dass sich der Schutz bis zum eigentlichen Endnutzer erstreckt: nur wenn dieser in der jeweiligen Anwendung für die korrekte Authentifizierung sorgt, z. B. durch Eingabe seines Paßwortes, bekommt er die wie auch immer gearteten Daten im Klartext zu sehen. Eine andere Person, die den selben Rechner benutzt, wird an dieser Anforderung scheitern.

Auf der anderen Seite wird so dem Benutzer eine große Verantwortung aufgebürdet. Wer weiß z. B. schon genau, ob ein X.509 Zertifikat, das einem angeboten wird, wirklich vertrauenswürdig ist - in der Regel geht man einfach davon aus [1]. Es gibt aber noch einen weiteren Grund dagegen, sich bei Sicherheitsaspekten nur auf die Anwendungsschicht zu beschränken: Jedes Programm definiert und implementiert seine eigenen Sicherheitsalgorithmen und tauscht dazu evtl. Schlüssel aus. Neben einem berträchtlichem Overhead, der dadurch das Netzwerk belastet, steigt auch mit jedem neuem Sicherheitsprogramm der Verwaltungsaufwand für den Benutzer, ganz zu schweigen von der Vervielfachung möglicher Fehlerquellen auf der Entwicklerseite.

2.5.2 Sicherheit auf der Netzwerkschicht

Mit einem Sicherheitsmechanismus wie IPSec, welcher auf der Netzwerkschicht operiert, ist es möglich, den *gesamten* Verkehr innerhalb eines Netzwerkes zu schützen, und das unabhängig von der jeweiligen Anwendung (einer höheren Schicht), von der die Daten stammen. Der Benutzer einer solchen Anwendung bekommt von diesem Vorgang nichts mit und muss sich vor allem keine Schlüssel oder Passwörter merken.

Es ist jedoch ganz wichtig, sich den Unterschied dieser Art von Sicherheit zu der unter 2.5.1 skizzierten vor Augen zu führen: IPSec sichert nur den Verkehr zwischen einzelnen

[1] Ein kleines Beipiel: In der Vorlesung „Internetprotokolle“ von Frau Professor Feldmann meldete sich kein einziger Student auf die Frage, wer schon einmal ein X.509 Zertifikat überprüft habe - wohlgemerkt allesamt Informatikstudenten.

Rechnern; was mit den Daten nach der Ankuft auf dem Zielhost geschieht, liegt nicht mehr in seinem Aufgabenbereicht. Es ist also durchaus möglich, dass eine geheimzuhaltende Nachricht sicher den Ziel*rechner* erreicht, dort jedoch von einer unbefugten Dritten Person mit Zugangsrechten gelesen wird. Ebensowenig erhöht IPSec die Sicherheit in einem Firmen- oder Universitätsnetzwerk, wenn Daten aus einem Partnernetz zwar sicher am Gateway ankommen, dann aber im eigenen privaten Netz aufgrund eines fehlenden Sicherheitskonzepts von Jedermann abgerufen werden können.

3 Die Modi von IPSec

Die beiden IPSec Protokolle ESP und AH können in zwei unterschiedlichen Modi arbeiten, dem so genannten Transportmodus und dem Tunnelmodus. Die beiden Modi weisen in ihrer Funktionsweise eine gewisse Parallelität zu den unter 2.3.1 und 2.3.2 beschriebenen Einsatz-Szenarien auf.

3.1 Der Transportmodus

Im Transportmodus werden Pakete aus höheren Protokollschichten, wie zum Beispiel TCP oder UDP, geschützt. Der IPSec-Header wird zwischen dem IP-Header und dem Header des übergeordneten Protokolls eingefügt (siehe Abb. 3). Nur zwischen zwei Entitäten, die beide mit einer Implementierung von IPSec arbeiten, kann der Transportmodus verwendet werden, also in einem Host-to-Host Szenario.

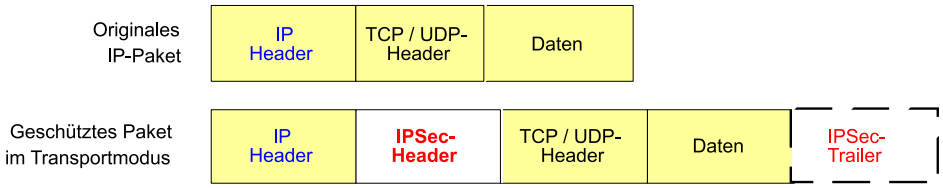

Abbildung 3: Geschütztes Paket im Transportmodus

3.2 Der Tunnelmodus

Im Tunnelmodus wird das gesamte ursprüngliche IP-Paket geschützt: Vor das IP-Paket wird ein IPSec-Header und zusätzlich ein weiterer IP-Header eingefügt, den man als „äußeren Header“ bezeichnet (siehe Abb. 4).

Abbildung 4: Geschütztes Paket im Tunnelmodus

Eine typische Anwendung des Tunnelmodus ist der Aufbau eines VPNs: Ein dezidiertes IPSec-Gateway versieht IP-Pakete aus einem privaten Netz mit einem IPSec-Header und dem äußeren IP-Header für den Transport und tunnelt diese an ein entferntes Gateway. Es ist jedoch genauso möglich, den Tunnelmodus im Host-to-Host Fall zu verwenden, mit dem Unterschied, dass die Ursprungs- und Zieladressen im neuen IP-Header identisch mit denen im ursprünglichen, jetzt eingepackten IP-Header sind.

4 Encapsulating Security Payload (ESP)

Das ESP Protokoll ist in RCF 2406 ([KA98c]) definiert und stellt Verschlüsselung, Datenintegrität und Authentifizierung der Quelle zur Verfügung, sowie optional einen begrenzten Schutz vor dem wiederholten Senden von Paketen, dem so genannten *„replaying“*. Enweder Vertraulichkeit oder Authentifizierung können deaktiviert werden, jedoch nicht beide zusammen.

ESP wird nur auf nicht fragmentierte IP-Pakete angewendet. Falls nötig, kann jedoch ein mit ESP geschütztes Paket zum Transport fragmentiert werden; es muss dann beim Empfänger wieder vollständig zusammengesetzt werden, bevor die weitere Bearbeitung gemäß ESP erfolgen kann.

4.1 Aufbau eines ESP Pakets

Der ESP-Header wird unmittelbar nach einen IPv4- bzw. IPv6-Header oder IPv6-Erweiterungsheader in das Datagramm eingefügt. Daneben bildet ESP einen Trailer, der unmittelbar an die Nutzdaten anschließt. ESP hat als Protokoll den Wert 50, daher wird der vorhergehende IP-Header diesen Wert im *Protocol*- bzw. *Next Header*-Feld tragen.

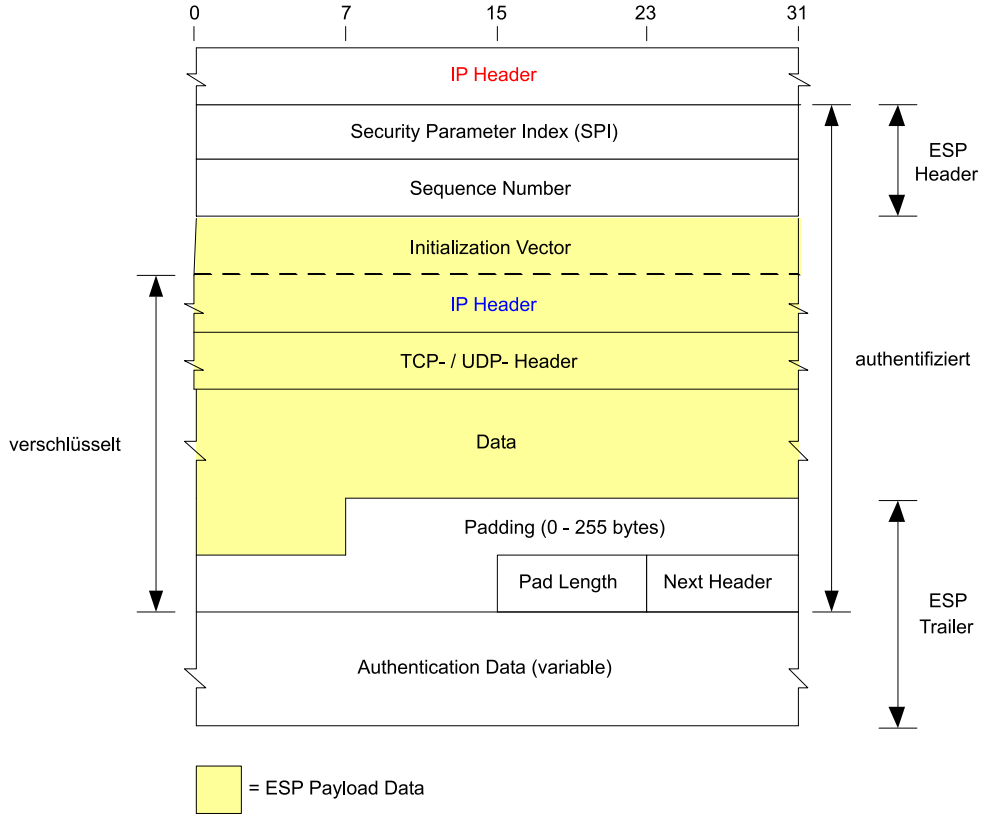

Abbildung 5: Durch ESP geschütztes IP-Paket im Tunnelmodus

Abbildung 5 zeigt den schematischen Aufbau eines IP-Pakets, das durch ESP im Tunnelmodus geschützt wird. Der erste IP-Header ist der neu generierte äußere Tunnel-Header, der zweite der des ursprünglichen Pakets. In den folgenden Abschnitten werden die einzelnen Felder des ESP-Headers und Trailers beschrieben.

- **Security Parameter Index (SPI)**. Der Sicherheitsparameter-Index (*Security Parameter Index - SPI*) ist ein 32-Bit Wert, der zusammen mit der Zieladresse und dem

verwendeten Sicherheitsprotokoll (in diesem Fall also ESP) die passende Sicherheitsassoziation zur Verarbeitung des Pakets festlegt. Der SPI wird vom Empfänger bestimmt und ist für die Dauer der SA gültig. Das SPI-Feld ist verpflichtend.

Da der SPI dem Empfänger dazu dient, die zur Entschlüsselung benötigte Information aufzufinden, wird der SPI zwar authentifiziert, jedoch nicht mit verschlüsselt. Andernfalls hätte man ein Henne-Ei-Problem [DH00].

- **Sequence Number**. Dieses 32-Bit Feld enthält eine eindeutige monoton steigende Seriennummer, die vor dem wiederholten Senden eines Pakets schützen soll. Der Sender ist verpflichtet, dieses Feld zu übermitteln, jedoch steht es dem Empfänger frei, es zu verwenden. Falls anti-Replay Schutz aktiviert ist, was standardmäßig der Fall ist, wird jedes Paket vom Empfänger verworfen, dessen Seriennummer bereits verwendet wurde. Diese Überprüfung auf Duplikate geschieht nach der Authentifizierung und vor der Entschlüsselung, da die Seriennummer nicht verschlüsselt wird.

  Die Zähler von Sender und Empfänger werden beim Etablieren einer Sicherheitsassoziation mit Null initialisiert. Das Seriennummernfenster darf nicht zyklisch sein, d.h. es muss eine neue SA erzeugt werden, wenn 2^{32} Pakete über eine SA versendet wurden.

- **Payload Data**. Die Nutzdaten des übergeordneten Protokolls (TCP oder UDP) plus ihr Header (im Tunnelmodus auch noch der IP-Header), also die Daten, die man eigentlich schützen möchte, sind in diesem Feld enthalten. Weiterhin kann es einen Initialisierungsvektor für den Verschlüssenlungsalgorithmus enthalten. Die Länge des Nutzdatenfelds hängt von der Länge der Daten ab.

  Initialisierungsvektor (IV). Wenn der Verschlüsselungsalgorithmus kryptographische Synchronisation benötig, wie z. B. einen Initialisierungsvektor für ein Blockchiffrierverfahren im *Cipher Block Chaining (CBC)* Modus, kann dieser im Nutzdatenfeld mitgeschickt werden. Für jeden solchen Algorithmus müssen Länge und Struktur des IV, dessen Lage innerhalb des Nutzdatenfeldes und das Zusammenspiel mit ESP in einem eigenen RFC spezifiziert werden. Für DES, der standardmäßig implementiert werden muss, belegt der IV die ersten acht Oktette (64 Bits) des Nutzdatenfelds ([MD98]).

- **Padding (Fülldaten)**. Manche Verschlüsselungsalgorithmen können nur Eingaben verarbeiten, deren Länge ein Vielfaches einer bestimmten Blockgröße ist. Das Padding Feld kann dazu benutzt werden, die zu verschlüsselnden Bits - das sind die Nutzdaten ohne einen möglichen IV, das Padding selbst, das *Pad Length*-Feld und das *Next Header*-Feld - auf eine feste Größe zu bringen. Auch wenn in der Sicherheitsassoziation keine Vertraulichkeit festgelegt wurde, werden manchmal dennoch Fülldaten dazu benötigt, um die beiden nachfolgenden, eben erwähnten Felder an die richtige Stelle innerhalb des 32 Bit „breiten“ Header zu bringen, das heisst an die letzten 16 Bits. Abgesehen davon kann man mit Padding auch die tatsächliche Länge der Nutzdaten verschleiern, was natürlich einen größeren Overhead zur Folge hat.

  Das Fülldatenfeld ist das erste des ESP-Trailers und kann zwischen 0 und 255 Bytes lang sein; es ist also optional. Alle Implementierungen von IPSec müssen jedoch das Generieren und Auswerten von Fülldaten unterstützen.

- **Pad Length**. Die Länge des unmittelbar vorgerhehenden Paddings wird in diesem 8-Bit Feld beschrieben, mit gültigen Werten zwischen 0 und 255. Auch wenn keine Fülldaten vorhanden sind, muss das *Pad Length* Feld angegeben werden.

- **Next Header**. Im *Next Header*-Feld wird die Art der im Nutzdatenfeld enthaltenen Daten spezifiziert. Dies kann z.B. ein IPv6 Erweiterungsheader oder der Bezeichner eines Protokolls aus einer höheren Schicht sein. Die dabei verwendeten Zahlen werden in [IANA] definiert.

  Wird ESP im Tunnelmodus angewendet, trägt dieses obligatorische Feld den Wert 4 für IP-in-IP; im Transportmodus ist es in der Regel 6 für TCP oder 17 für UDP.

- **Authentication Data**. Das Feld für die Authentifizierungsdaten ist optional und hängt in der Länge von dem verwendeten Authentifizierungsalgorithmus ab. Wenn in der SA Authentifizierung vereinbart wurde, enthält es den so genannten *Integrity Check*

Value (ICV); damit wird die Ausgabe des Authentifizierungsalgorithmus bezeichnet. Auf die Authentifizierung wird weiter unten in Abschnitt 4.3 näher eingegangen.

4.2 Verschlüsselung

ESP arbeitet mit symmetrischer Verschlüsselung, meistens mit Blockchiffrierverfahren im *Cipher Block Chaining (CBC)* Modus, aber auch Stream-Verschlüsselungsalgorithmen sind möglich. Bei der Verarbeitung mit CBC werden die einzelnen Blöcke verkettet: der vorangehende Chiffrenblock (oder ein Teil davon) wird mit der folgenden, zu verschlüsselnden, Nachricht mit XOR verknüpft, bevor die Berechnung des Chiffretextes erfolgt. Um zu verhindern, dass Paketverlust oder das Verändern der Paketreihenfolge die Entschlüsselung blockieren, muss man jedem Paket die zur kryptographischen Synchronisation notwendigen Daten entnehmen können. Diese finden sich entweder explizit als IV oder lassen sich aus dem Paket-Header berechnen.

Wie ein ESP-Paket ver- und entschlüsselt wird, bzw. wie die Synchronisation zu Stande kommt, ist in der Sicherheitsassoziation spezifiziert. Obwohl das ESP-Dokument ursprünglich DES als verpflichtend zu implementieren beschrieben hatte, wird inzwischen davon abgeraten, weil es gelungen ist, DES-Schlüsseln innerhalb von wenigen Stunden zu brechen; stattdessen wird die Verwendung von 3-DES empfohlen. Um der Möglichkeit Rechnung zu tragen, dass keine Verschlüsselung gewünscht wird, muss auch die so genannte NULL-Verschlüsselung unterstützt werden.

Die Verschlüsselung erfolgt, nachdem die Nutzdaten in ESP gekapselt und eventuell benötigte Fülldaten hinzugefügt wurden. Chiffriert werden die Nutz- und Fülldaten, sowie das Pad Length-Feld und das Next Header-Feld anhand der in der SA vereinbarten Parameter.

4.3 Authentifizierung

Mit welchen Algorithmen ein ESP-Paket authentifiziert wird, ist in [KA98c] nicht explizit festgelegt, jedoch werden *keyed Message Authentication Codes (MACs)* mit Einweg-Hashfunktionen empfohlen, vor allem in Verbindung mit dem HMAC-Verfahren ([KBC97]). Eine Implementierung von ESP muss HMAC-MD5 ([MG98a]) und HMAC-SHA-1 ([MG98a]) sowie die so bezeichnete NULL-Authentifizierung unterstützen, kann darüber hinaus aber weitere Authentifizierungsalgorithmen verwenden.

Der ICV wird über das ESP-Paket ohne das Authentifizierungsdaten-Feld berechnet, d. h. der SPI, die Seriennummer, die Nutzdaten, eventuell vorhandenes Padding, das Pad Length-Feld und das Next Header-Feld gehen in die Berechung ein. Falls Verschlüsselung gewählt wurde, sind die letzten vier Felder chiffriert, da die ICV-Berechnung nach der Verschlüsselung stattfindet. Der Empfänger eines ESP-Paktes führt selbst die ICV-Berechnung durch und vergleicht diesen Wert mit dem vom Sender mitgelieferten; sind beide Werte nicht identisch, wird das Paket verworfen.

Anders als bei einer Prüfsumme, die nur zur Kontrolle der korrekten Übertragung dient und nicht die Integrität der Daten beweist, erhält der Authentifizierungsalgorithmus neben den Daten einen symmetrischen Schlüssel als Eingabe. Einem Angreifer wird es ohne Kenntnis dieses Schlüssels nicht gelingen, nach Verändern der Nutzdaten einen ICV zu erzeugen, den der Empfänger akzeptiert [2].

5 Authentication Header (AH)

Das IPSec-Protokoll Authentication Header (AH) ist in RFC 2402 ([KA98b]) definiert und ermöglicht die Gewährleistung der Datenintegrität, die Authentifizierung der Quelle und optional einen begrenzten Schutz vor dem Replaying-Angriff. Bis auf die Tatsache, dass AH seine Nutzdaten nicht verschlüsselt, ist es in seiner Leistung ESP ähnlich.

[2] Genauer: es dürfte es ihm sehr schwer fallen, die Nachricht so zu verändern, dass der ICV durch eine Kollision des Hash-Algorithmus zufälligerweise für den Empfänger akzeptabel ist.

Ebenso wie ESP wird AH nur auf nicht fragmentierte IP-Pakete angewendet; ein AH-Paket für den Transport zu fragmentieren ist jedoch möglich.

5.1 Aufbau eines AH Pakets

Auch der AH-Header wird unmittelbar nach einem IP-Header in das Datagramm eingefügt. AH definiert allerdings keinen Trailer. Der Protokollwert von AH ist 51.

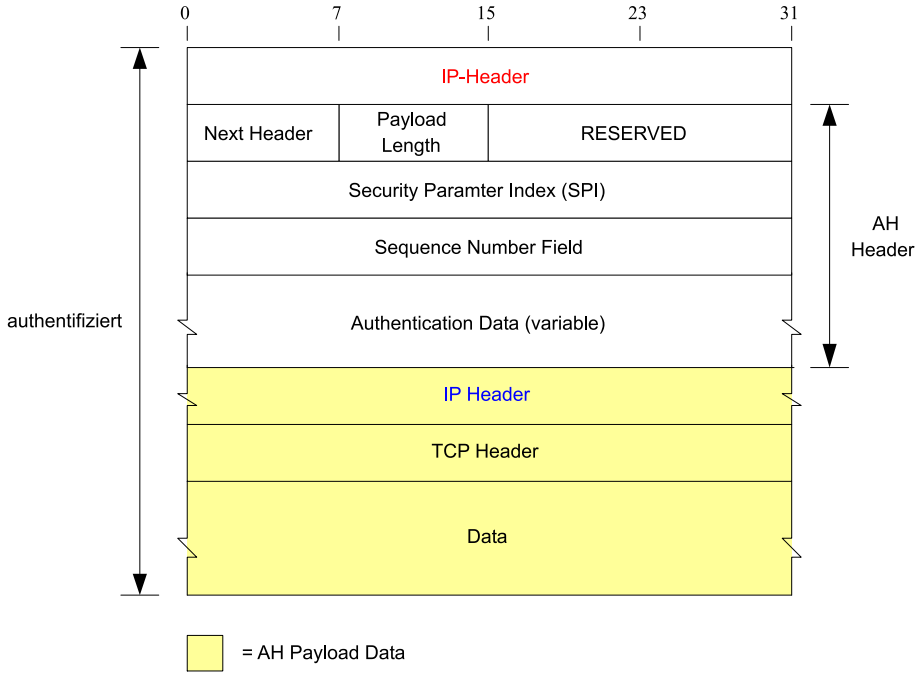

Abbildung 6: Durch AH geschütztes IP-Paket im Tunnelmodus

Die Abbildung 6 zeigt den schematischen Aufbau eines Pakets, das von AH im Tunnelmodus geschützt wird. Der erste IP-Header entspricht wiederum dem äußeren Tunnel-Header, der zweite ist der des urprünglichen IP-Datagramms. Die einzelnen Felder des AH-Headers werden in den folgenden Abschnitten beschrieben.

- **Next Header**. Wie im gleichnamigen Feld bei ESP beschreibt dieses 8-Bit Feld die Art der Nutzdaten anhand einer von [IANA] vergebenen Protokollnummer.
- **Payload Length**. Dieses 8-Bit Feld gibt die Länge des Headers in 32-Bit Worten minus 2 an[3]. In dem häufig anzutreffenden Szenario, dass 96 Bit für die Authentifizierung verwendet werden, plus die 3 festen 32-Bit Worte des Headers, enthält das Payload Length-Feld den Wert 4.
- **RESERVED**. Die letzten beiden Oktette des ersten 32-Bit Worts im AH-Header sind reserviert, um in Zukunft mit Bedeutung gefüllt zu werden. Das reservierte Feld muss vom Sender mit dem Wert Null versehen und vom Empfänger ignoriert werden.
- **Security Parameter Index (SPI)**. Das SPI-Feld entspricht in seiner Bedeutung exakt dem gleichnamigen Feld bei ESP (siehe Abschnitt 4.1).
- **Sequence Number**. Auch die Verwendung von Seriennummern erfolgt mit AH genau so wie mit ESP (siehe Abschnitt 4.1).

[3]Das rührt daher, dass bei allen IPv6 Extension Headern, zu denen AH zählt, die *Header Extension Length (Hdr Ext Len)* dadurch berechnet wird, dass zuerst 1 von der tatsächlichen Länge des Headers subtrahiert wird. IPv6 Header bestehen jedoch aus 64-Bit Worten, so dass zwei 32-Bit Worte unter IPv4 subtrahiert werden müsen.

- **Authentication Data**. Das Authentication Data-Feld enthält den Integrity Check Value (ICV), der zur Verifizierung der Datenintegrität dient. Die Länge des Feldes hängt von dem zur Authentifizierung verwendeten Algorithmus ab, muss jedoch ein ganzzahliges Vielfaches von 32 Bit (IPv4) bzw. 64 Bit (IPv6) sein. Um dies zu gewährleisten, müssen alle Implementierungen implizit das Verwenden von Padding unterstützen.
- **Payload Data**. Im Nutzdatenfeld befinden sich die Daten des übergeordneten Protokolls (TCP oder UDP) mit ihrem Header. Im Tunnelmodus kommt noch der IP-Header des ursprünglichen Pakets dazu.

5.2 Authentifizierung

Obwohl mit AH die gleichen Algorithmen zur Authentifizierung verwendet werden können - bzw. implementiert werden müssen - wie mit ESP (siehe 4.3) , gibt es doch einen grundsätzlichen Unterschied: AH erstreckt seinen Schutz auch auf den *äußeren* IP-Header. Der ICV wird über den äußeren IP-Header und den AH-Header mit den Feldern Next Header, Payload Length, Reserved, SPI, Sequence Number und Authentication Data, sowie natürlich die Nutzdaten, berechnet. Das Feld für die Authentifizierungsdaten wird dazu auf Null gesetzt.

Da der äußere IP-Header für das Routing benutzt wird (sowohl im Transport-, als auch im Tunnelmodus), können sich einige seiner Felder im Verlauf des Transports verändern. Diese werden von der ICV-Berechnung ausgeschlossen und dazu mit Null belegt. Die folgende Abbildung 7 zeigt den Aufbau eines IP-Headers; die von [KA98b] als veränderlich bezeichneten Felder sind grau hinterlegt.

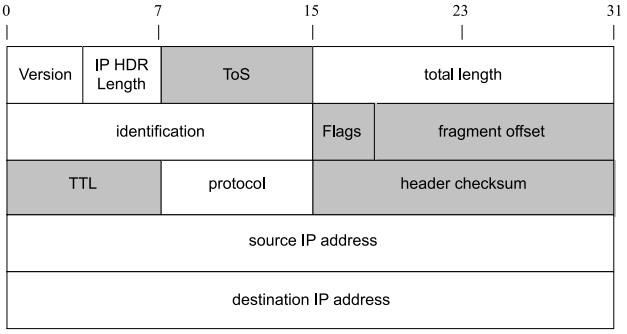

Abbildung 7: Veränderliche und unveränderliche Felder eines IPv4-Headers

- **Type of Service (ToS)**. In der Spezifikation von IP gilt dieses Feld als unveränderlich, dennoch wird es von manchen Routern verändert.
- **Flags**. Ein Router könnte das DF (*Don't fragment*) Bit setzten, auch wenn es von der Quelle nicht gewählt wurde.
- **Fragment offset**. AH wird nur auf nicht fragmentierte IP Pakete angewendet; daher muss dieses Feld Null sein. Obwohl sein Wert vorhersagbar ist, wird es von der Berechnung ausgeschlossen.
- **Time to live (TTL)**. Im normalen Vorgang des Routings wird die Lebensdauer von jedem Router dekrementiert.
- **Checksum**. Jede Änderung in einem der anderen Felder wirkt sich auf die Prüfsumme aus.

Der Empfänger eines AH-Pakets verfährt zur Überprüfung der Datenintegrität wie der Sender beim Erstellen des ICV, d.h. er ignoriert die veränderlichen Felder des IP-Headers.

6 IPSec-Sicherheitsassoziation

In den vorhergehenden Abschnitten wurde deutlich, dass für das Funktionieren von IPSec den Kommunikationspartnern eine Fülle von Parametern bekannt sein müssen. Diese werden in einem Konstrukt, das in dem Dokument für IP-Sicherheitsarchitektur ([KA98a]) als Sicherheitsassoziation (SA) bezeichnet wird, festgehalten. Man kann sich eine SA auch als logische Verbindung zwischen den beiden Entitäten vorstellen: obwohl IP kein verbindungsorientiertes Protokoll ist, kann IPsec-Kommunikation nur stattfinden, wenn die Teilnehmer anhand einer SA in einem bestimmten Zustand sind und dieser dem jeweiligen Partner bekannt ist.

Eine SA beschreibt u.a. welches Protokoll verwendet wird (AH oder ESP), wie authentifiziert wird (HMAC-MD5, HMAC-SHA-1, etc.) und wie im Fall von ESP Vertraulichkeit gewährt wird (DES, 3-DES, etc.). Weiterhin enthält sie alle Schlüssel und Initialisierungsvektoren. Falls zwei Gateways ihren IPSec-Verkehr mit ESP verschlüsseln und mit AH authentifizieren, dann existiert für AH und ESP je eine SA; IPSec-Kommunikation kann also durchaus von einem *Bündel* von Sicherheitsassoziationen geschützt werden.

Eine SA ist immer unidirektional, d. h. sie ist entweder für eingehenden oder für ausgehenden Verkehr zuständig. Aus diesem Grund werden SAs grundsätzlich paarweise erzeugt. Bezeichnet man zwei IPSec-Entitäten als Host A und Host B, dann hat die SA auf Host A für ausgehenden Verkehr (SA_{A-Aus}) einiges gemeinsam mit der SA auf Host B für eingehenden Verkehr (SA_{B-Ein}), wie etwa Schlüssel und den SPI.

6.1 Datenbanken für Sicherheitsassoziationen

Jeder Host, der an IPSec-Verkehr teilnehmen möchte, muss in der Lage sein, lokal Sicherheitsassoziationen zu verwalten. In [KA98a] wird dafür die **Security Association Database (SAD)** eingeführt, welche die Parameter aller gültigen SAs enthält. Da man generell den gesamten IP-Verkehr betrachtet und für jedes Paket entscheidet, wie damit zu verfahren ist, wird eine Datenbank mit Anwendungsstrategien benötigt, die so genannte **Security Policy Database (SPD)**. Beide Datenbanken werden im Folgenden kurz erläutert.

6.1.1 Security Policy Database (SPD)

Eine SA legt genau fest, wie ein bestimmter Teil des IP-Verkehrs behandelt werden soll. Bisher wurde aber noch nicht erklärt, woher die zum Erzeugen einer SA nötige Information stammt. In anderen Worten: woher weiß der Host oder das IPsec-Gateway, dass TCP-Verkehr in das Netz C mit ESP verschlüsselt werden soll? Zu diesem Zweck gibt es die Datenbank für Anwendungsstrategien, in der es für jedes IP-Paket anhand gewisser Selektoren möglich ist, die Verfahrensweise bezüglich IPSec zu bestimmen.

Man kann sich die SPD vom Aufbau her vorstellen wie ein Paketfilter einer Firewall: Eine geordnete Liste von Auswahlkriterien, meist vom Spezifischen ins Allgemeine gehend; jeder Eintrag enthält eine Regel für die Verfahrensweise. Für jedes IP-Paket werden seine Parameter mit denen in der SAD verglichen, und das erste Matching liefert die Anwendungsstrategie. Selektoren sind u. a. die Quell- und Zieladresse, das verwendete Transportprotokoll (z. B. TCP oder UDP) und die Quell- und Zielports. Drei mögliche Policies sind definiert:

- **Discard**. Das Paket wird fallengelassen.
- **Bypass IPSec**. Das Paket darf IPSec umgehen und wird an das „normale“ IP Routing übergeben.
- **Apply IPSec**. Das Paket wird durch IPSec geschützt. In diesem Fall muss angegeben werden, wie das geschehen soll, also mit welchen Protokoll(en) und Algorithmen.

Falls IPSec angewendet werden soll und zu der spezifizierten Policy eine Sicherheitsassotiation existiert, enthält der SPD-Eintrag einen Verweis auf die SA in der SAD. Abbildung 8 zeigt den stark vereinfachten Aufbau einer SPD.

source	destination	protocol	src-port	dst-port	…	policy	action	SA ID
10.10.10.0/24	192.168.68.0/24	TCP	80	80	…	apply IPSec	Tunnel-ESP with HMAC-MD5	SA<MUC-GAR>
10.10.10.0/24	129.187.60.70/32	any	any	any	…	apply IPSec	Tunnel-AH with HMAC-MD5	--
any	any	any	any	any	…	discard	–	--

Abbildung 8: Beispielhafter Aufbau einer SPD

6.1.2 Security Association Database (SAD)

Die Parameter, die mit einer Sicherheitsassoziationen verbunden sind, werden in der SAD gehalten. Oft kann ein Eintrag neben einem konkreten Wert auch einen Werteumfang oder Wildcards enthalten. Ein paar der wichtigsten Felder sind:

- Der Zähler für die Seriennumer. Das ist ein 32-Bit Wert, der bei ausgehendem Verkehr für die Berechnung der Seriennummer verwendet wird.
- Das Anit-Replaying Fenster. Ein 32-Bit Wert und eine Bitmap werden benutzt, um empfangene Pakete auf Duplikate zu überprüfen.
- AH Authentifizierung. Dazu gehören der Algorithmus, Schlüssel, etc.
- ESP Authentifizierung. Dazu gehören der Algorithmus, Schlüssel, etc.
- ESP Verschlüsselung. Dazu gehören der Algorithmus, Schlüssel, Angaben über die Verwendung eines IV, der IV etc.
- SA Lebensdauer. Gibt ein Zeitinterval an, nach dem die SA und der SPI erneuert bzw. verworfen werden müssen. Sie kann als Zeitwert oder Bytewert angegeben werden.
- IPSec Protokoll Modus. Transport- oder Tunnelmodus.

Die folgende Abbildung 9 soll den Aufbau einer SAD verdeutlichen - wiederum stark vereinfacht.

Tunnel - dest	src	dst	AH auth	ESP auth.	ESP encr.	Life-time	mode	…	SPI	SA ID
131.159.10.20	10.10.10.0/24	192.168.68.0/24	–	HMAC-MD5, key, etc	3-DES, key, IV, etc	4h	tunnel	…	47..11	SA<MUC-GAR>

Abbildung 9: Beispielhafter Aufbau einer SAD

6.2 Die Verarbeitung von IPSec-Verkehr

6.2.1 Ausgehender Verkehr

Auf einem IPSec-Rechner werden zur Verarbeitung des ausgehenden Verkehrs die Selektoren jedes IP-Pakets mit den Einträgen in der SPD verglichen. Falls keine Übereinstimmung vorhanden ist, wird das Paket fallen gelassen, ansonsten wird entsprechend der Anwendungsstrategie verfahren, d. h. das Paket wird entweder verworfen, an IPSec vorbei gelassen oder durch IPSec geschützt. Im letzten Fall liefert die SPD einen Zeiger auf eine SA (oder ein Bündel von SAs) in der SAD.

Die Selektorenfelder des Pakets werden jetzt mit denen der SA (oder des SA-Bündels) verglichen. Kann keine Übereinstimmung gefunden werden, muss eine neue SA erzeugt werden, und der SPD-Eintrag mit der SAD verlinkt werden. Existiert kein Mechanismus zur Schlüsselverwaltung (wie IKE, siehe Abschnitt 7), muss das Paket verworfen werden.

Die so gefundene oder neu erzeugte SA (oder das SA-Bündel) wird auf das Paket angewendet.

6.2.2 Eingehender Verkehr

Bevor mit der Verarbeitung von ESP oder AH begonnen werden kann, müssen eventuell fragmentierte IP-Pakete wieder zusammengesetzt werden. Anhand des AH- oder ESP-Protokollwertes im Next Header-Feld des IP-Headers kann festgestellt werden, dass IPSec-Behandlung notwendig ist. Um die korrekte SA zu finden, wird wie folgt vorgegangen:

Als erstes wird das Tupel ⟨Zieladresse, IPSec-Protokoll, SPI⟩ zum Auffinden der SA in der SAD verwendet. Die Zieladresse ist die des äußeren IP-Headers. Findet sich keine SA, ist das Paket zu verwerfen.

Die Policy der so gefundenen SA wird auf das Paket angewendet, d.h. es wird authentifiziert und entschlüsselt. Außerdem findet ein Vergleich der Selektoren des IP-Headers mit denen der SA statt; falls man sich im Tunnelmodus befindet, wird dazu der innere IP-Header benutzt.

In der SPD muss nun eine Regel für eingehenden Verkehr gefunden werden, auf die die Parameter des Pakets zutreffen. Das kann einerseits dadurch geschehen, dass man der SAD Zeiger auf die SPD gibt [4] und man so zu einer SA einen Verweis auf die SPD erhält. Andererseit kann man auch die Selektoren des Paktes mit der SPD vergleichen.

Es wird überprüft, ob das Paket die laut SPD-Eintrag korrekte Behandlung erfahren hat. Ist das nicht der Fall, wird die SPD weiter durchsucht, bis alle Regeln betrachtet wurden oder die Überprüfung erfolgreich ist. Anschließend wird das Paket an die Transportschicht übergeben (Transportmodus) oder an das im inneren IP-Header angegebene Ziel weitergeleitet (Tunnelmodus).

7 Internet Key Exchange (IKE)

Wie das Kapitel über Sicherheitsassoziationen gezeigt hat, ist für das Funktionieren von IPSec ein Mechanismus zum Erzeugen von authentifiziertem Schlüsselmaterial und Austausch von Sicherheitsparametern essentiell. Dazu wurde der Internet Key (IKE) Exchange entwickelt und in RCF 2409 ([HC98]) spezifiziert.

IKE ist ein hybrides Protokoll; das bedeutet es benutzt mehrere andere Protokolle bzw. Teile davon. Zu nennen ist als wichtigstes das **Internet Security Association and Key Management Protocol (ISAKMP)** ([MSSJ98]), das den Rahmen für die Authentifizierung und den Schlüsselaustausch liefert und zwei verschiedene Phasen definiert. **Oakley** ([Orm98]) beschreibt eine Reihe von Schlüsselaustauschvorgängen, die als Modi bezeichnet werden, und **SKEME** ([Kra98]) definiert eine vielseitige Schlüsselaustauschtechnik, die für Anonymität und ein schnelles Auffrischen von Schlüsseln sorgt. IKE-Kommunikattion findet mit UDP auf Port 500 statt.

Im Folgenden werden die Vorgänge innerhalb der beiden ISAKMP-Phasen erläutert.

7.1 Phase 1

Phase 1 hat das Ziel, die Grundlagen für das Aushandeln von IPSec-SAs zu legen. Dazu führen die beiden Kommunikationspartner einen Schlüsselaustausch nach Diffie-Hellman (siehe [DH]) durch und authentifizieren diesen. Auch weitere Parameter, wie z. B. die Lebensdauer der SA und die Methoden der Authentifizierung werden in Phase 1 verhandelt. Der so entstandene „Vertrag“ wird als ISAKMP-Sicherheitsassoziation bezeichnet. Anders als eine IPSec-SA ist sie bidirektional. Die ISAMKP-SA kann dann in Phase 2 verwendet werden, um SAs für ESP und AH zu erzeugen.

7.1.1 Verschiedene Modi und Authentifizierungsmethoden

In Phase 1 können zwei verschiedene Modi benutzt werden, der **Main Mode** und der **Aggressive Mode** . Beide haben das gleiche Ergebnis, doch ist der Aggresiv-Modus weniger

[4] Der umgekehrte Fall, nämlich die Verzeigerung von der SPD zur SAD, ist ja grundsätzlich gegeben.

flexibel und schützt die Identität der Kommunikationspartner nicht. Die Anzahl der ausgetauschten Nachrichten ist bei beiden Modi genau festgelegt. Im Main Mode tauschen Sender und Empfänger insgesamt sechs, im Aggresiv-Modus drei Nachrichten aus.

Für beide Modi existieren vier Methoden der Authentifizierung: eine mit digitaler Signatur, zwei verschiedene mit öffentlichen Schlüsseln und eine mit einem (vorher festzulegenden) gemeinsam bekannten Geheimnis. Abhängig von der Wahl der Authentifizierungsmethode haben die IKE-Nachrichten unterschiedlichen Inhalt, wobei das Ziel das gleiche ist. Im Folgenden wird am Beispiel eines Main-Mode-Austauschs mit öffentlichen Schlüsseln zur Authentifizierung die Funktionsweise von ISAKMP Phase 1 gezeigt.

7.1.2 Beispiel: Main Mode mit öffentlichen Schlüsseln

Im Main Mode werden drei Paare von Nachrichten ausgetauscht: die ersten beiden verhandeln die Anwendungsstrategie, die nächsten beiden führen einen Schlüsselaustausch nach Diffie-Hellman durch und tauschen Hilfsdaten wie z. B. Noncen aus, und die letzten beiden authentifizieren den Diffie-Hellman-Austausch. Bei dem in Abschnitt 5.2 von [HC98] beschriebenen Vorgang mit Public-Key-Verschlüsselung werden verschlüsselte Noncen zur Authentifizierung verwendet. Der Aufbau der einzelnen Nachrichten kann aus Abbildung 10 entnommen werden:

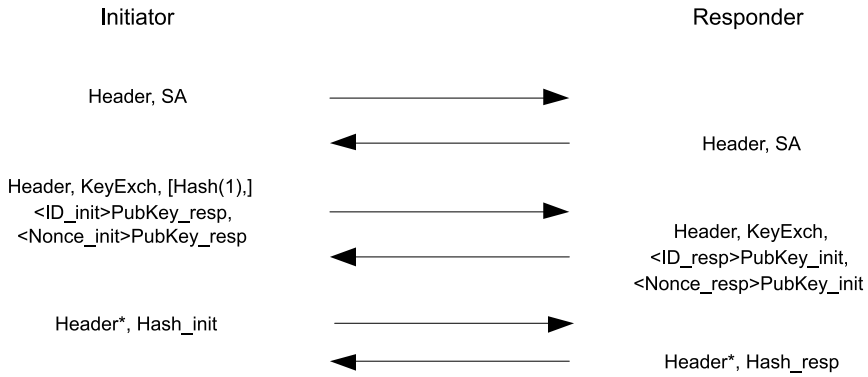

Abbildung 10: ISAKMP Main Mode mit Public-Key-Verschlüsselung

Header bezeichnet jeweils den ISAKMP-Header, der den Modus angibt. Ein mit einem Stern versehener Header hat die Bedeutung, dass die nachfolgenden Nutzdaten verschlüsselt sind.

Das Aushandeln der Anwendungsstrategie (**SA**) geschieht so, dass der Initiator mehrere Vorschläge machen *kann*, und der Responder einen davon auswählen *muss*. Falls der Responder keinen der Vorschläge akzeptabel findet, kommt keine ISAKMP-SA zu Stande; auf keinen Fall dürfen die Vorschläge des Initiators modifiziert werden.

Für den Diffie-Hellman-Austausch (**KeyExch**) wählen beide Seiten je eine Primzahl als privaten Wert (mit x bzw y bezeichnet). Der Initiator wählt zudem einen Generator g und eine Primzahl p anhand einer beiden bekannten Gruppe [5]. Er berechnet seinen öffentlichen Wert $A = g^x \mod p$ und sendet A, g und p an den Responder. Dieser sendet seinen öffentlichen Wert $B = g^y \mod p$ zurück. Beide Seiten können nun den öffentlichen Wert des anderen mit ihrem eignen privaten Wert potenzieren und erhalten so das gemeinsame Geheimnis $g^{ab} \mod p = A^y \mod p = B^x \mod p$.

Die Identitäten des Initiators und Responders (**ID_init**, bzw. **ID_resp**) - in der Regel deren IP-Adresse - werden jeweils mit dem öffentlichen Schlüssel des Gegenübers verschlüsselt; ebenso wird mit den Noncen der beiden Kommunikationspartner (**Nonce_init**, bzw. **Nonce_resp**) verfahren. Diese dienen dazu, einen „Man-in-the-middle Angriff“ zu verhindern

[5] IKE definiert vier Gruppen für den Diffie-Hellman-Austausch. Sie stammen alle aus dem Oakley-Protokoll. In der SA wird bestimmt, welche Gruppe verwendet wird.

und den so genannten „proof of liveliness“ zu erbringen, d. h. dass es sich um keine veraltete Nachricht handelt, die von einem Angreifer abgefangen wurde und jetzt wiederverwendet wird. Die öffentlichen Schlüssel müssen vor einem Phase 1 Austausch bekannt sein. Falls der Responder mehrere öffentliche Schlüssel hat, schickt der Initiator einen Hash über den verwendeten Schlüssel (**Hash(1)**) mit.

Die Authentifizierung geschieht dadurch, dass beide Seiten die (verschlüsselte) Nonce des anderen mit ihrem privaten Schlüssel dechiffrieren und in einem Hash zurücksenden (**Hash_init**, bzw. **Hash_resp**). So beweisen sie, dass sie im Besitz des privaten Schlüssels sind (was einem Angreifer nicht gelingt wird) und dass sie auf eine aktuelle Nachricht antworten (was mit einer abgefangenen Nachricht mit einer alten Nonce nicht möglich ist). Hash_init und Hash_resp sind jeweils mit einem Schlüssel, der aus dem gemeinsamen Geheimnis g^{ab} gebildet wurde, verschlüsselt. Aus dem gemeinsamen Geheimnis wird - anhand eines deterministischen Algorithmus - auch der Quell-Schlüssel für IPSec-SAs abgeleitet, die in Phase 2 erzeugt werden.

7.2 Phase 2

Für Phase 2 ist der **Quick Mode** definiert. Er kann nur nach einem Phase 1 Austausch erfolgen und hat das Ziel, für ESP und AH Anwendungsstrategien zu verhandeln und Schlüsselmaterial zu erzeugen. Die im Quick Mode ausgetauschten Nachrichten sind grundsätzlich von einer ISAKMP-SA geschützt. Es ist möglich, dass mehrere Quick-Mode-Austauschvorgänge gleichzeitig mittels einer ISAKMP-SA ausgeführt werden.

Es werden wiederum Noncen für den „Lebendigkeitsbeweis“ ausgetauscht. Sie finden auch beim Generieren des Schlüsselmaterials Verwendung. Da alle IPSec-Schlüssel von der selben, in Phase 1 erzeugten, Quelle abgeleitet sind, weisen sie per se keine *perfect forward secrecy (PFS)* auf. PFS ist eine Eigenschaft von Systemen wie dem Diffie-Hellman Schlüsselaustausch, bei dem ein Langzeitschlüssel für die grundlegende Kommunikation (hier: Phase 1) zum Einsatz kommt und je nach Bedarf mehrere kurzlebige Schlüssel für den eigentlichen Datentransport verwendet werden. Wenn die kurzlebigen Schlüssel jeweils aus unterschiedlichem Datenmaterial abgeleitet sind, kann ein Angreifer, der einen Schlüssel dechiffriert hat, keine Nachrichten lesen, die mit einem früheren oder späteren Schlüssel chiffriert wurden. Um die kurzlebigen Schlüssel, die im Quick Mode ausgetauscht werden, nicht von der in Phase 1 erzeugten Quelle ableiten zu müssen, kann ein zusätzlicher Diffie-Hellman-Austausch stattfinden und so PFS geschaffen werden.

Für die Identitäten (IDs) der an der SA beteiligten Kommunikationspartner werden standardmäßig die in Phase 1 ausgetauschten und authentifizierten IDs verwendet, das heißt IP-Adressen ohne Portnummern und Angaben zu Protokollen höherer Schichten. Wird der ISAKMP Quick Mode verwendet, um eine SA für andere Protokolle (ESP oder AH) auszuhandeln, was der Regelfall ist, müssen die für die SA gültigen IDs mit angegeben werden. Das Format hierfür ist in [Pip98] definiert. Eine ID kann eine IP-Adresse (bzw. ein *fully qualified domain name*), die Adresse eines Subnetztes oder ein Intervall von IP-Adressen sein. Außerdem können Angaben zu Ports und Protokollen gemacht werden. So ist es möglich, SAs von unterschiedlicher Granularität zu erzeugen. Eine SA könnte z.B. TCP-Verkehr auf Ports 80 und 8080 schützen, eine andere sämtlichen Verkehr außer FTP.

Abbildung 11 zeigt alle im Quick Mode ausgetauschten Felder; optionale Werte stehen in eckigen Klammern. Hash(1) ist der Hash-Wert der Nachrichten-ID des ISAKMP-Headers konkateniert mit der gesamten folgenden Nachricht. Hash(2) wird wie Hash(1) gebildet, außer dass die Nonce des Initiators hinzugefügt wird. Hash(3) wird über die Nachrichten-ID und die Noncen des Initiators und des Responders berechnet. So wird, wie schon in Phase 1 geschehen, die „Lebendigkeit“ der Kommunikationspartner verifiziert.

Existiert erst einmal eine ISAKMP-SA, kann jeder der Teilnehmer einen Austausch im Quick Mode initiieren. Dies ist besonders im Hinblick darauf wichtig, dass eine ISAKMP-SA in der Regel eine höhere Lebensdauer als eine IPSec-SA hat. Beiden Partnern muss es also vor Ablauf der Lebensdauer möglich sein, durch Erneuern der IPSec-SA die Schlüssel zu wechseln.

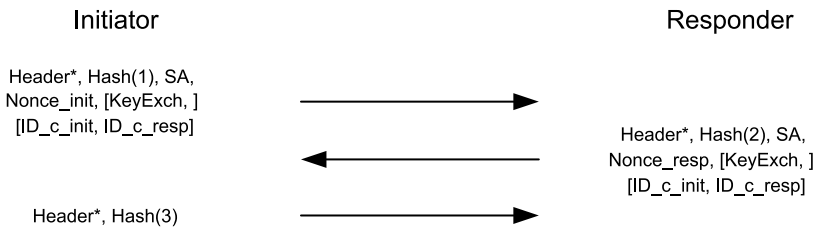

Abbildung 11: Der ISAKMP Quick Mode

8 Zusammenfassung und Ausblick

Die Leistung von IPSec - wie in den vorhergehenden Abschnitten deutlich wurde - besteht darin, Netzwerkkommunikation auf IP-Ebene zu sichern. Die Integrität der Nachrichten wird gewährleistet, das Mitlesen verhindert und die Identität der Kommunikationspartner verifiziert.

Die Stärken von IPSec sind

- seine Flexibilität, da ein Einsatz sowohl im End-to-End Bereich als auch zwischen Subnetzten möglich ist.
- seine Skalierbarkeit, da es aus Sicht des Konfigurationsaufwands ein Leichtes ist, einem IPSec-VPN neue Clients hinzuzufügen.
- seine weitgehende Plattformunanbhängigkeit: Es gibt Implementierungen unter Windows 2000 / XP, Linux, MacOS etc., und auf spezieller Hardware.

Daneben existieren aber eine Reihe von Schwachstellen, die sowohl einen breiten Einsatz erschweren als auch die beabsichtigte Sicherheit unterminieren. In ihrer kryptographischen Analyse von IPSec listen [FS] etliche Mängel auf. Als ein wichtiger Kritikpunkt erscheint die Tatsache, dass die Protokollsuite sehr kompliziert ist: Es gibt den Tunnel- und den Transportmodus, AH und ESP können parallel eigesetzt oder miteinander kombiniert werden, ESP kann verschlüsseln, ohne zu authentifizieren, bzw. authentifizieren, ohne zu verschlüsseln, ISAKMP Phase 1 kennt den Aggresiv- und den Main Mode, dieser wiederum hat vier verschiedene Methoden der Authentifizierung, davon zwei verschiedene mit öffentlichen Schlüsseln, usw. Diese Komplexität erhöhe die Gefahr von Implementierungsfehlern gewaltig. Außerdem seien die RFCs für IPSec teilweise unübersichtlich, unvollständig und an manchen Stellen sogar widersprüchlich [6]. Doch obwohl IPSec weit von Perfektion entfernt sei, sei es das derzeit beste Sicherheitsprotokoll auf IP-Ebene.

Ungeachtet seiner Schwächen findet IPSec heute in vielen Firmen und Universitäten Verwendung. Im universitären Umfeld kommen oft Softwarelösungen auf Linuxbasis wie z.B. Linux FreeS/WAN ([Freeswan]) zum Einsatz. FreeS/WAN wie Linux sind kostenlos zu erwerben, und die benötigte Hardware ist nicht teuer, oft sogar in Gestalt von gebrauchten Rechnern vorhanden. Dem steht jedoch ein nicht unerheblicher Konfigurationsaufwand gegenüber. Zudem hat jede Software ihre Lücken in Bezug auf die Korrektheit der Implementierung und auf die Sicherheit, was eine regelmäßige Wartung erforderlich macht. In Firmen sind die Anforderungen an die Sicherheit und einfache Bedienbarkeit in der Regel höher, so dass hier meist fertige Hardware-Produkte eingesetzt werden. Bei diesen kann IPSec mit weiteren Sicherheitsmechanismen, etwa für die Authentifizierung, kombiniert sein.

Die IPSec Working Group arbeitet derzeit an einer Weiterenwicklung, um einerseits bekannte Schwachstellen zu beseitigen und andererseits weitere Anforderungen zu erfüllen. Dazu zählt z. B. der Quality-of-Service Aspekt, der unter anderem bei einem Einsatz von IPSec für Voice-over-IP Verkehr eine wichtige Rolle spielt. Auch sollen Fortschritte auf dem Gebiet der Kryptographie und der Kryptanalysis berücksichtigt werden: DES soll keine Verwendung

[6] Zumindest dem ersten Punkt kann ich nach meinen Recherchen in Rahmen dieser Arbeit nur zustimmen!

mehr finden und dafür der *Advanced Encryption Standard (AES)* als Verschlüsselungsalgorithmus empfohlen werden.

So besteht die Hoffnung, dass IPSec mit zunehmender Verbreitung und nach erfolgter Verbesserung in einigen Punkten dazu beiträgt, das Netz ein wenig sicherer zu machen.

Abbildungsverzeichnis

Literatur

[DH] Diffie, W. und M. Hellman: *New Directions in Cryptography*; IEEE Transactions on Information Theory, V. IT-22, n. 6, Juni 1977

[DH00] Doraswamy, Naganand und Dan Harkins: *IPSec - Der neue Sicherheitsstandard für das Internet, Intranets und virtuelle private Netze*; Deutsche Übersetzung: Stefan Landvogt, Andrea Wutzer Addison-Wesley Verlag, 2000

[Freeswan] *Linux FreeS/WAN*; Internet: http://www.freeswan.org

[FS] Ferguson, Niels und Bruce Schneier: *A Cryptographic Evaluation of IPSec*; ohne Jahresangabe, siehe http://www.counterpane.com/ipsec.html

[HC98] Harkins, D. und D. Carrel: The Internet Key Exchange (IKE); RFC 2409, November 1998

[IANA] *Internet Assigned Numbers Authority*; Die Datenbank ist zu finden unter: http://www.iana.org/numbers.html

[KA98a] Kent, S. und R. Atkinson: *Security Architecture for the Internet Protocol*; RFC 2401, November 1998

[KA98b] Kent, S. und R. Atkinson: *IP Authentication Header*; RFC 2402, November 1998

[KA98c] Kent, S. und R. Atkinson: *IP Encapsulating Security Payload*; RFC 2406, November 1998

[Kra98] Krawczyk, H.: *SKEME: a versatile secure key exchange mechanism for Internet*; aus dem IEEE Symposium on Network and Distributed System Security, 1996

[KBC97] Krawczyk, H., M. Bellare und R. Canetti: *HMAC: Keyed-Hashing for Message Authentication*; RFC 2104, February 1997

[MD98] Madson, C. und N. Doraswamy: *The ESP DES-CBC Cipher Algorithm with Explicit IV*; RFC 2405, November 1998

[MG98a] Madson, C. und R. Glenn: *The Use of HMAC-MD5-96 within ESP and AH*; RFC 2403, November 1998

[MG98b] Madson, C. und R. Glenn: *The Use of HMAC-SHA-1-96 within ESP and AH*; RFC 2404, November 1998

[MSSJ98] Maughan, D., M. Schertler, M. Schneider und J. Turner: *Internet Security Association and Key Management Protocol (ISAKMP)*; RFC 2408, November 1998

[Orm98] Orman, H.: *The OAKLEY Key Determination Protocol*; RFC 2412, November 1998

[Pip98] Piper, D.: *The Internet IP Security Domain of Interpretation for ISAKMP*; RFC 2407, November 1998

Inhaltsverzeichnis