Benutzerhandbuch - Deutsch

FUJITSU

IPSec V1.4

Internet-Sicherheit im BS2000/OSD

Ausgabe November 2010

Kritik… Anregungen… Korrekturen…

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an manuals@ts.fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2008

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2008 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright und Handelsmarken

Inhalt

1 Einleitung

Die Liefereinheit openNet Server enthält das Transportsystem des BS2000/OSD. Der BS2000/OSD Communication Manager BCAM unterstützt neben den proprietären NEA-Protokollen auch das ISO-Protokoll sowie die TCP/IP-Protokolle IPv4 und IPv6.

Ab der Version 3.0 unterstützt openNet Server das IP Security-Protokoll IPSec, das umfangreiche Sicherheitsmechanismen für den verbindungslosen, paketvermittelnden IP-Verkehr bereitstellt.

1.1 Kurzbeschreibung von IPSec (IP Security Protocol)

Das IP-Protokoll selbst gewährleistet Datensicherheit nur in einem sehr eingeschränkten Umfang. Um die Integrität der Datenpakete sicherzustellen, berechnet das IP-Protokoll die so genannte 16 Bit-Header-Prüfsumme. Dieser Mechanismus ist unzulänglich, da er es Hackern und anderen unbefugten Personen leicht macht,

- IP-Adressen zu fälschen (IP-Spoofing),
- Inhalte von IP-Paketen während des Transfers zu lesen und zu ändern,
- bereits abgeschickte IP-Pakete abzufangen und erneut einzuspielen (Replay).

Demgegenüber bietet IPSec für IPv4- und IPv6-Pakete einen umfassenden, hochwertigen, interoperablen und auf kryptografischen Algorithmen basierenden Schutz.

Damit sind folgende Aspekte der Datensicherheit gewährleistet:

- Zugriffskontrolle
- Integrität des verbindungslosen IP-Datentransfers
- Authentifizierung des Ursprungs der IP-Pakete
- Schutz vor wiederholtem Einspielen von Datenpaketen (Anti-Replay)
- Datenvertraulichkeit
- Verhinderung von Verkehrsfluss-Analysen

Die von IPSec unterstützten Sicherheitsmechanismen sind innerhalb der Netzwerkschicht des TCP/IP-Protokoll-Stack realisiert.

1.2 Zielgruppen des Handbuchs

Das vorliegende Handbuch wendet sich an folgenden Leserkreis:

- Netz-Administratoren
- Entwickler von Netz-Anwendungen im BS2000/OSD
- Alle, die sich für Fragen der Internet-Sicherheit, insbesondere im Umfeld des BS2000/OSD, interessieren.

Kenntnisse des Betriebssystems BS2000/OSD, der TCP/IP-Grundbegriffe sowie von kryptografischen Sicherheitsmechanismen werden vorausgesetzt.

1.3 Lizenzrechtliche Bestimmungen

Die folgenden Copyright-Vermerke betreffen die Programme Racoon2, SPMD und CTRL-PROG.

---

iked/ikev1/evt.h

```
/*
 * Copyright (C) 2004 Emmanuel Dreyfus
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the project nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE PROJECT AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE PROJECT OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 */
```

---

iked/ikev1/genlist.c
iked/ikev1/genlist.h
iked/ikev1/ikev1_natt.c
iked/ikev1/ikev1_natt.h

```
/*
 * Copyright (C) 2004 SuSE Linux AG, Nuernberg, Germany.
 * Contributed by: Michal Ludvig <mludvig@suse.cz>, SUSE Labs
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
```

1.4 Wegweiser durch das Handbuch

Das Handbuch gliedert sich

- in einen allgemeinen Teil (Kapitel 2 -4), der über das Konzept von IPSec informiert, und
- in einen BS2000/OSD-spezifischen Teil (Kapitel 5 und 6), der die IPSec-Implementierung im BS2000/OSD sowie Installation, Konfiguration und Generierung des IPSec-Subsystems beschreibt.

Der erfahrene Leser kann die Kapitel 2-4 überblättern und direkt mit Kapitel 5 beginnen.

Im Einzelnen behandeln die Kapitel dieses Buchs die folgenden Themen:

- Kapitel 2 gibt einen Überblick über die Bedrohungen für die Internet-Sicherheit und die Maßnahmen zur Abwehr der damit verbundenen Gefahren.
- Kapitel 3 streift kurz die Historie der IPSec-Entwicklung und stellt Vorteile und Nutzen von IPSec dar.
- Kapitel 4 erläutert das Konzept der Sicherheitsarchitektur der Internet Protokolle und beschreibt die zugehörigen Datenbanken Security Associations-Datenbank (SAD) und Security Policies-Datenbank (SPD), erläutert die Protokoll-Elemente Authentication Header (AH) und Encapsulating Security Payload (ESP) in den Transfermodi Transportmodus und Tunnelmodus und die prinzipiellen Verarbeitungsabläufe in IPSec.
- Kapitel 5 beschreibt die IPSec-Implementierung im BS2000/OSD.
- Kapitel 6 beschreibt Installation, Konfiguration und Administration des IPSec-Subsystems im BS2000/OSD sowie die Nutzung und Konfiguration der Programme Racoon2 und SPMD.
  Kapitel 6 enthält außerdem eine Kurzanleitung zum IPSec Betrieb, um den Einstieg zu erleichtern.
- Kapitel 7 enthält Informationen zu den Meldungen von IPSec.
- Der Anhang liefert weiterführende Beschreibungen zur Security Associations-Datenbank (SAD) und Security Policies-Datenbank (SPD), zu den Protokoll-Elementen Authentication Header (AH) und Encapsulating Security Payload (ESP) in den Transfermodi Transportmodus und Tunnelmodus, zur Struktur des AH und die Abwicklung des AH-Protokolls, sowie zur Struktur des ESP und der Abwicklung des ESP-Protokolls.
- Das Fachwortverzeichnis erläutert Begriffe, die im Zusammenhang mit IPSec von Bedeutung sind.
- Das Abkürzungsverzeichnis erläutert Abkürzungen, die in diesem Handbuch verwendet werden.

1.5 Änderungen gegenüber IPSec V1.3

Neue Funktionen

IPSec unterstützt ab IPSec V1.4 NAT-Traversal, DNS und IPCompression für IKEv2.

Aktuelle RFC-Basis

Der Implementierungsstand wurde auf die aktuelle RFC-Basis gebracht.

IKEv1-Implementierungen müssen folgende Funktionalität erfüllen (RFC4109):

- Verschlüsselungsalgorithmus 3DES-CBC
- Hash-Algorithmen SHA-1
- Authentifikation mit einem gemeinsamen geheimen Schlüssel (preshared secret key)
- Diffie-Hellman Group 2

IKEv2-Implementierungen müssen folgende Funktionalität erfüllen (RFC4307):

- Verschlüsselungsalgorithmus 3DES-CBC
- Hash-Algorithmen SHA-1
- Authentifikation mit einem gemeinsamen geheimen Schlüssel (preshared secret key)
- Diffie-Hellman Group 2

IPSec-Beispiele

Für NAT-Traversal und DNS wurden Beispiele aufgenommen.

1.6 Änderungen gegenüber IPSec V1.2

Entfallene Abschnitte und Funktionen

- Der bisherige Abschnitt 3.4 „IPSec-RFCs“ entfällt.
- Das automatische Erzeugen von Schlüsseln entfällt (Operand KEY-VALUE=*AUTOMATIC in der KEY- und SIGNATURE-Anweisung).
- Der Operand MODE=IKE(..) in der SECURITY-ASSOCIATION-Anweisung entfällt.

TU-Subsysteme Racoon2 und SPMD

IPSec unterstützt ab IPSec V1.3 neben dem IKE-Protokoll der Version 1 (IKEv1) auch das IKE-Protokoll der Version 2 (IKEv2), das durch die TU-Subsysteme Racoon2 und SPMD realisiert ist.

IPSec-Konfigurationsdatei

Der POLICY Satz wurde geändert:

- Bei IPv4 und IPv6-Adressen wird die Angabe der Präfix-Länge unterstützt.
- Der Operand COMPRESSION wurde neu eingeführt.

IPSec-Administrationskommandos

LOAD-IPSEC-SECRETS entfällt.

IPSec-Beispiele

Bestehende Beispiele wurden der neuen Implementierung angepasst und neue Beispiele wurden aufgenommen.

Meldungen

- geänderte Meldungen:
  YIS0004
  YIS0401
  YIS0402
  YIS0403
- gelöschte Meldungen:
  YIS0404
  YIS0409

1.7 Änderungen gegenüber IPSec V1.1

Die Struktur des Handbuchs zu IPSec V1.2 wurde im Vergleich zu IPSec V1.1 teilweise umgestellt, was zu einer kompakteren Darstellung führt. Untergeordnete Details wurden in den Anhang verlagert.

Außerdem wurde Folgendes geändert:

TU-Subsystem Pluto

IPSec unterstützt ab IPSec V1.2 das IKE-Protokoll, das durch das TU-Subsystem Pluto realisiert ist.

IPSec-Konfigurationsdatei

In der IPSec-Konfigurationsdatei wurden neue Steueranweisungen eingeführt:

- INCLUDE
- FLUSH
- ADD
- DELETE

Außerdem wurden ein neuer Konfigurationssatz eingeführt und die vorhandenen Konfigurationssätze geändert:

- Der PARTNER-SAS-Satz wurde neu eingeführt.
- Der KEY-Satz wurde wie folgt geändert:
  Das Schlüsselwort KEY-VALUE unterstützt den Wert *AUTOMATIC.
- Der SIGNATURE-Satz wurde wie folgt geändert:
  Das Schlüsselwort SIGNATURE-VALUE unterstützt den Wert *AUTOMATIC.
- Der SECURITY-ASSOCIATION-Satz wurde wie folgt geändert:
  das Schlüsselwort MODE unterstützt den Wert IKE und damit verbunden LIFETIME und CHECK-SEQUENCE-NUMBER.
- Der POLICY-Satz wurde geändert.

IPSec-Administrationskommandos

Das Kommando LOAD-IPSEC-SECRETS wurde neu eingeführt.

Meldungen

- geänderte Meldungen:

  YIS0218
  YIS0251
  YIS0252
  YIS0253
  YIS0405
  YIS0406
  YIS0407

- Neue Meldungen:

  YIS0220
  YIS0254
  YIS0255
  YIS0256

1.8 Darstellungsmittel

Typographische Gestaltungsmittel

In diesem Handbuch werden folgende Mittel zur Darstellung von funktional wichtigen Textteilen verwendet:

für Hinweistexte

ACHTUNG!

für Warnhinweise

kursive Schrift
: kennzeichnen Variable, wenn Sie dafür Werte einsetzen müssen.

`dicktengleiche Schrift`
: für die Darstellung von Eingaben für das System, Systemausgaben und für Dateinamen in Beispielen.

kommando
: In der Syntaxbeschreibung für Kommandos werden diejenigen Bestandteile (Bezeichnungen von Kommandos und Parametern) fett dargestellt, die unverändert eingegeben werden müssen.

► kennzeichnet Eingaben und Arbeitsschritte, die der Anwender durchführen muss.

Syntax der Kommandobeschreibung

Für die Syntaxbeschreibung einiger IPSec-Kommandos wird eine SDF-angenäherte Darstellung gewählt. Die Zeichen und Darstellungsformen sowie die Datentypen sind in den beiden folgenden Tabelle erläutert.

Kennzeichnung	Bedeutung	Beispiele
GROSSBUCHSTABEN	Großbuchstaben bezeichnen Schlüsselwörter (Kommando-, Anweisungs-, Operandennamen, Schlüsselwortwerte) und konstante Operandenwerte. Schlüsselwortwerte beginnen mit *.	**HELP-SDF** **SCREEN-STEPS** = ***NO**
GROSSBUCHSTABEN in Halbfett	Großbuchstaben in Halbfett kennzeichnen garantierte bzw. vorgeschlagene Abkürzungen der Schlüsselwörter.	**GUID**ANCE**-MODE** = ***Y**ES
=	Das Gleichheitszeichen verbindet einen Operandennamen mit den dazugehörenden Operandenwerten.	**GUID**ANCE**-MODE** = ***NO**
< >	Spitze Klammern kennzeichnen Variablen, deren Wertevorrat durch Datentypen und ihre Zusätze beschrieben wird.	**SYNTAX-F**ILE = <filename 1..54>
Unterstreichung	Der Unterstrich kennzeichnet den Default-Wert eines Operanden.	**GUID**ANCE**-MODE** = ***NO**
/	Der Schrägstrich trennt alternative Operandenwerte.	**NEXT-FIELD** = ***NO** / ***Y**ES
(...)	Runde Klammern kennzeichnen Operandenwerte, die eine Struktur einleiten.	,**UNGUID**ED**-DIA**LOG = ***Y**ES (...) / ***NO**
[ ]	Eckige Klammern kennzeichnen struktureinleitende Operandenwerte, deren Angabe optional ist. Die nachfolgende Struktur kann ohne den einleitenden Operandenwert angegeben werden.	**SELECT** = [***BY-ATTR**IBUTES](...)

Kennzeichnung	Bedeutung	Beispiele
Einrückung	Die Einrückung kennzeichnet die Abhängigkeit zu dem jeweils übergeordneten Operanden.	,**GUID**ED-**DIA**LOG = ***YES** (...) ***YES**(...) **SCREEN-STEPS** = ***NO** / ***YES**
|	Der Strich kennzeichnet zusammengehörende Operanden einer Struktur. Sein Verlauf zeigt Anfang und Ende einer Struktur an. Innerhalb einer Struktur können weitere Strukturen auftreten. Die Anzahl senkrechter Striche vor einem Operanden entspricht der Strukturtiefe.	**SUP**PORT = ***TAPE**(...) ***TAPE(...)** | **VOL**UME = ***ANY**(...) | ***ANY**(...) | | ...
,	Das Komma steht vor weiteren Operanden der gleichen Strukturstufe.	**GUID**ANCE-**MODE** = ***NO** / ***Y**ES ,**SDF-COM**MANDS = ***NO** / ***Y**ES
list-poss(n):	Aus den list-poss folgenden Operandenwerten kann eine Liste gebildet werden. Ist (n) angegeben, können maximal n Elemente in der Liste vorkommen. Enthält die Liste mehr als ein Element, muss sie in runde Klammern eingeschlossen werden.	list-poss: ***SAM** / ***ISAM** list-poss(40): <structured-name 1..30> list-poss(256): ***OMF** / ***SYSLST**(...) / <filename 1..54>
Kurzname:	Der darauf folgende Name ist ein garantierter Aliasname des Kommando- bzw. Anweisungsnamens.	**HELP-SDF** Kurzname: **HPSDF**

Datentyp	Zeichenvorrat	Besonderheiten
alphanum-name	A…Z 0…9 $, #, @	
command-rest	beliebig	

Datentyp	**Zeichenvorrat**	**Besonderheiten**
composed-name	A…Z 0…9 $, #, @ Bindestrich Punkt Katalogkennung	alphanumerische Zeichenfolge, die in mehrere durch Punkt oder Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann. Ist auch die Angabe eines Dateinamens möglich, so kann die Zeichenfolge mit einer Katalogkennung im Format :cat: beginnen (siehe Datentyp filename).
c-string	EBCDIC-Zeichen	ist in Hochkommata einzuschließen; der Buchstabe C kann vorangestellt werden; Hochkommata innerhalb des c-string müssen verdoppelt werden
filename	A…Z 0…9 $, #, @ Bindestrich Punkt	Eingabeformat: [:cat:][$user.] { datei | datei(nr) | gruppe | gruppe { (*abs) | (+rel) | (-rel) } } :cat: wahlfreie Angabe der Katalogkennung; Zeichenvorrat auf A…Z und 0…9 eingeschränkt; max. 4 Zeichen; ist in Doppelpunkte einzuschließen; voreingestellt ist die Katalogkennung, die der Benutzerkennung laut Eintrag im Benutzerkatalog zugeordnet ist. $user. wahlfreie Angabe der Benutzerkennung; Zeichenvorrat ist A…Z, 0…9, $, #, @; max. 8 Zeichen; darf nicht mit einer Ziffer beginnen; $ und Punkt müssen angegeben werden; voreingestellt ist die eigene Benutzerkennung. $. (Sonderfall) System-Standardkennung

Datentyp	Zeichenvorrat	Besonderheiten
filename (Fortsetzung)		#datei (Sonderfall) @datei (Sonderfall) \# oder @ als erstes Zeichen kennzeichnet je nach Systemparameter temporäre Dateien und Jobvariablen. datei(nr) Banddateiname nr: Versionsnummer; Zeichenvorrat ist A…Z, 0…9, $, #, @. Klammern müssen angegeben werden. gruppe Name einer Dateigenerationsgruppe (Zeichenvorrat siehe unter „datei“) gruppe{(*abs) (+rel) (-rel)} (*abs) absolute Generationsnummer (1..9999); * und Klammern müssen angegeben werden. (+rel) (-rel) relative Generationsnummer (0..99); Vorzeichen und Klammern müssen angegeben werden.
integer	0…9	+ bzw. - kann nur erstes Zeichen (Vorzeichen) sein.
name	A…Z 0…9 $, #, @	darf nicht mit einer Ziffer beginnen.

Datentyp	Zeichenvorrat	Besonderheiten
partial-filename	A…Z 0…9 $, #, @ Bindestrich Punkt	Eingabeformat: [:cat:][$user.][teilname.] :cat: siehe filename $user. siehe filename teilname wahlfreie Angabe des gemeinsamen ersten Namensteils von Dateien und Dateigenerationsgruppen in der Form: $name_1$.[$name_2$.[...]] $name_i$ siehe filename. Das letzte Zeichen von teilname muss ein Punkt sein. Es muss mindestens einer der Teile :cat:, $user. oder teilname angegeben werden.
text	beliebig	Das Eingabeformat ist den jeweiligen Operandenbeschreibungen zu entnehmen.
x-string	Sedezimal: 00…FF	ist in Hochkommata einzuschließen; der Buchstabe x oder X muss vorangestellt werden; die Anzahl der Zeichen darf ungerade sein.

1.9 Readme-Datei

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei. Sie finden die Readme-Datei auf Ihrem BS2000/OSD-Rechner unter dem Dateinamen `SYSRME.IPSEC.014.D`. Die Benutzerkennung, unter der sich die Readme-Datei befindet, erfragen Sie bitte bei Ihrer zuständigen Systembetreuung. Die Readme-Datei zeigen Sie mit dem Kommando `/SHOW-FILE` oder mit einem Editor an, oder Sie drucken sie auf einem Standarddrucker mit folgendem Kommando aus:

```
/PRINT-DOCUMENT SYSRME.IPSEC.014.D,LINE-SPACING=*BY-EBCDIC-CONTROL
```

2 Sicherheit im Internet

Durch das explosionsartige Wachstum des Internet und sein Vordringen in nahezu alle Bereiche des täglichen Lebens rücken auch die damit verbundenen Sicherheitsaspekte immer mehr in den Vordergrund:

- Kommunikationssicherheit, d.h. Datenauthentizität, Datenintegrität, Datenvertraulichkeit etc.
- Ausfallsicherheit, d.h. weitgehende Verfügbarkeit der beteiligten Systeme
- Schutz vor Viren, Würmern, Trojanischen Pferden, Backdoors u.a.

Gegenstand von IPSec und damit des vorliegenden Kapitels sind die Aspekte der Kommunikationssicherheit.

2.1 Gefahren für die Internet-Sicherheit

Bei den Bedrohungen für die Sicherheit im Internet lassen sich aktive und passive Angriffe unterscheiden.

Aktive Angriffe auf die Internet-Sicherheit

Zu den aktiven Angriffen auf die Internet-Sicherheit zählen:

- Sender-Adresse der Nachricht fälschen (Address Spoofing)
- Inhalt der Nachrichten fälschen
- Nachrichten abfangen und erneut einspielen (Replay)
- Reihenfolge der gesendeten Nachrichten ändern

Passive Angriffe auf die Internet-Sicherheit

Zu den passiven Angriffen auf die Internet-Sicherheit zählen:

- Nachrichteninhalte lesen
- Verkehrsfluss der Nachrichten analysieren:
  - Wer sind die Kommunikationspartner?
  - Nachrichtenaufkommen im zeitlichen Verlauf
  - Länge und Häufigkeit der Nachrichten

2.2 Maßnahmen zur Gewährleistung der Internet-Sicherheit

Zur Abwehr der im vorigen Abschnitt genannten Bedrohungen der Internet-Sicherheit gibt es eine breite Palette an Strategien und Mechanismen mit zum Teil unterschiedlichen Ansatzpunkten.

Bild 1 gibt einen Überblick über heute verfügbare Maßnahmen für die Internet-Sicherheit.

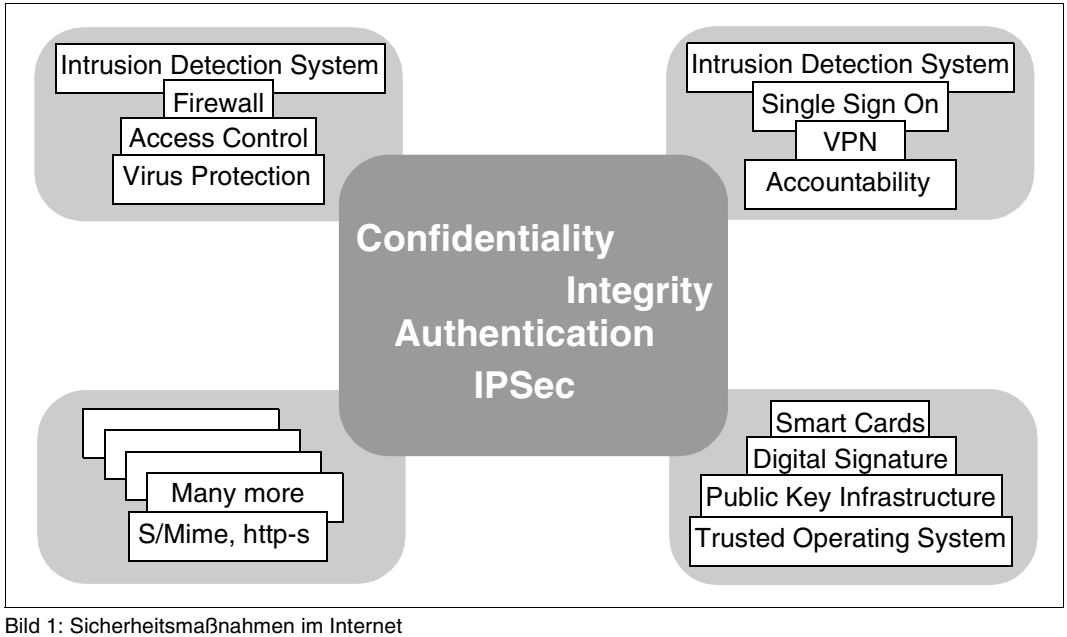


Bild 1: Sicherheitsmaßnahmen im Internet

2.2.1 Ziele der Sicherheitsmaßnahmen

Ziele dieser Maßnahmen sind im Einzelnen:

- Authentizität des Datenursprungs
  weist die angegebene Datenquelle als tatsächlichen Absender aus.
- Datenvertraulichkeit
  verhindert, dass Daten von nicht autorisierten Personen gelesen werden.
- Datenintegrität
  garantiert, dass Daten nicht verändert wurden.
- Anti-Replay
  verhindert, dass Daten von einem Eindringling abgefangen und anschließend wieder eingespielt werden.
- Vertraulichkeit des Verkehrsflusses
  verhindert die nicht autorisierte Analyse des Nachrichtenverkehrs.
- Zugriffschutz auf Teile des Netzwerks
  schützt diese Netzbereiche vor nicht autorisiertem Zugriff.

Mit IPSec lassen sich wahlweise einzelne oder auch alle diese Ziele realisieren. Dabei bietet IPSec hohe Flexibilität sowohl was die eingesetzten Sicherheitsmechanismen anbelangt als auch im Hinblick auf das Granulat des gesicherten IP-Verkehrs.

2.3 IPSec im Kontext der anderen Sicherheitsprotokolle

IPSec bietet seine Sicherheitsdienste auf der Netzwerk-Ebene (Ebene 3) des TCP/ IP-Protokoll-Stack an. Daneben gibt es eine Reihe weiterer standardisierter Sicherheitsprotokolle, die auf der Anwendungsschicht angesiedelt sind (Ebene 5 des TCP/IP-Stack bzw. Ebene 7 des OSI-Referenzmodells).

Folgende Tabelle zeigt Beispiele standardisierter Sicherheitsprotokolle.

Name	**Erläuterung**	**definiert durch**	**Netzwerkschicht TCP/IP (OSI)**
IPSec	Internet Protocol Security	IETF	L3
SSL	Secure Socket Layer	Netscape Communications	L5 (L7)
TLS	Transport Layer Security	IETF	L5 (L7)
S-HTTP	Secure Hypertext Transfer Protocol	Enterprise Integration Technologies, IETF	L5 (L7)
SET	Secure Electronic Transaction	Visa und Mastercard	L5 (L7)
HBCI	Homebanking Computer Interface	Bundesverband deutscher Banken	L5 (L7)

Sicherheitsstandards im Internet

Bild 2 zeigt die Anordnung der verschiedenen Sicherheitsprotokolle im TCP/IP-Protokoll-Stack.

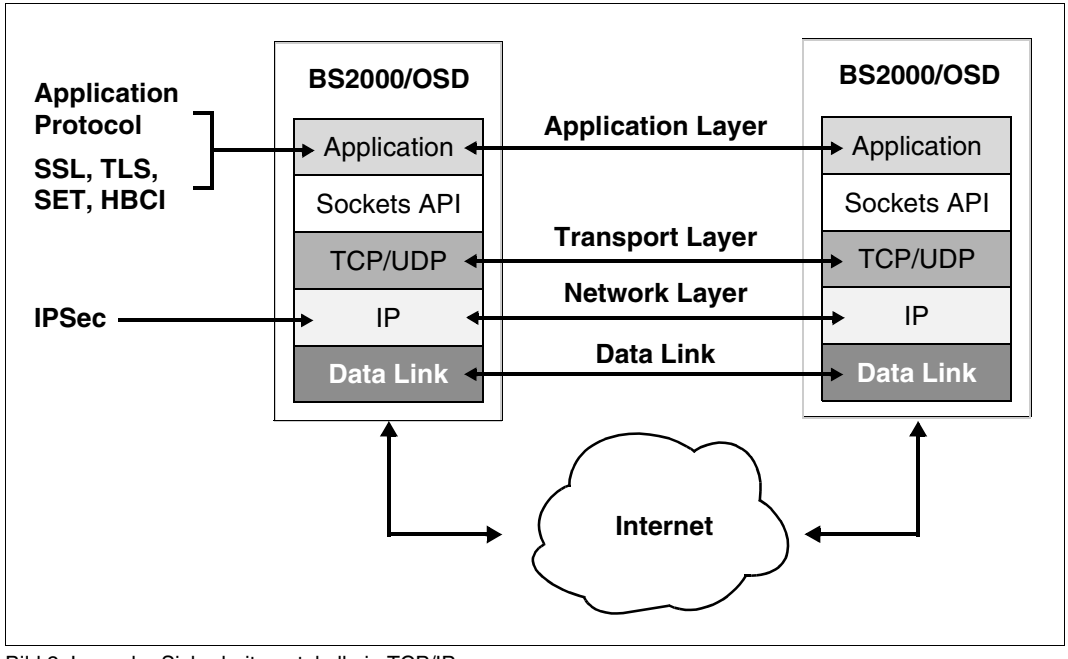


Bild 2: Lage der Sicherheitsprotokolle in TCP/IP

SSL - Secure Socket Layer

Secure Socket Layer (SSL) ermöglicht die gegenseitige Authentifizierung zweier kommunizierender Anwendungen und garantiert darüber hinaus Vertraulichkeit, Integrität und Authentizität der ausgetauschten Anwendungsdaten. So können Client-/Server-Systeme via SSL kommunizieren, ohne Gefahr zu laufen, dass die ausgetauschten Daten abgehört oder verfälscht werden.

Authentifizierung sowie Datenintegrität und Datenvertraulichkeit realisiert SSL mithilfe zweier untergeordneter Protokolle:

- SSL Record Protocol
- SSL Handshake Protocol

Das SSL Record Protocol definiert das für den Transfer der Daten verwendete Format. Das SSL Handshake Protocol ermöglicht es SSL-Client und SSL-Server, sich gegenseitig zu authentifizieren und Verschlüsselungsalgorithmen samt kryptografischem Schlüssel auszutauschen, bevor ein Protokoll der Anwendungsschicht die ersten Daten transferiert.

TLS - Transport Layer Security

Das Transport Layer Security-Protokoll V1.0 basiert auf dem Konzept von SSL. Wie bei SSL handelt es sich bei TLS um ein Protokoll der Ebene 5. Obwohl zwischen TLS V1.0 und SSL V3.0 keine wesentlichen Unterschiede bestehen, ist Interoperabilität zwischen den beiden Protokollen nicht ohne Weiteres möglich.

Derzeit wird diskutiert, inwieweit sich TLS zur zusätzlichen Sicherung von Anwendungsprotokollen wie FTP, SMTP oder LDAP einsetzen lässt, da SSL ausschließlich für das HTTP-Protokoll entwickelt wurde.

S-HTTP - Secure HTTP

Als Erweiterung des HTTP-Protokolls (HyperText Transfer Protocol) ermöglicht Secure HTTP (S-HTTP) den sicheren Datenaustausch im WWW. Jede Nachricht, die mit S-HTTP übertragen wird, kann durch eine beliebige Kombination aus Datenverschlüsselung, digitaler Unterschrift und Authentifizierung gesichert werden. Eine S-HTTP-Nachricht besteht aus einer gekapselten HTTP-Nachricht und einem Header, der das Format der gekapselten Daten beschreibt.

SET - Secure Electronic Transaction

Das Secure Electronic Transaction™-Protokoll SET ist ein offener technischer Standard, der von den Firmen Visa und Mastercard für die sichere Abwicklung des Zahlungsverkehrs im Internet entwickelt wurde.

HBCI - Homebanking Computer Interface

Der Internet Banking-Standard Home Banking Computer Interface (HBCI) ermöglicht die sichere Abwicklung von Bankgeschäften über das Internet. Grundlagen dieser Technik sind die Verschlüsselung aller Transaktionen mithilfe kryptografischer Verfahren. Die im HBCI verwendete Verschlüsselungstechnik ist eine elektronische Unterschrift für zuverlässigen Schutz gegen Hacker-Angriffe.

2.4 Firewalls

Die Internet-Protokolle wurden zunächst nicht unter Sicherheitsaspekten wie Authentizität, Integrität und Vertraulichkeit der zu übertragenden Daten entwickelt. Deshalb mussten Technologien wie die Firewalls entwickelt werden, mit deren Hilfe der Zugriff aus dem Internet auf das private lokale Netz und umgekehrt kontrolliert werden konnte. Firewalls kontrollieren den Datenverkehr hinsichtlich Ursprung, Ziel und Art sowie die Richtung der übertragenen Daten.

Dazu werden Regeln verwendet, die von Firewall-Systemen umgesetzt werden. Diese Regeln können auf der Netzwerkschicht (Paketfilter oder auch Screening-Router), der Verbindungsschicht (Circuit Level Gateways) oder auf der Anwendungsschicht (Application Level Gateways, ALG) wirken. In einer Firewall werden in der Regel Kombinationen dieser Filtertechniken eingesetzt.

3 Überblick über IPSec

IP Security (IPSec) bietet eine hochwertige, auf kryptografischen Mechanismen basierende Sicherheit für IPv4- und IPv6-Datagramme. Diese Mechanismen sind innerhalb der Netzwerk-Schicht (Network Layer) des TCP/IP-Protokoll-Stack realisiert.

Dieses Kapitel informiert über folgende Themen:

- Historie der IPSec-Entwicklung
- Vorteile und Nutzen von IPSec
- Sicherheitserweiterungen des IP-Protokolls durch IPSec
- IPSec-RFCs

3.1 Historie der IPSec-Entwicklung

Mit dem raschen Anwachsen des Internet rückten neben dem Problem der Vergabe von IP-Adressen auch vielfältige Aspekte der Internet-Sicherheit immer mehr in den Vordergrund. Zu nennen sind hier insbesondere

- Zugriffsschutz auf Teile des Netzes,
- Authentizität, Integrität, Vertraulichkeit der übermittelten Daten,
- Schutz vor wiederholtem Einspielen von Datenpaketen (Anti-Replay-Schutz) und
- Vertraulichkeit des Verkehrsflusses.

Bereits Anfang der neunziger Jahre war klar, dass eine Lösung für die genannten Probleme geschaffen werden musste. Neben der Verbreiterung des Adressraums und den Aspekten der Internet-Sicherheit waren Leistung und Vereinfachung des Routing weitere Aspekte der Forschungsarbeiten.

Das Ergebnis dieser Arbeiten war die Spezifikation für das Internet-Protokoll IPv6. Den gesteigerten Anforderungen an die Internet-Sicherheit wurde in IPv6 u.a. mit Einführung zweier neuer Header Rechnung getragen, dem Authentication Header (AH) und dem Encapsulating Security Payload Header (ESP). AH und ESP waren als so genannte Erweiterungsheader (Extension Header) Teil des Erweiterungsheader-Konzepts von IPv6, einer wesentlichen Neuerung gegenüber IPv4.

Da sich jedoch die Einführung von IPv6 entgegen den ursprünglichen Einschätzungen verzögert hat, mussten die für IPv6 vorgesehenen Sicherheitsmechanismen auch für IPv4 verfügbar sein. Zu diesem Zweck definierte die zur Internet Engineering Task Force (IETF) gehörende IP Security Working Group Sicherheitsmechanismen außerhalb der IPv6-RFCs. Zu diesen Sicherheitsmechanismen, die in eigenen RFCs definiert sind und gleichermaßen für IPv4 und IPv6 gelten, zählen insbesondere die Header AH und ESP.

3.2 Vorteile und Nutzen von IPSec

IPSec bietet umfassende Sicherheit auf der Ebene des IP-Protokolls, die sowohl für IPv4 als auch für IPv6 genutzt werden kann. Mithilfe zusätzlicher Header (Authentication Header und Encapsulating Security Payload Header) gewährleistet IPSec die Sicherheit des Datentransfers im Internet.

Folgende Vorteile zeichnen das IPSec-Protokoll aus:

- IPSec lässt sich gleichermaßen auf IPv4 und IPv6 anwenden.
- IPSec ist für Anwendungen transparent.
- IPSec ist unabhängig von anderen Sicherheitsmechanismen im Internet.
- IPSec definiert keine starre Sicherheitsarchitektur.
- IPSec ermöglicht die Definition einer variablen Sicherheitspolitik.

IPSec ist für Anwendungen transparent

Die Nutzung der IPSec-Sicherheitsmechanismen erfordert keine Änderungen an vorhanden Anwendungen, da die IPSec-Sicherheitsmechanismen auf administrativer Ebene in das System eingebracht werden.

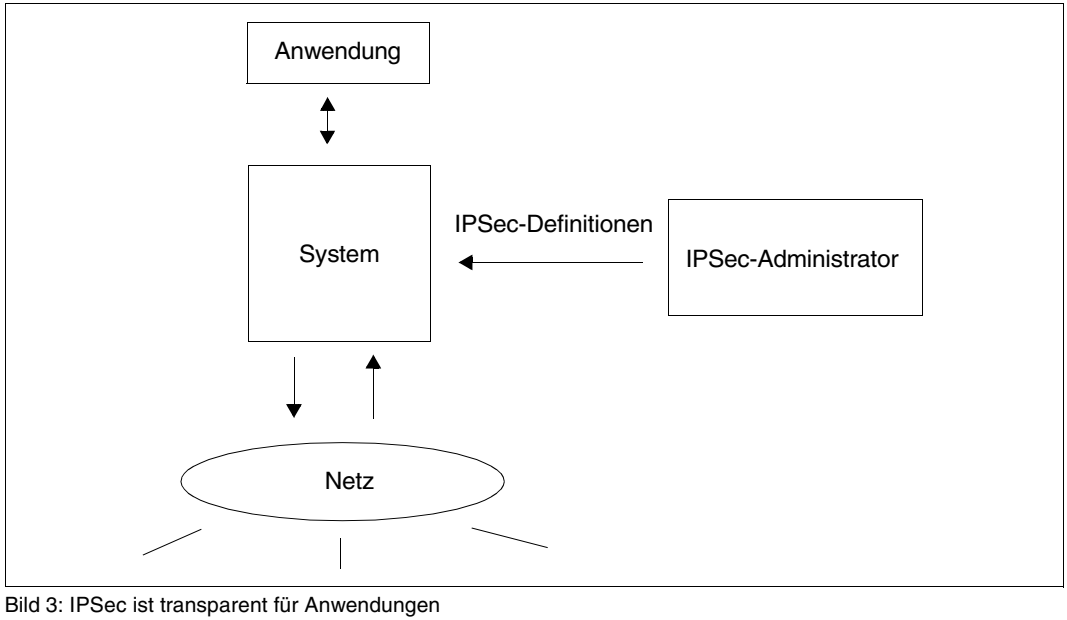


Bild 3: IPSec ist transparent für Anwendungen

IPSec ist unabhängig von anderen Sicherheitsmechanismen

Die Sicherheitsmechanismen von IPSec sind innerhalb des IP-Protokolls auf der Netzwerk-Ebene definiert. IPSec trifft keine Aussagen zu den anderen Ebenen der Kommunikation und definiert auch keine Abhängigkeit zu diesen Ebenen. Auf diese Weise lassen sich die IPSec-Sicherheitsmechanismen parallel mit anderen Sicherheitsmechanismen nutzen, und zwar kooperierend oder auch völlig unabhängig voneinander.

Das nachfolgend beschriebene Szenario gibt ein Beispiel für eine kooperierende Nutzung von IPSec und SSL.

Beispiel einer kooperierenden Nutzung von IPSec und SSL

Der FTP-Server eines BS2000/OSD-Systems ist via IPSec-Authentifizierung vor dem Zugriff unberechtigter Partner geschützt. Der vertrauliche Datentransfer zwischen FTP-Server und FTP-Client wird aber auf der Ebene einer SSL-Verschlüsselung abgewickelt. Ein Vorteil dieser Strategie gegenüber einer Authentifizierung auf SSL-Ebene besteht darin, dass „Denial of Service“-Attacken auf den FTP-Server bereits im IP-Layer, d.h. auf der Netzwerkschicht, abgefangen werden. Auf diese Weise werden hängende TCP-Verbindungen vermieden, die ein häufiges Hilfsmittel für „Denial of Service“-Attacken sind. Bei einer „Denial of Service“-Attacke versucht der Angreifer, legitimierte Benutzer bestimmter Services an der Nutzung dieser Services zu hindern, indem er beispielsweise die Netzverbindung zwischen Client- und Server-Rechner unterbricht.

IPSec definiert keine starre Sicherheitsarchitektur

IPSec definiert keine abgeschlossene Sicherheitsarchitektur. Eine solche Sicherheitsarchitektur wird oft als Public Key Infrastucture (PKI) bezeichnet. Vielmehr stellt IPSec mit den Protokollelementen Authentication Header (AH) und Encapsulating Security Payload Header (ESP) sowie mit den Konzepten der Security Association (SA) und Security Policy (SP) Mittel zur Verfügung, mit denen sich beliebige, auf spezifische Anforderungen zugeschnittene Sicherheitsarchitekturen gestalten lassen.

IPSec ermöglicht die Definition einer variablen Sicherheitspolitik

Die Protokollelemente Authentication Header (AH) und Encapsulating Security Payload Header (ESP) lassen sich beliebig mit den Transfermodi Transportmodus und Tunnelmodus kombinieren. Auf diese Weise ermöglicht IPSec eine breite Variabilität von Sicherheitskonzepten.

3.3 Sicherheitserweiterungen des IP-Protokolls durch IPSec

Ohne IPSec beschränkt sich die Integritätsprüfung für ein IP-Paket auf die Bildung der so genannten Header-Prüfsumme, die nur den IP-Header validiert und zudem leicht gefälscht werden kann.

IPSec erweitert die IP-Sicherheit um folgende Sicherheitsdienste:

- Zugangskontrolle
- Authentifizierung des Datenursprungs
- Integrität des verbindungslosen IP-Datentransfers
- Schutz vor wiederholtem Einspielen eines IP-Pakets (Anti-Replay)
- Datenvertraulichkeit
- Vertraulichkeit des Verkehrsflusses (teilweise)

IPSec realisiert diese Dienste mithilfe der folgenden Mechanismen:

- Konzept der Security Association
- Sicherheitsprotokolle Authentication Header (AH) und Encapsulating Security Payload Header (ESP) in den Transfermodi Transportmodus und Tunnelmodus
- Algorithmen für Authentifizierung und Verschlüsselung

Die IPSec-Sicherheitsdienste sichern die Datenkommunikation zwischen zwei Hosts, zwischen zwei Gateways sowie zwischen einem Security Gateway und einem Host. Dabei wird Interoperabilität gewahrt, d.h. die IPSec-Sicherheitsdienste haben keine negativen Auswirkungen auf Benutzer, Hosts und Netz-Komponenten, die IPSec nicht unterstützen.

Da IPSec seine Sicherheitsdienste auf der Ebene der Netzwerkschicht des IP-Protokoll-Stacks zur Verfügung stellt, können Protokolle und Dienste der höheren Schichten diese Dienste problemlos und ohne Änderung nutzen.

4 Die Sicherheitsarchitektur der Internet-Protokolle

IPSec bietet Systemen Sicherheitsdienste auf der Netzschicht im Rahmen des IP-Protokolls an. Um die Sicherheitsdienste erbringen zu können, werden bestimmte Ressourcen und Mechanismen benötigt. So muss die Möglichkeit der Wahl der Sicherungsverfahren und der Sicherheitsprotokolle gegeben sein. Die zu verwendenden Schlüssel und deren Gültigkeitsdauer muss ebenso festgelegt werden können wie der Mechansimus, mit dem der Partner das Schlüsselmaterial erhält. Es muss definiert werden, auf welche Daten die Dienste anzuwenden sind, also welche Datenströme zu welchen Partnern geschützt sein sollen.

Die Sicherheitsdienste können zwischen einem Paar von Endsystemen, einem Paar von Sicherheitsgateways oder einem Endsystem und einem Sicherheitsgateway vereinbart werden. In welchen Systemen die Sicherheitsdienste erbracht werden, hat Einfluss auf die Funktionsweise der Komponenten, welche die Dienste erbringen.

In den folgenden Abschnitten werden die zum Verständnis von IPSec notwendigen Begriffe eingeführt und die Komponenten der IPSec-Architektur vorgestellt.

Die BS2000/OSD Implementierung von IPSec orientiert sich an dieser Architektur, Abweichungen im Detail sind jedoch möglich und beabsichtigt.

4.1 Die Selektoren

Der im Kontext von IPSec verwendete Selektorbegriff ist eine Menge von Beschreibungsmitteln, mit der der Umfang des Datentransfers, der durch IPSec zu behandeln ist, erfasst werden kann. Mit Hilfe der Selektoren ist es möglich, diejenigen IP-Segmente auszuwählen, für die Sicherheitsfunktionen durch IPSec zu erbringen sind. Die Kombination der Selektoren zusammen mit ihren Werten bestimmen, ob die zu sichernde Datenmenge groß oder klein ist. Es wird von Granularität der Sicherheitsregeln gesprochen, die durch die Selektoren festgelegt wird. Sind die Selektoren z.B. so definiert, dass der gesamte Datenverkehr zwischen einem Paar von Systemen erfasst wird, spricht man von einer grob(körnig)en Granularität. Im Gegensatz dazu würde man von einer fein(körnig)en Definition sprechen, wenn der Datenverkehr zwischen zwei Anwendungen ausgewählt wird.

Die Vergleichswerte für die Selektoren müssen sinnvollerweise aus den Protokollfeldern der Pakete gewonnen werden können.

- Zieladresse (destination IP-address)
  Dieser Selektor bezieht sich auf die Adresse des innersten IP-Protokolls. IP-Adressen von möglichen IP-Tunnelprotokollen dürfen nicht zum Vergleich heran gezogen werden. Es kann eine einzelne Adresse (z.B. eine unicast address), ein Adressbereich, eine Adresse mit Netzmaske oder Präfixlänge, oder eine wildcard Adresse angegeben werden.
  Die Möglichkeiten der Adressangabe sind von der Implementierung abhängig und haben Einfluss auf den Komfort des Bedienungssystems. In jedem System muss jedoch die Zieladresse als Selektor unterstützt werden.
- Ursprungsadresse (source IP-address)
  Die zur Zieladresse gemachten Aussagen gelten analog.
- Namen
  Durch IPSec werden zwei Namenstypen berücksichtigt:
  - Benutzernamen in der DNS-Darstellung (full qualified user name) und in der X.500-Darstellung (distinguished name) und
  - Systemnamen, ebenfalls in der DNS- und X.500 Darstellung.

  Die Werte für den Namensselektor können nicht aus Protokollfeldern abgelesen werden. Es bedarf eines Mechanismus, der die Abbildung von der IP-Adresse auf den Namen vornimmt, z.B. einer DNS- bzw. einer X.500 Directory-Abfrage.
- Typ des Transportprotokolls
  Unter dem Transportprotokoll ist hier das Protokoll zu verstehen, das dem IP-Protokoll folgt (inklusive seiner Erweiterungen, zu denen auch die Sicherheitsprotokolle von IPSec zählen).

- Portnummer
  In Verbindung mit der Angabe TCP oder UDP im Transportprotokollselektor ist es durch die Angabe von Ziel- und/oder Ursprungsportnummer möglich, den Datenverkehr genauer zu spezifizieren. Die Angabe einer wildcard Portnummer sollte möglich sein.

- Grad der Daten-Vertraulichkeit (data sensitivity level)
  Kennzeichnung der Daten innerhalb eines Systems wird von Sicherheitskonzepten wie MAC und MLS ausgenutzt.

  Mandatory Access Control - MAC ist ein Konzept für die Kontrolle und Steuerung von Zugriffsrechten auf IT-Systeme, bei der die Entscheidung über Zugriffsberechtigungen nicht auf der Basis der Identität des Akteurs (Benutzers, Prozesses) und des Objektes (Datei, Gerät) gefällt wird, sondern aufgrund allgemeiner Regeln und Eigenschaften des Akteurs und Objektes. Auch erhalten häufig Programme eigene Rechte, die die Rechte des ausführenden Benutzers weiter einschränken. Voraussetzung ist, dass Akteure niemals direkt, sondern nur durch einen Referenzmonitor auf Objekte zugreifen können.

  Die Multi-Level Sicherheitssysteme (MLS) entsprechen der ursprünglichen Form der Mandatory Access Control.
  Bei der Bearbeitung der IP-Segmente werden geeignete Sicherheitsdienste ausgewählt, bzw. es wird auf die angewendeten Sicherheitsdienste hin überprüft.
  IPSec benötigt dafür Unterstützung, z.B. durch die IP-Protokollarchitektur, so dass die Datenmarkierungen übertragen werden können, und/oder eine Instanz, die zum richtigen Zeitpunkt, also beim Zugriff auf sensitive Daten und nach der Prüfung des Regelwerkes für das Objekt und den Akteur, dafür sorgt, dass die erforderlichen Schutzmechanismen eingerichtet werden.

Bei der Bearbeitung eines IP-Segmentes kann die Selektorinformation der höheren Protokollschicht unklar sein, z.B. wenn die Daten des Segments verschlüsselt sind. Die IPSec-Implementierung muss dies berücksichtigen.

4.2 Sicherheitsrichtlinie (Security Policy)

Bevor zwischen einem Paar von Systemen Sicherheitsdienste in Anspruch genommen werden können, müssen gewisse Vereinbarungen zwischen den Instanzen der beteiligten Systeme getroffen werden. Absprachen über den Umfang und die Art des Schutzmechanismus werden als Sicherheitsrichtlinie (Security Policy, kurz SP) bezeichnet. Eine Security Policy legt fest, wie die Datenpakete von und zu einem bestimmten Partner zu behandeln sind und welche Schutzmechanismen gegebenenfalls eingesetzt werden müssen.

Eine Security Policy beinhaltet eine Anweisung, wie die Pakete, auf die die Security Policy angewendet werden muss, zu behandeln sind. IPSec bietet drei Alternativen an:

- Bypass IPSec
  Die betroffenen IP-Segmente werden ohne weitere Bearbeitung durch IPSec an die nächste Instanz des Protokoll Stacks geleitet.
- Discard IPSec
  IP-Segmente, auf die die Discard Policy anzuwenden ist, werden nicht weiter verarbeitet, sondern gelöscht. Die Instanzen der benachbarten Protokollschichten werden darüber nicht informiert. Der entstehende Paketverlust muss durch eine höhere Protokollschicht erkannt werden. Eine Behebung oder Umgehung dieser Situation innerhalb IPSec ist nicht möglich.
- Apply IPSec
  Für die selektierten IP-Segmente sind Sicherheitsdienste vorgesehen.

Enthält die Security Policy die Apply Anweisung, beschreibt die Security Policy zusätzlich, welche Sicherheitsverbindung (Security Association) zu benutzen ist. Die Security Association wird durch folgende Angaben logisch spezifiziert:

- Sicherheitsprotokoll
- Transfermodus
- kryptografische Algorithmen

Security Policies müssen für diverse Suchalgorithmen organisiert sein. Schließlich muss anhand der Security Policies überprüft werden, ob Datenpakete korrekt behandelt wurden bzw. wie sie behandelt werden müssen. Diese Organisation der Security Policies bezeichnet man als Security Policy Database (SPD).

Eigenschaften der Security Policy Database (SPD)

Für die SPD und ihre Elemente sind folgende Eigenschaften gefordert:

- Vollständige Ordnung

  Die Einträge der SPD müssen so angeordnet sein, dass wiederholte Suchvorgänge mit dem gleichen Suchargument zum gleichen Eintrag führen. Das Suchargument besteht aus einem oder mehreren Selektoren, die aus Protokollfeldern der IPSec umgebenden Protokolle zusammengesetzt sind. Die Suche nach einer passenden Security Policy bei der Filterung des Datenverkehrs erfolgt immer durch die Angabe eines expliziten Wertes in den einzelnen Selektorfeldern, obwohl es explizit erlaubt ist, Security Policies mit wildcard Techniken und Wertebereichsangaben in einzelnen Selektoren zu definieren. Dadurch ergibt sich implizit die Anforderung an die Sortierung der SPD, dass genauer spezifizierte Einträge bei Suchvorgängen früher gefunden werden müssen als andere. Eine Suche wird mit dem ersten passenden SPD-Eintrag beendet.

- Ableitungsregeln für Security Associations (SAs)

  Security Policies beschreiben die zu verwendende(n) Security Association(s). Es ist von Bedeutung, wie SAs von der SP abgeleitet werden, wenn der Umfang des betroffenen Datenverkehrs durch Wildcards oder Bereichsangaben spezifiziert ist. Die SA Selektoren können vom einzelnen Datenpaket oder von den Selektoren der Security Policy abgeleitet werden.

- Verwaltung der Security Policy Database

  Die SPD muss eine administrative Schnittstelle bieten, damit Security Policies definiert, manipuliert und gelöscht werden können. Die Schnittstelle muss dem Systemverwalter zugänglich sein und kann Anwendungen angeboten werden, was eine Priorisierung der SP-Definition erfordert. Es muss festgelegt werden, ob es Anwendungen erlaubt ist, Security Policies des Systemverwalters außer Kraft zu setzen.

- Kombination von Security Associations

  Durch eine Security Policy können mehrere Security Associations miteinander verknüpft werden. Es entsteht dadurch ein so genanntes Security Association Bundle. Die Security Policy legt die Reihenfolge fest, in der die Sicherheitsdienste für den spezifizierten Datenverkehr zu erbringen sind.

- Default Regelung

  Immer wenn IPSec eingesetzt wird, regelt die Security Policy Database den gesamten Datenverkehr von und zu diesem System. Es ist deshalb von Interesse, was mit den Paketen passiert, für die keine Security Policy definiert ist. Die SPD muss die Konfigurationsoption bieten, für solche Pakete entweder die Discard- oder die Bypass-Aktion festzulegen.

4.3 Sicherheitsverbindung (Security Association)

Die Systeminstanzen, die die Sicherheitsdienste erbringen und die Beachtung der festgelegten Sicherheitsrichtlinie gewährleisten, verwenden zu diesem Zweck Sicherheitsverbindungen (Security Associations, kurz SA).

Eine SA ist gerichtet (unidirektional), d.h. die Parameter einer SA gelten nur für Pakete eines bestimmten Datenstroms zu einem Partnersystem bzw. für die Pakete von einem Partnersystem.

Eine SA beschreibt den Umfang des Sicherheitsdienstes, der von verschiedenen Kriterien abhängt. Eine wesentliche Einflussgröße ist das ausgewählte Sicherheitsprotokoll, durch das bestimmt wird, ob Vertraulichkeit und/oder Authentizität gewährleistet ist. Die Sicherheitsdienste einer SA werden durch genau ein Sicherheitsprotokoll erbracht. Ist für die Erfüllung einer Sicherheitsrichtlinie mehr als ein Sicherheitsprotokoll erforderlich, werden entsprechend mehrere SAs benötigt (SA Bundle).

Die Sicherheitsverbindung beschreibt, mit welchen kryptografischen Algorithmen die geforderte Sicherheit erbracht wird.

Sind im Rahmen des Sicherheitsprotokolls gewisse Dienste optional anwendbar, wie beispielsweise der Schutz vor dem wiederholten Einspielen von Paketen, wird die Information in der SA hinterlegt. Ebenso werden Konfigurationsparameter hinsichtlich der Schlüsselverwaltung, wie z.B. die Lebensdauer des Schlüssels oder das Austauschverfahren, abgespeichert.

Die Beschaffenheit des Schlüssels wiederum ist abhängig vom kryptografischen Verfahren und von der Güte des zu erbringenden Schutzes.

Abhängig von der Platzierung der IPSec Instanzen wird der Sicherheitsdienst im Transport- oder im Tunnelmodus erbracht. Diese Eigenschaft, auch als Operations- oder Transfermodus bezeichnet, wird als Parameter der SA hinterlegt. Dabei gilt:

- Eine SA im Transportmodus kann nur zwischen einem Paar von Endsystemen aufgebaut werden.
- Ist ein Security Gateway (SG) zur Erfüllung der geforderten Sicherheitsdienste involviert, muss eine SA im Tunnelmodus eingerichtet werden, da ein Security Gateway seine Sicherheitsdienste anstelle eines Endsystems anbietet und die Daten nach der Anwendung von Sicherheitsdiensten an das Endsystem weiterleitet.
- Andererseits bedeutet dies auch, dass ein Endsystem SAs sowohl im Transport- als auch im Tunnelmodus unterstützen muss. Daraus folgt, dass zwischen einem Paar von Endsystemen SAs im Tunnelmodus möglich sind.

Security Association Database (SAD)

Die Menge aller Sicherheitsverbindungen bildet die Security Association Database (SAD). Für die SAD ist im Gegensatz zur Security Policy Database keine Ordnung gefordert. Dennoch wird erwartet, dass anhand der verfügbaren Selektoren die geeignete SA gefunden wird, und das sollte sich auch nicht mit jedem Paket ändern. Es gibt einen weiteren wesentlichen Unterschied zur SPD hinsichtlich der Suchvorgänge. Während in der SPD für ankommende und ausgehende Pakete mit der gleichen Menge von Selektoren gesucht wird, müssen in der SAD für ankommende Pakete andere Suchargumente berücksichtigt werden. Eine Security Association für ankommende Pakete wird durch folgendes Tripel eindeutig identifiziert:

- Die Zieladresse, die im (äußeren) IP-Protokoll enthalten ist.
- Der Typ des Sicherheitsprotokolls, das nach dem IP-Protokoll folgt.
- Der Security Parameter Index (SPI), der im IPSec-Protokoll enthalten ist. Der SPI ist ein Bitstring und dient dazu, SAs bei gleicher Zieladresse und gleichem Sicherheitsprotokoll zu unterscheiden.

Die Gesamtheit der Security Associations bildet den aktuellen Dienstumfang eines IPSec-Systems ab. Ihre Lebensdauer ist begrenzt, sie ist zwar nicht an die Existenz einer Security Policy gebunden, aber nur funktionsfähig, wenn eine zugehörige Security Policy existiert. Mit Hilfe der Security Policy werden Dienste von IPSec definiert, mit Hilfe der Security Associations werden sie ausgeführt.

4.4 Kombinationen von Security Associations (SA Bundle)

Eine Sicherheitsarchitektur sollte flexibel gestaltet sein, um an die Bedürfnisse der Anwender anpassbar zu sein. Möglicherweise kann der geforderte Sicherheitsgrad nicht mit einer Schutzfunktion allein erbracht werden, sondern nur in Verbindung mit einem zusätzlichen Mechanismus. Existierende oder geplante Netz-Infrastrukturen mit Routern, Gateways, Firewalls und im Zusammenhang mit Sicherheitsdiensten Security Gateways bzw. Security Proxies erfordern ebenso Flexibilität der Protokollarchitektur. Dadurch wird die Entscheidungsfreiheit geboten, in welchen Systemen Sicherheitsfunktionen erbracht werden sollen und welche Funktionen das sind.

Es sind zwei Transfermodi verfügbar, ebenso können Security Associations kombiniert werden.

Transfermodus

Der Transfermodus bestimmt, welche Daten eines IP-Segments gesichert werden und wie das Sicherheitsprotokoll innerhalb des Protokollstacks positioniert wird.

- Transportmodus

  Die Sicherheitsprotokolle werden unmittelbar nach dem (End-to-End) IP-Protokoll eingefügt. (Für IPv6 heißt das, nach dem IPv6-Protokoll und den Erweiterungsprotokollen, die von Routern gebraucht werden, um ihre Dienste zu erbringen.) Im Transportmodus werden nur die Daten des IP-Segments und allenfalls die durch Router unveränderlichen Felder des IP-Protokolls gesichert.

  Die in IPSec definierten Sicherheitsprotokolle dürfen kombiniert werden. Dabei muss auf die Daten des IP-Segments zunächst der Verschlüsselungsdienst, danach der Authentifizierungsdienst angewendet werden.

  Security Associations im Transportmodus können nur zwischen Endsystemen eingesetzt werden.

- Tunnelmodus

  Im Tunnelmodus stellt das originale IP-Segment die Daten eines umschließenden IP-Segments dar. Das Sicherheitsprotokoll wird zwischen die beiden IP-Protokolle eingefügt. Die Sicherheitsfunktion wirkt also immer für ein komplettes IP-Segment. Security Associations im Tunnelmodus können auf ein End-to-End IP-Segment, das möglicherweise bereits durch eine Transport Security Association gesichert wurde, wiederholt angewendet werden.

Kombinationsmöglichkeiten

Für Endsysteme, im Folgenden Host 1 und Host 2 genannt, sind folgende Kombinationen zu unterstützen. Die Darstellung des logischen Protokollstacks erfolgt aus der Sicht von Host1.

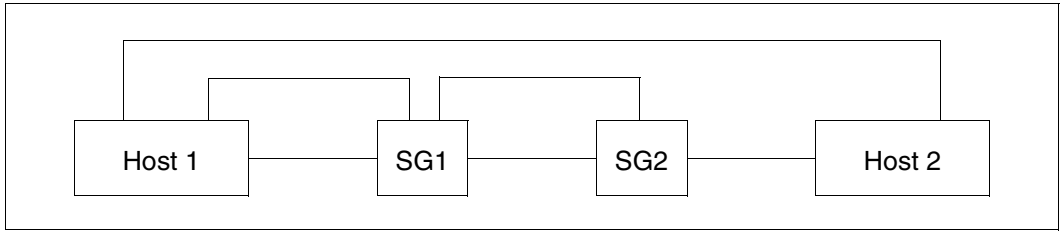


Bild 4: Darstellung von Endsystemen und Security Gateways (SG1, SG2)

Zwischen Host1 und SG1 sowie SG1 und SG2 sind folgende Kombinationen möglich (ULP bedeutet Upper Layer Protocol):

IP2	AH	IP1	ULP

IP2	ESP	IP1	ULP

Bild 5: Kombinationen zwischen Host und Security Gateway

Zwischen Host1 und Host2 sind folgende Kombinationen möglich:

IP	AH	ULP

IP	ESP	ULP

IP	AH	ESP	ULP

IP	AH	IP1	ULP

IP	ESP	IP1	ULP

Bild 6: Kombinationen zwischen zwei Endsystemen

Die SA zwischen den beiden Security Gateways SG1 und SG2 ist für die Endsysteme nicht sichtbar. Durch die Kombinationsvielfalt von SAs für Endsysteme entstehen jedoch komplexe Konfigurationen und Abläufe.

4.5 Zusammenspiel der Komponenten

Dieser Abschnitt erläutert das Zusammenspiel bei ausgehenden und eingehenden Daten.

4.5.1 Ausgehende Daten

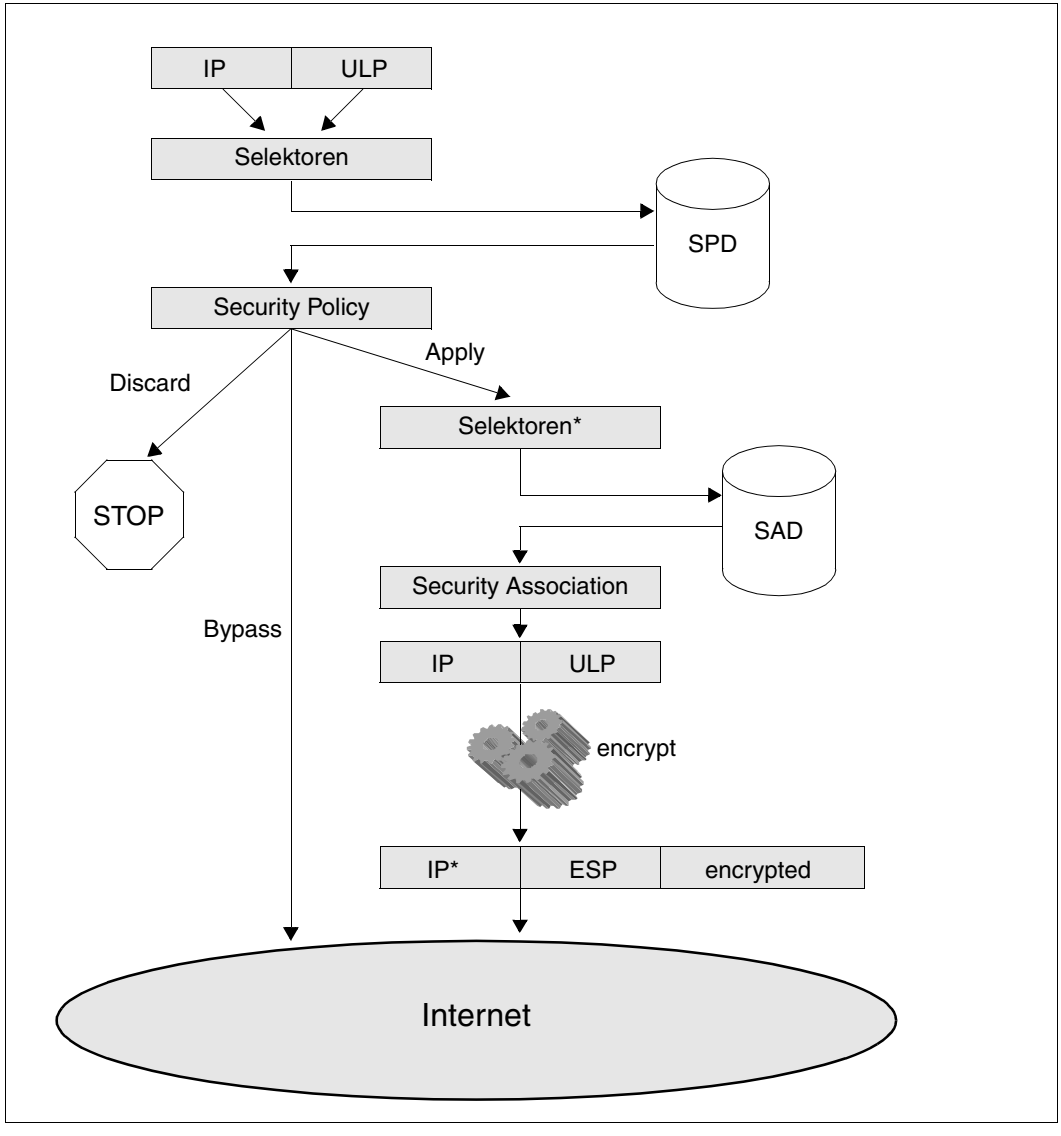


Bild 7: Zusammenspiel der Komponenten bei ausgehenden Daten

IPSec inspiziert jedes ausgehende IP-Segment, ob Sicherheitsdienste für dieses Segment gefordert sind. IPSec stellt sich aus den Protokollfeldern des IP-Headers und des Upper Layer Protocols (ULP) die Selektoren zusammen. In der Security Policy Database wird nach einer Policy gesucht, die möglichst exakt durch die Selektoren beschrieben ist. Die Security Policy enthält die Anweisungen für IPSec, wie das IP-Segment verarbeitet werden soll:

– Wenn die Security Policy die Discard Anweisung definiert, dann lässt IPSec das IP-Segment "verschwinden".

– Die Bypass Anweisung veranlasst IPSec zum Senden des IP-Segments ohne weitere IPSec-Aktion.

– Durch die Apply Anweisung wird IPSec aufgefordert, in der Security Associations Database nach einer möglichst exakt passenden outbound SA zu suchen. Die Selektoren sind dabei gemäß der SA Definition in der Security Policy anzupassen. So sind neben dem Sicherheitsprotokoll der Transfermodus, eventuell die IP-Adresse des Tunnelendpunktes, die Ableitungsregel für die SA usw. zu berücksichtigen. Nach einer erfolgreichen SA Suche ändert IPSec den ursprünglichen IP-Header, fügt das Sicherheitsprotokoll unmittelbar hinter dem IP-Header ein und wendet die in der SA festgelegte Transformation an, wobei der SA spezifische Schlüssel zum Einsatz kommt.

4.5.2 Ankommende Daten

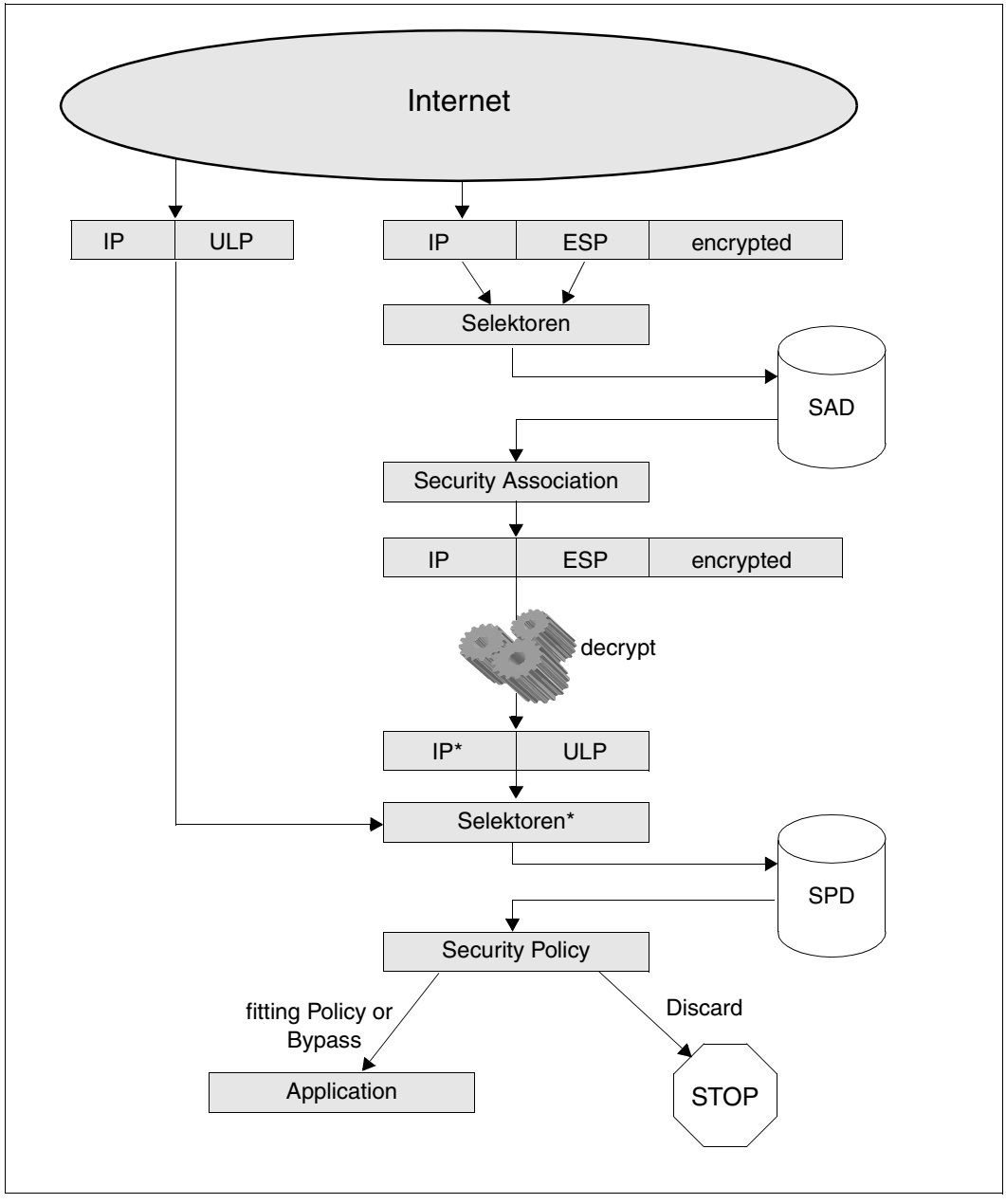


Bild 8: Zusammenspiel der Komponenten bei eingehenden Daten

Ankommende IP-Segmente müssen von IPSec zwangsläufig anders als ausgehende IP-Segmente behandelt werden. Einerseits gestaltet sich die Suche nach der angesprochenen inbound Security Association deutlich einfacher, anderseits sind erforderliche Selektorinformationen nicht lesbar, wenn die Daten des IP-Segments verschlüsselt sind. IPSec verwendet zur Suche der SA die IP-Zieladresse des Segments, den Typ des Sicherheitsprotokols und den im Sicherheitsprotokoll enthaltenen SPI. Nach der erfolgreichen SA-Suche wendet IPSec die in der SA festgelegte Transformation an, wobei der SA-spezifische Schlüssel zum Einsatz kommt. Danach wird das Sicherheitsprotokoll entfernt und der IP-Header angepasst.

Aus dem so neu gewonnenen IP-Segment bzw. dem ungeschützt angekommenen IP-Segment formt IPSec die Selektoren für die Suche in der Security Policy Database. Wird eine den Selektoren entsprechende Security Policy gefunden, überprüft IPSec, ob die geforderten Sicherheitsdienste mit denen des angekommenen IP-Segments übereinstimmen. Für empfangene IP-Segmente ohne Sicherheitsprotokoll stehen die Alternativen der Bypass und Discard Policy zur Verfügung, was zur Weiterverarbeitung bzw. zum Vernichten des Segments führt.

4.6 Sicherheitsprotokolle

Für die IPSec-Architektur sind zwei Sicherheitsprotokolle mit unterschiedlichen Sicherheitsfunktionen definiert. Die Sicherheitsprotokolle stellen die Schnittstelle zwischen den IPSec-Instanzen der Systeme dar. In den Protokollen sind die Sprachmittel vorhanden, die notwendig sind, um die Sicherheitsdienste zu erfüllen. Die Vereinbarung einer Security Association umfasst die Auswahl eines Sicherheitsprotokolls, die Anwendung optionaler Parameter innerhalb des Protokolls und die Mittel zur Identifikation der Security Association. Die beiden Protokolle haben innerhalb der IP-Protokollfamilie eigene Nummern.

4.6.1 Authentication Header

Die Sicherheitsdienste, die mit Hilfe des Authentication Header (AH) Protokolls erbracht werden können, sind:

- Authentifikation der Datenquelle, also des Senders.
- Schutz vor Daten(ver)fälschung.
- Schutz vor Replay Angriffen.

Der Sender errechnet eine Prüfsumme (Integrity Check Value, ICV), die in das AH-Protokoll eingetragen und so dem Empfänger mitgeteilt wird. Sender und Empfänger berechnen unter Verwendung des gleichen geheimen Schlüssels die Prüfsumme. Dadurch wird sichergestellt, dass die Daten unverfälscht sind und vom richtigen Sender kommen.

Der Replayschutz wird anhand einer Sequenznummer erbracht und ist optional. Der Dienst ist nur verfügbar, wenn die entsprechende SA nicht manuell konfiguriert wurde. Der Empfänger teilt es dem Sender während des dynamischen Aufbaus der SA mit, wenn er den Replayschutz nicht ausführen wird. Der Sender verhält sich entsprechend der Wahl des Empfängers.

AH hat die Protokollnummer 51.

IP-Paket, durch AH-Tunnelmodus geschützt

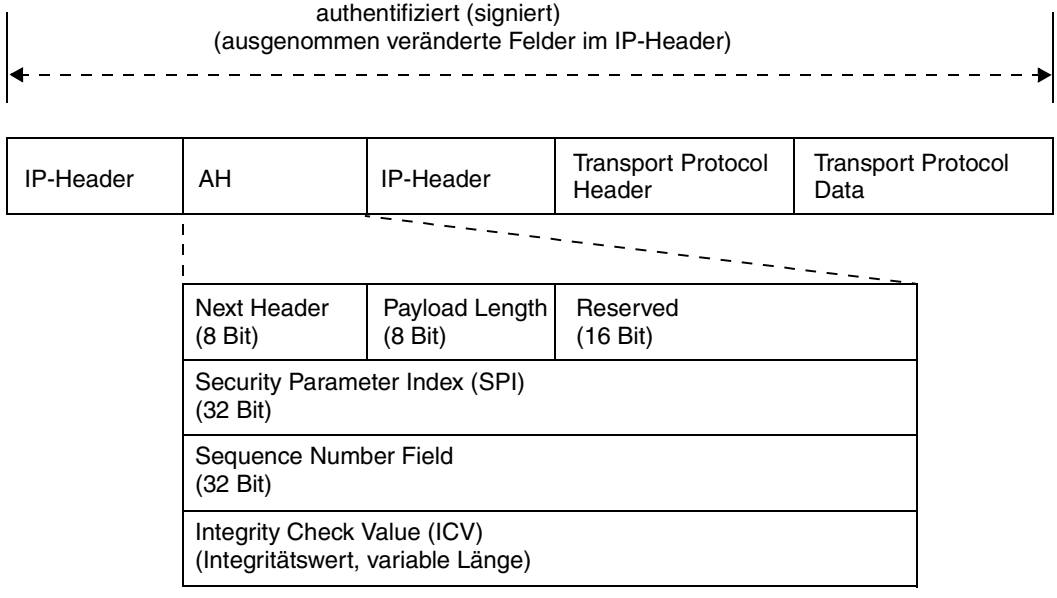


Bild 9: AH im Tunnelmodus

Bedeutung der AH-Protokollfelder

Next Header

Dieses Protokollfeld beinhaltet die Protokollnummer des in den Daten enthaltenen Protokolls.

Der Sender übernimmt beim Einfügen des AH diesen Wert aus dem entsprechenden Protokollfeld des Headers, der unmittelbar vor dem AH steht.

Der Empfänger überträgt den Inhalt dieses Feldes in den vor dem AH stehenden Protokollheader, wenn er den Authentication Header entfernt.

Payload Length

Dieses Feld zeigt die Länge des Authentication Headers an. Die Länge wird in vielfachen von 32 Bits dargestellt, wobei die ersten 64 Bits unberücksichtigt bleiben.

Reserved

16 Bits, die für Protokollerweiterungen vorgesehen sind. Solange sie nicht gebraucht werden, muss der Sender sie auf Null setzen und der Empfänger muss sie ignorieren.

Security Parameter Index (SPI)

Der Security Parameter Index ist der Identifikator der Authentication Header SA im Zielsystem. Der SPI wird in der sendeseitigen SA gespeichert. Er stellt für den Sender das Protokollsprachmittel dar, die (empfangsseitige) Sicherheitsverbindung im Zielsystem zu adressieren. Der Wertebereich von 0 bis 255 ist reserviert.

Der SPI wird beim Anlegen einer Security Association vergeben und dem Partner mitgeteilt. Die Vergabe erfolgt durch den Systemverwalter (bei manual keying) oder dynamisch durch Systeminstanzen.

Sequence Number

Die Sequenznummer dient dem optionalen Schutz vor wiederholtem Einspielen von IP-Segmenten. Die Empfangsseite einer SA entscheidet, ob der Dienst, abweichend vom Standardfall, abgewählt wird, und teilt dies dem Sender mit.

Beim Aufbau einer Sicherheitsverbindung wird ein entsprechender Zähler in der SA mit Null initialisiert. Der Sender erhöht für jedes Segment den Zähler und trägt den Wert in den Authentication Header ein, bevor die Prüfsumme berechnet wird - unabhängig davon, ob der Empfänger signalisiert hat, dass er die Sequenznummer nicht prüfen wird. Jedes Paket einer AH SA ist also mit einer fortlaufenden Sequenznummer versehen, die durch den Sicherheitsdienst geschützt ist.

Droht der Überlauf der Sequenznummer auf Null, ist das Verhalten des Senders davon abhängig, ob der Dienst vom Empfänger in Anspruch genommen wird. Der Sender muss eine neue SA vereinbaren, wenn der Dienst nicht abgewählt wurde. Ansonsten darf die Sequenznummer Null verschickt werden.

Der Empfänger prüft anhand der Sequenznummer zum einen, ob das IP-Segment dupliziert ist, und zum anderen, ob sich das Segment innerhalb seines Empfangsfensters befindet. Das Empfangsfenster hat eine konstante maximale Größe und wird mit dem Eintreffen des Segments mit der kleinsten Sequenznummer innerhalb des Empfangsfensters in Richtung höherer Nummern verschoben (sliding window, fixed size).

Die Überprüfung der Sequenznummer kann vor der Prüfung des ICV (siehe unten) erfolgen. Die Anpassung des Empfangsfensters muss nach der Bestätigung der Gültigkeit des Paketes erfolgen.

Hat sich der Empfänger gegen den Service entschieden, wird die Sequenznummer ignoriert.

Der Sender kennt die Größe des Empfangsfensters nicht.

Beim manuellen Einrichten von SAs macht der Anti-Replay Dienst keinen Sinn, da bei Überlauf der Sequenznummer keine neue SA bereitgestellt werden kann.

Der Anti-Replay Dienst kann nicht eingesetzt werden, wenn mehrere Sender an eine Empfangs-SA schicken, da keine Synchronisierung der Sequenznummern erfolgen kann.

Integrity Check Value (ICV)

In diesem Feld wird die vom Sender berechnete Prüfsumme übertragen. Die Länge der Prüfsumme ist abhängig vom zugrunde liegenden kryptografischen Algorithmus. Um der Längenanforderung der benutzten IP-Protokollversion (IPv4 oder IPv6) an den AH-Header gerecht zu werden, muss dieses Feld u.U. aufgefüllt werden (padding).

Der berechnete Integritätswert umfasst die Daten des Originalpaketes, bestimmte Informationen des Authentication Headers und die bei der Übertragung als unveränderlich betrachteten Felder des IP-Protokolls.

4.6.2 Encapsulating Security Payload

Die Sicherheitsdienste, die mit Hilfe des Encapsulating Security Payload (ESP) Protokolls erbracht werden können, sind:

- Vertraulichkeit.
- Verschleierung des Datenflusses.
- Authentifikation der Datenquelle, also des Senders.
- Schutz vor Daten(ver)fälschung.
- Schutz vor Replay Angriffen.

Die Vertraulichkeit der Daten wird durch ihre Verschlüsselung, also einer Transformation, erreicht. Die Verschlüsselung basiert auf einem gemeinsamen geheimen Schlüssel von Sender und Empfänger.

Die Vertraulichkeit und/oder Verschleierung des Datenflusses ist ein Nebeneffekt der Datentransformation, weil die Protokolle der höheren Protokollschichten und damit die enthaltenen Adressen nach der Transformation nicht lesbar sind. Mit ESP im Tunnelmodus kann zusätzlich verheimlicht werden, welche Systeme miteinander Daten austauschen.

Um die Authentifikation des Senders und die Integrität der Daten zu gewährleisten, bedient sich ESP im Prinzip der gleichen Algorithmen wie AH. Es wird eine Prüfsumme (Integrity Check Value, ICV) errechnet, die in das ESP-Protokoll eingetragen und so dem Empfänger mitgeteilt wird.

Sowohl der Verschlüsselungsdienst als auch der Authentifikationsdienst können abgewählt werden. Einer von beiden muss jedoch durch eine ESP SA erbracht werden.

Entscheiden sich die beiden IPSec-Instanzen im Quell- bzw. Zielsystem für beide Dienste, muss je eine ESP SA mit zwei Algorithmen und zwei Schlüsseln für jede Übertragungsrichtung vereinbart werden.

Der Replayschutz wird anhand einer Sequenznummer erbracht und ist optional. Da die Sequenznummer im ESP-Protokoll nur durch den Authentifikationsdienst erfasst wird, darf der Anti-Replay-Service nur in Verbindung mit dem Authentifikationsdienst gewählt werden. Der Dienst ist nur verfügbar, wenn die entsprechende SA nicht manuell konfiguriert wurde. Der Empfänger teilt es dem Sender während des dynamischen Aufbaus der SA mit, wenn er den Replayschutz nicht ausführen wird. Der Sender verhält sich entsprechend der Wahl des Empfängers.

ESP hat die Protokollnummer 50.

IP-Paket, durch ESP-Transportmodus geschützt

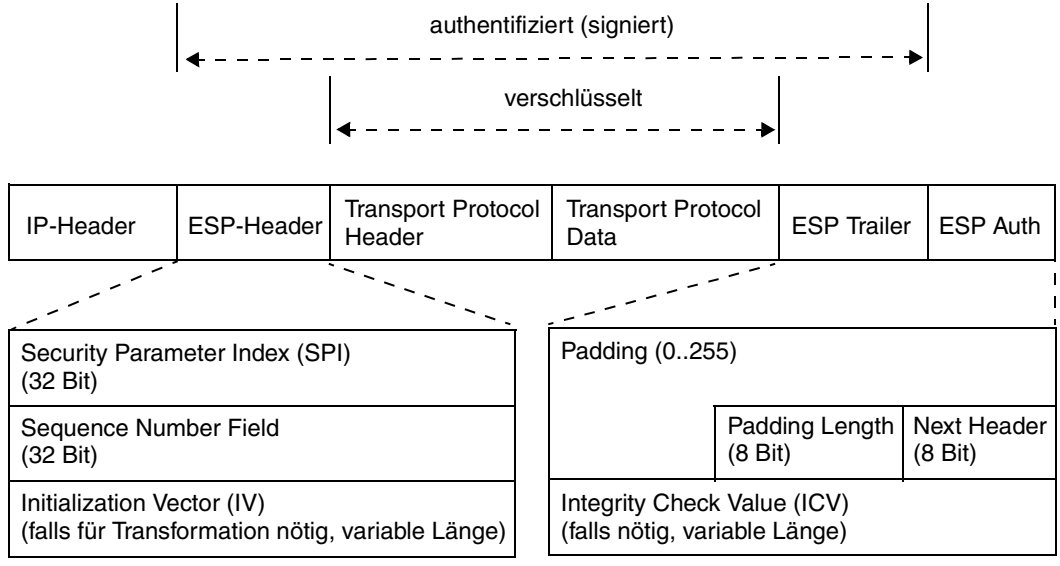


Bild 10: ESP im Transportmodus

Bedeutung der ESP-Protokollfelder

Security Parameter Index (SPI)

Der Security Parameter Index ist der Identifikator der Encapsulating Security Payload SA im Zielsystem. Der SPI wird in der sendeseitigen SA gespeichert. Er stellt für den Sender das Protokollsprachmittel dar, die (empfangsseitige) Sicherheitsverbindung im Zielsystem zu adressieren. Der Wertebereich von 0 bis 255 ist reserviert.

Der SPI wird beim Anlegen einer Security Association vergeben und dem Partner mitgeteilt. Die Vergabe erfolgt durch den Systemverwalter (bei manual keying) oder dynamisch durch Systeminstanzen.

Sequence Number

Die Sequenznummer dient dem optionalen Schutz vor wiederholtem Einspielen von IP-Segmenten.

Die Empfangsseite einer SA entscheidet, ob der Dienst, abweichend vom Standardfall, abgewählt wird und teilt dies dem Sender mit.

Beim Aufbau einer Sicherheitsverbindung wird ein entsprechender Zähler in der SA mit Null initialisiert. Der Sender erhöht für jedes Segment den Zähler und trägt den Wert in den ESP-Header ein, bevor die Prüfsumme berechnet wird und unabhängig davon, ob der Empfänger signalisiert hat, dass er die Sequenznummer nicht prüfen wird. Jedes Paket einer ESP SA ist also mit einer fortlaufenden Sequenznummer versehen, die durch den Sicherheitsdienst geschützt ist.

Droht der Überlauf der Sequenznummer auf Null, ist das Verhalten des Senders davon abhängig, ob der Dienst vom Empfänger in Anspruch genommen wird. Der Sender muss eine neue SA vereinbaren, wenn der Dienst nicht abgewählt wurde. Ansonsten darf die Sequenznummer Null verschickt werden.

Der Empfänger prüft anhand der Sequenznummer zum einen, ob das IP-Segment dupliziert ist, und zum anderen, ob sich das Segment innerhalb seines Empfangsfensters befindet. Das Empfangsfenster hat eine konstante maximale Größe und wird mit dem Eintreffen des Segments mit der kleinsten Sequenznummer innerhalb des Empfangsfensters in Richtung höherer Nummern verschoben (sliding window, fixed size).

Die Überprüfung der Sequenznummer kann vor der Prüfung des ICV erfolgen. Die Anpassung des Empfangsfensters muss nach der Bestätigung der Gültigkeit des Paketes erfolgen.

Hat sich der Empfänger gegen den Service entschieden, wird die Sequenznummer ignoriert.

Der Sender kennt die Größe des Empfangsfensters nicht.

Beim manuellen Einrichten von SAs macht der Anti-Replay Dienst keinen Sinn, da bei Überlauf der Sequenznummer keine neue SA bereitgestellt werden kann.

Der Anti-Replay Dienst kann nicht eingesetzt werden, wenn mehrere Sender an eine Empfangs-SA schicken, da keine Synchronisierung der Sequenznummern erfolgen kann.

Payload Data

Die transformierten (verschlüsselten) Daten des Originalpaketes werden im Payload Data Feld des ESP-Protokolls übertragen. Benötigt der verwendete kryptografische Algorithmus so genannte Synchronisationsdaten, werden diese zusätzlich im Payload Data Feld transportiert. Die Länge, die Struktur und die Position dieser

Information (z.B. eines Initialization Vector, IV) sind durch den Algorithmus bestimmt. Der Algorithmus definiert auch, wie diese Daten zur Entschlüsselung durch den Empfänger verwendet werden.

Padding Bytes

Die Padding Bytes im ESP-Header erfüllen verschiedene Funktionen. Es können bis zu 255 Bytes in das Protokoll eingetragen werden, dabei gilt:

- Die Padding Bytes werden zur Ausrichtung der nachfolgenden Protokollfelder verwendet.
- Der verwendete Transformationsalgorithmus arbeitet blockweise mit Vielfachen von N Bytes. Ein Auffüllen der Gesamtdatenlänge - bestehend aus Initialisierungsvektor, unverschlüsselten Daten und der zwei Bytes für die Felder Padding Length und Next Header - auf die nächste Blockgröße ist erforderlich.
- Es werden Padding Bytes zur Verschleierung der Länge des Originalpaketes eingefügt. Dies wäre dann ein Teil des Dienstes zum Schutz vor Datenflussanalyse.

Es existieren Regeln, wie die verwendeten Padding Bytes auszusehen haben, wenn deren Aussehen nicht durch den Transformationsalgorithmus definiert ist.

Padding Length

Gibt an, wieviele Padding Bytes unmittelbar vor diesem Feld eingefügt wurden.

Next Header

Dieses Protokollfeld beinhaltet die Protokollnummer des vom ESP-Header umschlossenen Protokolls.

Der Sender übernimmt beim Einfügen des ESP diesen Wert aus dem entsprechenden Protokollfeld des Headers, der unmittelbar vor dem ESP steht.

Der Empfänger überträgt den Inhalt dieses Feldes in den vor dem ESP stehenden Protokollheader, wenn der ESP-Header entfernt wird.

Integrity Check Value (ICV)

Das ICV-Feld ist optional vorhanden, wenn zwischen den beiden ESP Instanzen die Nutzung des Authentifikationsdienstes vereinbart wurde.

Der Sender trägt in dieses Feld den berechneten Wert ein. Der ICV wird nach der Verschlüsselung der Daten berechnet. Die Länge der Prüfsumme ist abhängig vom zu Grunde liegenden kryptografischen Algorithmus.

Der berechnete Integritätswert umfasst das ESP-Protokoll und dessen Daten. Das IP-Protokoll und das ICV-Feld gehen in die Berechnung nicht ein.

4.7 Kryptografische Verfahren

IPSec verwendet ausschließlich symmetrische kryptografische Verfahren in den Sicherheitsprotokollen Authentication Header und Encapsulating Security Payload. Sender und Empfänger wenden einen identischen Schlüssel auf die zu verabeitenden Daten an. Der gemeinsame geheime Schlüssel ist ein Parameter, der zwischen den Kommunikationspartnern vereinbarten Security Association.

Es gibt eine Reihe von Transformations- und Signaturalgorithmen, die in IPSec eingesetzt werden. Um als IPSec konform zu gelten, müssen bestimmte Algorithmen unterstützt werden.

Welche Algorithmen zu unterstützen sind, ist für IKEv1 ab dem RFC 2401 und für IKEv2 ab dem RFC 4301 beschrieben.

4.8 Verwaltung von Sicherheitsverbindungen und deren Schlüssel

4.8.1 Administrative Verwaltung

Die administrative (d.h. manuelle) Verwaltung von Sicherheitsdiensten wird durch Personen durchgeführt, ungeachtet der Möglichkeit, dass die Systemverwalter durch Maschinen unterstützt werden können. Die manuelle Verwaltung von Sicherheitsdiensten und deren ausführende Instanzen umfasst folgende Aktionen:

- Die Abstimmung der zu erbringenden Sicherheitsdienste
- Die Festlegung der kryptografischen Verfahren, die eingesetzt werden
- Das Erfinden der einzusetzenden Schlüssel
- Das Installieren einer Security Association
- Die Vergabe des Security Parameter Index einer SA
- Das Übermitteln dieser Informationen an den Partner, der die entsprechenden Aktionen durchzuführen hat

Die Übermittlung geheimer Schlüssel und deren Aufbewahrung sind besonders sicherheitssensitiv.

Administrativ eingerichtete Sicherheitsverbindungen bieten keinen Replay Schutz, da dieser einen Mechanismus für rechtzeitige Schlüsselwechsel erfordert, der durch den Systemverwalter nicht erbracht werden kann.

Trotzdem muss jedes IPSec-konforme System diese Art der Verwaltung ermöglichen. Durch diese Anforderung wird erreicht, dass für einzelne ausgewählte Datenübertragungsvorgänge die Anwendung eines Sicherheitsschutzes erzwungen werden kann.

Ein Sicherheitssystem kommt nie ganz ohne administrative Eingriffe aus, auch wenn die Verwaltung der Sicherheitsverbindungen, die dynamische Vergabe von (Protokoll-)Ressourcen und deren Verteilung automatisch durch Systeminstanzen wahr genommen wird. Die Sicherheitsrichtlinien (Security Policies) eines Systems legt nach wie vor der Systemverwalter in Absprache mit seinen Kollegen fest. IPSec fordert deshalb eine administrative Schnittstelle zur Security Policy Database.

4.8.2 Automatisierte Verwaltung

Die automatisierte Verwaltung, also der Auf- und Abbau von Security Associations, die Verhandlung der Attribute der Security Association, die Schlüsselerzeugung und -verteilung, ist in drei RFCs definiert.

- RFC2408 Internet Security Association and Key Management Protocol (ISAKMP) bildet einen abstrakt definierten Rahmen für solche Funktionen. ISAKMP ist nicht auf IPSec beschränkt, sondern kann seine Dienste beliebigen Sicherheitssystemen anbieten. ISAKMP ist unabhängig von konkreten Sicherheitsmechanismen, wie Verschlüsselungsverfahren und Authentifikationsverfahren spezifiziert. Innerhalb von ISAKMP können beliebige Key Exchange Protokolle angewendet werden, sofern sie die erforderlichen Eigenschaften haben.
- RFC2407 The Internet IP-Security Domain of Interpretation for ISAKMP legt die Nomenklatur, die Syntax und Semantik der Protokollfelder und deren Parameterwerte für die Key Management Protokolle (KMP) fest, die innerhalb von ISAKMP die von IPSec benötigten Funktionen und Sprachmittel zur Verfügung stellen.
- RFC2409 The Internet Key Exchange (IKE) stellt ein Protokoll dar, das die Anforderung von IPSec im Rahmen von ISAKMP erfüllt.

4.8.2.1 Internet Security Association and Key Management Protocol (ISAKMP)

ISAKMP, RFC 2408, definiert kein Protokoll, das den Austausch von Schlüsselmaterial zwischen den beteiligten Systemen regelt. ISAKMP ist kein implementierbares Protokoll. ISAKMP stellt den beteiligten Instanzen vielmehr ein System, einen Rahmen, zur Verfügung, der es ihnen erlaubt, sich auf ein Schlüsselaustauschverfahren und die damit verbundenen Parameter zu einigen. ISAKMP beschreibt in einer abstrakten Form, welche Anforderungen ein konkretes Protokoll hinsichtlich gewisser Aspekte erfüllen muss, um als konformes, für den Schlüsselaustausch geeignetes Protokoll gelten zu können.

Die abstrakte Definition eines konformen Protokolls umfasst die syntaktische und semantische Beschreibung der Protokollelemente (Payloads), die Protokollabläufe (Exchanges), in denen die Informationen ausgetauscht werden, sowie die Vorschriften der Payload Verarbeitung im Kontext eines Exchange. Es wird definiert, wie die Vorschläge für einen zu erbringenden Sicherheitsdienst auszusehen haben und wie einer dieser Vorschläge akzeptiert werden muss bzw. wie alle diese Vorschläge abgelehnt werden müssen. Neben dieser Protokollverhandlung sind Abläufe definiert, mit denen Sicherheitsdienste modifiziert und terminiert werden können.

ISAKMP befasst sich mit den Aspekten eines sicheren Verfahrens zum Austausch von Schlüsselmaterial, obwohl es unabhängig von den auszuführenden kryptografischen Mechanismen, dem verwendeten Schlüsselgenerierungsverfahren und dem Schlüsselaustauschprotokoll ist.

Die wesentlichen Anforderungen von ISAKMP an ein Key Exchange Protokoll sind:

- Es muss gewährleistet sein, dass die beteiligten Instanzen zu dem erforderlichen Satz an gemeinsamen, geheimen Schlüsselinformation gelangen.
- Die ausgetauschte Schlüsselinformation muss auf ihren Ursprung und Aktualität überprüfbar sein, d.h. die Schlüsselinformation muss authentifizierbar sein.
- Die Schlüsselinformationen unterschiedlicher Verhandlungen müssen voneinander unabhängig sein. Damit wird verhindert, dass bei der Aufdeckung eines geheimen Schlüssels mehrere Kommunikationsbeziehungen kompromittiert werden. Diese Eigenschaft wird als Perfect Forward Secrecy (PFS) bezeichnet.
- Um der großen Anzahl an potentiellen Kommunikationspartnern Rechnung zu tragen, sollte ein Mechanismus zur Verfügung stehen, der es ermöglicht, geheime Schlüsselinformationen zwischen Systemen zu erzeugen und auszutauschen, die kein gemeinsames Geheimnis installiert haben. Dies bedeutet die Integration einer Public Key Infrastructure (PKI).

Ein gemeinsamer geheimer Satz an Schlüsselinformation nutzt den beteiligten Instanzen nur in Verbindung mit einer gemeinsamen Vereinbarung, in welchen kryptografischen Algorithmen die Schlüssel zu verwenden sind, wie lange sie gültig sein sollen, wie sich die Kommunikationspartner auf die Schlüssel beziehen. Die Schlüssel selbst, o.g. Eigenschaften der Schlüssel, zusammen mit den Identitäten der Kommunikationspartner kann man als Sicherheitsverbindung betrachten.

ISAKMP erbringt seine Dienste in zwei unterschiedlichen Verhandlungsschritten, die als Phasen bezeichnet werden.

Phase 1

In Phase 1 einigen sich die ISAKMP Instanzen darüber, wie die folgenden Kommunikationsschritte innerhalb des ISAKMP zu sichern sind. Während der Phase 1 authentifizieren sich die Instanzen gegenseitig. Dadurch ist gewährleistet, dass geheim zu haltende Informationen nicht an Unberechtigte ausgeliefert werden. In Phase 1 wird eine Sicherheitsverbindung zwischen den ISAKMP Instanzen eingerichtet. Sie wird als ISAKMP SA bezeichnet und von den Instanzen selbst verwaltet. Die ISAKMP SA ist bidirektional, d.h. die beiden Instanzen haben dasselbe Wissen über die Sicherheitsparameter der SA und beide Instanzen können zu jedem Zeitpunkt die SA nutzen. Zwischen zwei Instanzen können mehrere ISAKMP SAs existieren. In der Phase 1 wird für jede ISAKMP SA implizit die Rolle der einzelnen Instanz festgelegt. Die aktive Instanz erfüllt die Rolle des Initiators, die passive Instanz erhält die Rolle des Responders. Das Protokoll hält Sprachmittel zur Identifikation einzelner SAs bereit. Die Rollen der Instanzen haben dabei eine wichtige Funktion, weil durch sie die Reihenfolge der Identifikatoren innerhalb der Protokollelemente festgelegt wird.

Phase 2

Phase 2 wird zur Errichtung von Sicherheitsverbindungen für andere Dienste angeboten. Unter den Schutzmechanismen der ISAKMP SA oder unter dem Schutz der Phase 1 können viele Sicherheitsverbindungen zwischen den beteiligten Systemen und Diensten eingerichtet werden.
In Phase 1 kommen rechenintensive Verfahren zur gegenseitigen Authentifikation der Instanzen (asymmetrische kryptografische Verfahren) und zur Generierung der Schlüssel für die ISAKMP SA zum Einsatz. Dies ist jedoch akzeptabel, da es sich um seltene Kommunikationsschritte handelt. Die wesentlich häufiger erforderlichen Verhandlungen über Sicherheitsdienste in Phase 2, wie z.B. den Aufbau einer IPSec-SA, können dafür deutlich günstiger erbracht werden, da sie unter dem Schutz der ISAKMP SA ablaufen.

Zusammenfassend lässt sich festhalten, dass die Verhandlungen über zu erbringende Sicherheitsdienste innerhalb von ISAKMP immer folgenden Inhalt haben:

- das Sicherheitsprotokoll, mit dem der Dienst erbracht wird
- die kryptografischen Algorithmen, die ausgeführt werden
- die Schlüsselinformationen zusammen mit ihren Attributen

Exchanges

ISAKMP definiert für beide Phasen Dialoge, sog. Exchanges, mit deren Hilfe die Dienste erbracht werden. Ein Dialog beschreibt die Anzahl und Reihenfolge der Protokollelemente und die Syntax, Semantik und Reihenfolge der Sprachmittel (Payloads) innerhalb der Protokollelemente. Die Payloads werden durch ein ISAKMP Protokoll zusammen gefasst. Im ISAKMP Protokoll stehen die Identifikatoren der SA, unter der der Dialog ablaufen soll.
ISAKMP definiert fünf Standard-Dialoge:

- Base Exchange
  umfasst vier Nachrichten und ermöglicht den Schlüsselaustausch und Authentifikation.
- Identity Protection Exchange
  umfasst sechs Nachrichten. Der Schlüsselaustausch erfolgt vor dem Austausch der Identifikations- und Authentifizierungsdaten, wodurch diese geschützt werden können.
- Authentication Only Exchange
  umfasst drei Nachrichten und stellt nur einen Authentifizierungsdienst ohne Verschlüsselung zur Verfügung.
- Aggressive Exchange
  umfasst drei Nachrichten, bietet Schlüsselaustausch, ISAKMP SA Aufbau ohne Verhandlungsoption und Authentifizierung.
- Informational Exchange
  umfasst eine Nachricht und dient dazu, SA-spezifische Information an die Partnerinstanz zu übermitteln.

ISAKMP kann alle Transportprotokolle nutzen. Die Unterstützung über UDP an Portnummer 500 muss jedoch gewährleistet sein.

4.8.2.2 Internet Key Exchange (IKE)

Das Internet Key Exchange (IKE, RFC2409) Protokoll ist ein häufig eingesetztes Protokoll mit den erforderlichen Mechanismen zur automatisierten Verwaltung von IPSec-Sicherheitsverbindungen. IKE bedient sich der Sprachregelung von IPSec-DOI und erfüllt die Anforderungen, die ISAKMP an Key Exchange und Management Protokolle stellt.

IKE verbindet Bestandteile zweier anderer Security Association und Key Management Protokolle, dem Oakley und dem SKEME Protokoll, und bildet diese auf das generisch definierte ISAKMP ab. IKE bedient sich daher der Begriffe aus allen Protokollspezifikationen.

IKE baut eine ISAKMP SA entweder in einer "Main Mode" oder in einer "Aggressive Mode" Verhandlung auf. Bei beiden handelt es sich um Phase 1 Exchanges, wobei "Main Mode" einem Identity Protection Exchange und "Aggressive Mode" einem Aggressive Exchange entspricht.

Die Erzeugung des geheimen Schlüsselsatzes basiert - unabhängig vom ausgewählten Mode - auf dem Diffie-Hellman Verfahren. Es sind vier Diffie-Hellman Gruppen vordefiniert. Andere Gruppen können verhandelt werden, wobei in diesen Verhandlungen die algebraischen Details spezifiziert werden müssen.

IKE unterstützt vier verschiedene Authentifikationsverfahren.

- Preshared Secret Key (PSK)
- Public Key Signature
- Public Key Encoding
- Revised Public Key Encoding

Die Authentifikationsdaten werden in Abhängigkeit des ausgewählten Authentifikationsmechanismus erzeugt. Bestimmte Felder der Protokollelemente werden ebenfalls in Abhängigkeit des Authentifikationsmechanismus interpretiert.

Ungeachtet dessen werden bei der Berechnung der Authentifikationsdaten bei jedem Verfahren dieselben Daten berücksichtigt.

Daneben sind andere Algorithmen und Verfahren zulässig und können zwischen den IKE-Instanzen ausgehandelt werden.

IKE führt für die Phase 2 einen neuen Dialog (Exchange) ein, den "Quick Mode". Quick Mode umfasst drei Protokollelemente. Die Vorschläge im Protokoll beziehen sich auf einzurichtende IPSec-SAs. Die Sicherheitsprotokolle sind also ESP und AH. Die Transformationen bzw. Signaturverfahren müssen von den IPSEC-Protokollen unterstützt werden. In die Berechnung des Schlüssels für die Transformation gehen u. a. der Protokolltyp und der Security Parameter Index (SPI) ein. Sowohl Initiator als auch Responder ermitteln je einen

SPI. Da der SPI in das Verfahren zur Schlüsselberechnung eingeht, sind die Schlüssel verschiedener IPSec-SAs voneinander unabhängig. IKE bildet für IPSec also die PFS Eigenschaft.

4.8.2.3 Internet Key Exchange Protocol Version 2 (IKEv2)

In der bisher unterstützten Version 1 des Internet Key Exchange Protocol (IKEv1) zeigten sich Schwächen hinsichtlich Zuverlässigkeit, Effektivität und Eindeutigkeit der Protokolldefinition. Das "Internet Key Exchange Protocol Version 2" erfüllt die gleichen Funktionen wie IKEv1, also die gegenseitige Authentifizierung der beteiligten Protokollinstanzen und den Aufbau und die Verwaltung von Security Associations.

IKEv2 ist in RFC4306 definiert. RFC4306 fasst die IKEv1 RFCs 2407, 2408, 2409 und eine Reihe weiterer RFCs zusammen. Beispiele hierfür sind RFC3748 "Extensible Authentication Protocol (EAP)", RFC3715 "IPsec-Network Address Translation (NAT) Compatibility Requirements" und RFC3948 "UDP Encapsulation of IPsec ESP Packets". Die Protokollversion 2 ist in der Absicht entstanden, die Schwächen der Version 1 zu eliminieren. Das Protokoll sollte dennoch einfacher gestaltet werden und eine höhere Flexibilität in der praktischen Anwendbarkeit erhalten. Obwohl die grundsätzlichen Funktionen beider Protokollversionen übereinstimmen, ist IKEv2 mit IKEv1 nicht verträglich. Dies findet Ausdruck in der Anzahl, den Inhalten und der Semantik der Protokollelemente der Protokollphasen 1 bzw. 2 sowie in der neu gewählten Nomenklatur. Beide Protokollversionen werden durch Protokollinstanzen bearbeitet, die durch die UDP-Portnummer 500 identifiziert werden. RFC4306 legt die Regeln fest, wie sich die Protokollinstanzen auf eine Version einigen müssen.

4.8.2.4 Änderungen in IKEv2 gegenüber IKEv1

IKEv2 hat eine Reihe von Änderungen gegenüber IKEv1 erfahren, damit die angestrebten Ziele erreicht werden können. Einige wesentliche Aspekte werden im Folgenden angeführt.

Kompatibilität

Der IKE Protocol Header enthält die beiden Felder "major version number" und "minor version number", anhand derer die Protokollinstanzen erkennen können, welche Protokollversion und welche Funktionalität dieser Version der jeweilige Partner unterstützt. Die "minor version number" dient lediglich der Anzeige von Funktionalität. Die Instanz, die eine kleinere Versionsnummer unterstützt, muss die größere des Partners ignorieren. Die Instanz, die eine größere Versionsnummer unterstützt, erkennt die geringere Funktionalität des Partners und berücksichtigt dies während der Verbindungsdauer. So wird beispielsweise verhindert, dass Informational Exchanges eingeleitet werden, die der Partner nicht verstehen kann.

Versionen mit unterschiedlichen "major version numbers" sind inkompatibel. Die "major version number" zeigt die Protokollversion des aktuellen Paketes an. Zusätzlich existiert ein Anzeigefeld, in dem der Sender mitteilen kann, dass er eine höhere Protokollversion unterstützt. Erkennen die Protokollinstanzen anhand dieses "ich könnte eine höhere Version Flags" die Möglichkeit, eine höhere Protokollversion zu vereinbaren, wird der laufende Verbindungsaufbau abgebrochen und ein neuer Verbindungsversuch mit höherer Version gestartet. Empfängt ein Responder jedoch einen Request mit einer höher Versionsnummer als die höchste, die er unterstützt, darf er die Nachricht nicht bearbeiten. Er sollte den Initiator darüber informieren, wobei gleichzeitig die höchste akzeptable Version mitgeteilt wird.

Die Regeln für die Verhandlung der Protokollversion gelten ab IKEv2 (RFC4306) und sollen helfen, zukünftige IKE-Protokollversionen kompatibel einzuführen. Die Verhandlung zwischen IKEv1 und IKEv2 Instanzen ist nicht definiert. Zum einen fehlt in IKEv1 das Flag, mit dem die Unterstützung einer höheren Version angezeigt werden kann. Zum anderen kann im Informational Exchange, mit dem ein Request wegen einer zu großen Versionsnummer abgewiesen wird, die höchste unterstützte Version nicht mitgeteilt werden. Eine mögliche Umgehung dieses Migrationsproblems wäre, zunächst einen Request mit "major version number 2" zu schicken und nach Ablauf einer Überwachungszeit bzw. nach dem Empfang einer ablehnender Antwort den Request mit "major version number 1" zu wiederholen, sofern dies durch den Systemverwalter erlaubt ist. Ansonsten muss konfiguriert werden, welche Protokollversion zu den einzelnen Partnerinstanzen zu verwenden ist und akzeptiert werden darf.

Als weiteres Protokollsprachmittel für zukünftige Erweiterungen enthält jeder Payload Header ein "critical flag", mit dem der Absender anzeigt, ob der Inhalt des Payload ignoriert werden darf. Wird ein IKEv2 Paket empfangen, das einen unbekannten Payload mit "critical flag" enthält, darf das gesamte Paket nicht bearbeitet werden. Auf diese Weise wird gewährleistet, dass Protokollerweiterungen innerhalb einer Version möglich sind.

Zuverlässigkeit

Informationen werden vom Initiator eines Dialogs immer in Request Protokollelementen übertragen, die vom Responder beantwortet werden müssen. Responses werden anhand einer zu spiegelnden Sequenznummer dem auslösenden Request zugeordnet.

Erhält der Initiator innerhalb einer festgelegten Zeitspanne keine Quittung, wird der Request wiederholt, bis die Aktion (erfolgreich bzw. erfolglos) abgeschlossen ist.

Die Protokollsequenznummer erlaubt es, mehrere Aktionen gleichzeitig durchzuführen.

Lebensdauer einer SA

In IKEv2 ist definiert, wann die Protokollinstanzen ihren Verbindungspartner als nicht verfügbar betrachten müssen und wie solche Situationen erkannt werden können (Dead Peer Detection). IKEv1 kennt keine Dead Peer Detection. RFC3706 "A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers" macht jedoch einen Vorschlag, wie in IKEv1 die Erreichbarkeit eines Partners festgestellt werden kann. Durch die DPD ist es den Verbindungspartnern möglich, zu jedem Zeitpunkt die IKE-SA und ihre untergeordneten CHILD-SAs zu löschen. Systemneustarts sind daher jederzeit möglich, da gewährleistet ist, dass der Verbindungspartner dies erkennt und definierte Maßnahmen ergreift, um eine Synchronisation beider Seiten herbeizuführen.

Rekeying

Das Rekeying einer SA ist jedem Verbindungspartner zu jedem Zeitpunkt erlaubt. Die Lebensdauer einer SA kann durch die Protokollinstanzen unabhängig voneinander festgelegt werden. Eine Aushandlung bzw. ein Abgleich der SA Lebensdauer ist in IKEv2 nicht vorgesehen. Der Ablauf zur Erneuerung eines veralteten Schlüssels ist genau definiert. Das Rekeying in IKEv2 wird durch das Ersetzen einer SA durch eine neu verhandelte SA bewerkstelligt.

Semantische Vereinfachung der Protokolle

Die zu verschlüsselnden Teile einer IKEv2 Nachricht werden als Daten des Encrypted Payload übertragen. Der Encrypted Payload muss der letzte Payload eines IKEv2 Protokolls sein. Er beinhaltet ein zusätzliches Signaturfeld. Die Signatur erstreckt sich über das gesamte Protokoll, einschließlich des IKEv2 Headers. Das Layout des Encrypted Payload Headers ist dem eines ESP Trailers sehr ähnlich.

In IKEv1 wurde anhand des Headers entschieden, ob die dem Header folgenden Daten verschlüsselt sind. Unverschlüsselte Protokollsprachmittel sind nicht möglich. Die Signatur für eine Integritätsprüfung wird in einem speziellen (HASH) Payload übertragen, der ebenfalls verschlüsselt ist.

Die Vereinbarung der Partnerinstanzen darüber, welche Datenströme durch eine IPsec SA zu schützen sind, ist starr und unflexibel gestaltet. IKEv1 lässt zwar die Angabe von IP Adressbereichen zu, beim Aufbau der SA kann der Responder die vom Initiator vorgeschlagenen Adressen dann aber lediglich akzeptieren oder ablehnen. Eine weitere Auswahl oder Einschränkung ist ihm nicht möglich. Portnummernbereiche für die Protokolle TCP und UDP bzw. die genauere Spezifikation von ICMP Nachrichten sind nicht darstellbar. Für eine erfolgreiche SA Verhandlung bedeutet dies, einen höheren Konfigurationsaufwand in den Initiator- und Responder-Systemen. Andererseits provoziert es eine höhere Fehlerrate in den Konfigurationen.

In IKEv2 wird den Kommunikationspartnern die Möglichkeit einer echten Aushandlung des zu schützenden Datenverkehrs gegeben. Mit Hilfe des Protokollsprachmittels "Traffic Selector (TS) Payload" kann der Initiator einer Sicherheitsverbindung eine Liste von Protokoll/Port/IP-Address Beschreibungen angeben, wobei Port- und Adressbereiche möglich sind. Dem Responder ist es gestattet, einen Abgleich der empfangenen TS Liste und der lokalen Konfiguration vorzunehmen. Zusätzliche Sprachmittel im IKEv2 Protokoll wie Hinweise, dass weitere Adressbereiche akzeptiert werden bzw. dass nur ein einzelnes Adresspaar unterstützt wird, erhöhen die Erfolgsaussichten der Verhandlung.

Die Aushandlung einer Security Association wurde in IKEv2 einfacher und dadurch effizienter gestaltet. Der Initiator beschreibt jede zu verhandelnde SA durch eine Liste von Vorschlägen, den Proposal Payloads. Proposal Payloads sind dem Security Association Payload untergeordnet. Proposal Payloads außerhalb eines SA Payloads sind nicht erlaubt. Jeder Vorschlag enthält eine Liste von gewünschten Sicherheitsprotokollen (z.B. AH, ESP), und zu jedem Sicherheitsprotokoll wird eine Auswahl an Verschlüsselungs- bzw. Signaturalgorithmen angeboten. Diese Algorithmen sind in Transform Payloads beschrieben, die ihrerseits einem Proposal Payload untergeordnet sind. Dem Responder ist es erlaubt, einen der Vorschläge auszuwählen, wobei er für jedes Sicherheitsprotokoll den aus seiner Sicht am besten geeigneten Transformationsalgorithmus festlegt. Eine Besonderheit ist dabei, dass für das IPsec Protokoll ESP eine Liste von Verschlüsselungs- und Signaturalgorithmen angegeben werden kann, aus der der Responder je einen auswählen darf. Wird keiner der Vorschläge akzeptiert, ist die Security Association abzulehnen.

Obwohl der Inhalt eines SA payloads in IKEv1 dem in IKEv2 sehr ähnlich ist, ist seine Interpretation deutlich unterschiedlich. In IKEv1 sind die Proposal- und Transform Payloads eigenständige Strukturen, neben dem Security Association Payload. In IKEv1 muss für das IPsec Protokoll ESP jeder Verschlüsselungsalgorithmus mit jedem Signaturalgorithmus kombiniert und in einen eigenen Transform Payload eingetragen werden. Die Anzahl der Transformationen steigt daher exponentiell mit der Anzahl Algortihmen.

Das IP payload compression protocol (RFC3173) verliert in IKEv2 den Status eines IPsec Protokolls. IPCOMP ist dadurch in IKEv2, im Gegensatz zu IKEv1, nicht als einzelnes Protokoll bzw. als Protokoll einer Protection Suite (eines SA bundles) verhandelbar. Die Unterstützung von IPCOMP zusammen mit den Komprimierungsalgorithmen muss in IKEv2 durch einen Notify Payload angezeigt werden. Obwohl IPCOMP ein eigenständiges

Protokoll ist, gibt es in IKEv2 kein mit einer Security Association vergleichbares Konstrukt. Es ist vielmehr so, dass eine "IPCOMP Verbindung" mit der CHILD-SA verschwindet, mit der sie aufgebaut wurde.

Initial Exchanges

Die Initial Exchanges (IKE-SA-INIT und IKE-AUTH) entsprechen der Phase 1 in ISAKMP. In IKEv1 wird die Abfolge der notwendigen Protokollelemente als Main Mode oder ihre abgekürzte Ausprägung als Aggressive Mode bezeichnet. Die Phase 1 dient den IKE-Protokollinstanzen zur gegenseitigen Authentifizierung und führt in der Regel zum Aufbau einer IKE SA.

Der Initiator einer IKEv2 SA sendet eine IKE-SA-INIT request Nachricht an den Partner, der für einen erfolgreichen Exchange mit einer gültigen IKE-SA-INIT response antwortet. Während dieses Dialogs werden Vereinbarungen über die anzuwendenden kryptografischen Verfahren getroffen, Initiator und Responder Nonces ausgetauscht und Diffie-Hellman Werte übertragen. Diese Informationen dienen der Erzeugung von Schlüsselmaterial und finden im folgenden IKE-AUTH exchange, bestehend aus IKE-AUTH request und IKE-AUTH reponse, Anwendung. Der IKE-AUTH Dialog authentifiziert die Nachrichten des IKE-SA-INIT exchange und dadurch implizit die Partnerinstanzen. Der Inhalt der IKE-AUTH Nachrichten ist teilweise verschlüsselt, so dass die Identität der Partner, für die eine CHILD-SA aufgebaut werden soll, bereits geheim bleibt. Eine CHILD-SA kann bereits im IKE-AUTH exchange aufgebaut werden.

IKE-SA und CHILD-SA in IKEv2 haben ihre Entsprechung in ISAKMP-SA bzw. IPSEC-SA in IKEv1.

CREATE-CHILD-SA Exchange

Der CREATE-CHILD-SA Dialog dient dem Aushandeln einer CHILD-SA. Er ist funktionell mit der Phase 2 in ISAKMP vergleichbar. Mit Hilfe dieses Dialogs wird eine CHILD-SA, deren Lebensdauer überschritten ist bzw. kurz davor ist, überschritten zu werden, durch eine neue CHILD-SA mit den gleichen Selektoren ersetzt. Die CHILD-SA wird der IKE-SA untergeordnet, unter deren (kryptografischen) Schutz sie ausgehandelt worden ist. Die eventuell existierende, zu ersetzende CHILD-SA muss mit Hilfe eines Informational Exchange gelöscht werden.

Eine CHILD-SA kann nur existieren, wenn eine IKE-SA existiert. Das Rekeying einer IKE-SA bedeutet demnach, dass zunächst eine neue IKE-SA aufgebaut werden muss, dieser alle CHILD-SAs der zu ersetzenden IKE-SA vererbt werden und danach die zu ersetzende IKE-SA gelöscht werden muss.

In IKEv1 wird der entsprechende Protokollablauf als Quick Mode bezeichnet.

Informational Exchange

Kontrollnachrichten, die zum Erhalt existierender SAs oder zur Anzeige einer Fehlersituation dienen, werden in Informational Exchanges ausgetauscht. Im Gegensatz zu IKEv1 umfassen auch sie einen Request/Response Dialog. Die Nachrichten sind verschlüsselt, d.h. sie werden unter dem Schutz der IKE-SA verschickt, für die sie einen Dienst erbringen. Betrifft der Dialog eine CHILD-SA, dann werden die Nachrichten durch die IKE-SA geschützt, der die CHILD-SA untergeordnet ist.

Das Löschen einer SA erfolgt mit Hilfe eines Informational Exchange, der als Inhalt einen DELETE Payload hat.

Mit Hilfe der Informational Exchanges können die beteiligten IKE-Instanzen erkennen, ob bestimmte Partner erreichbar sind (Dead Peer Detection). Dazu wird ein Request mit einem verschlüsselten "leerem" Payload verschickt. Erhält der Sender nach mehrmaligem Wiederholen keine gültige Response, betrachtet er den Partner als nicht erreichbar und löscht die IKE-SA und die zugehörigen CHILD-SAs.

4.8.2.5 Funktion von IKE (IKEv1 und IKEv2)

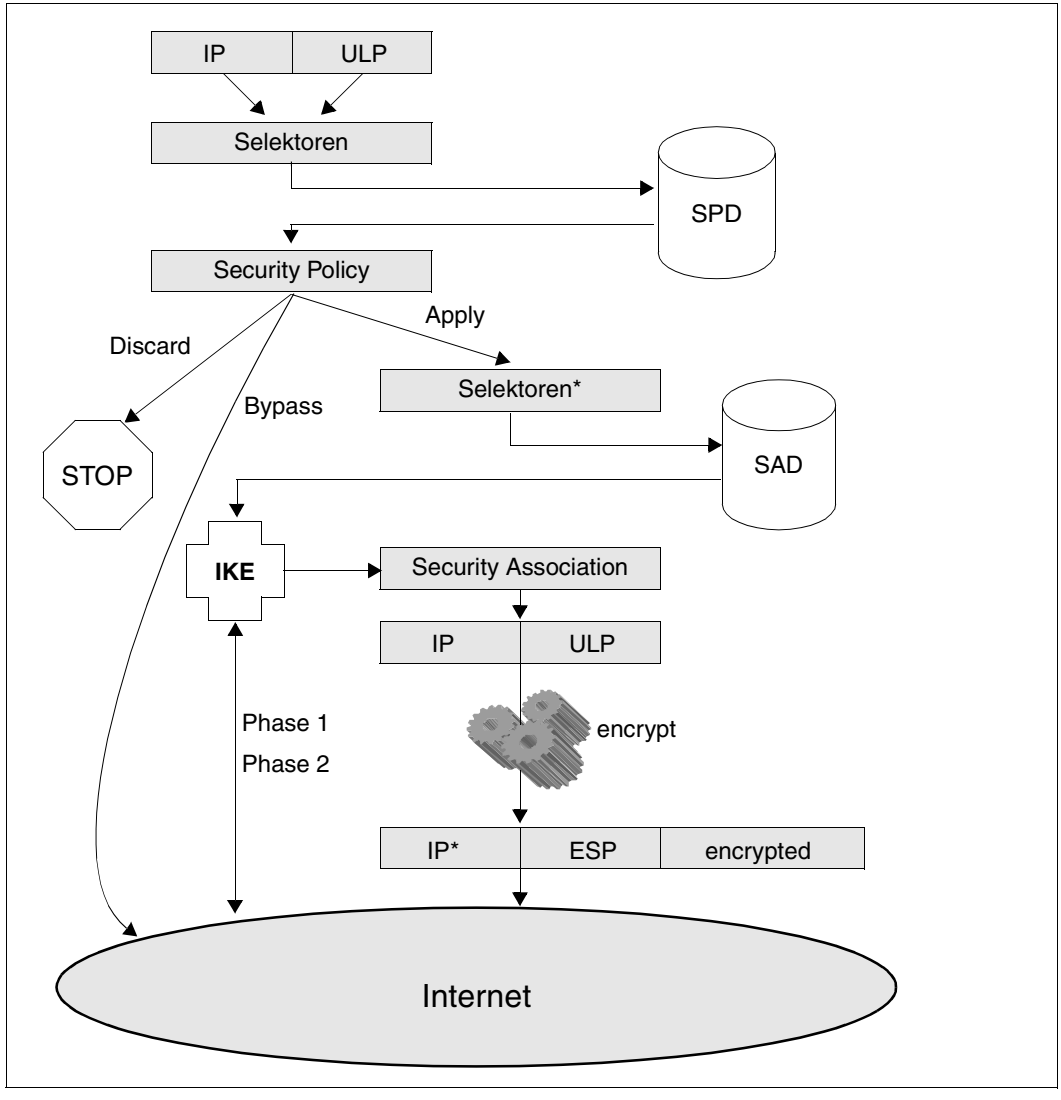


Bild 11: Zusammenspiel der Komponenten mit IKE

Existiert für ein ausgehendes IP-Segment eine Security Policy, die die Anwendung von IPSec-Diensten vorschreibt, aber keine passende Security Association, wird mit Hilfe des Internet Key Exchange Protocols eine entsprechende Sicherheitsverbindung aufgebaut. Das zu sendende IP-Segment wird für die Dauer der IKE Main/Quick Mode bzw. Initial-/ Child-SA Verhandlung in IPSec gepuffert. Nach einer erfolgreichen Installation der outbound SA wird das IP-Segment entsprechend der SA weiterverarbeitet.

5 Realisierung von IPSec im BS2000/OSD

Die IPSec-Funktionalität ist im BS2000/OSD durch Subsysteme realisiert. Folgendes Bild gibt einen Überblick über die Einbettung von IPSec in BS2000/OSD.

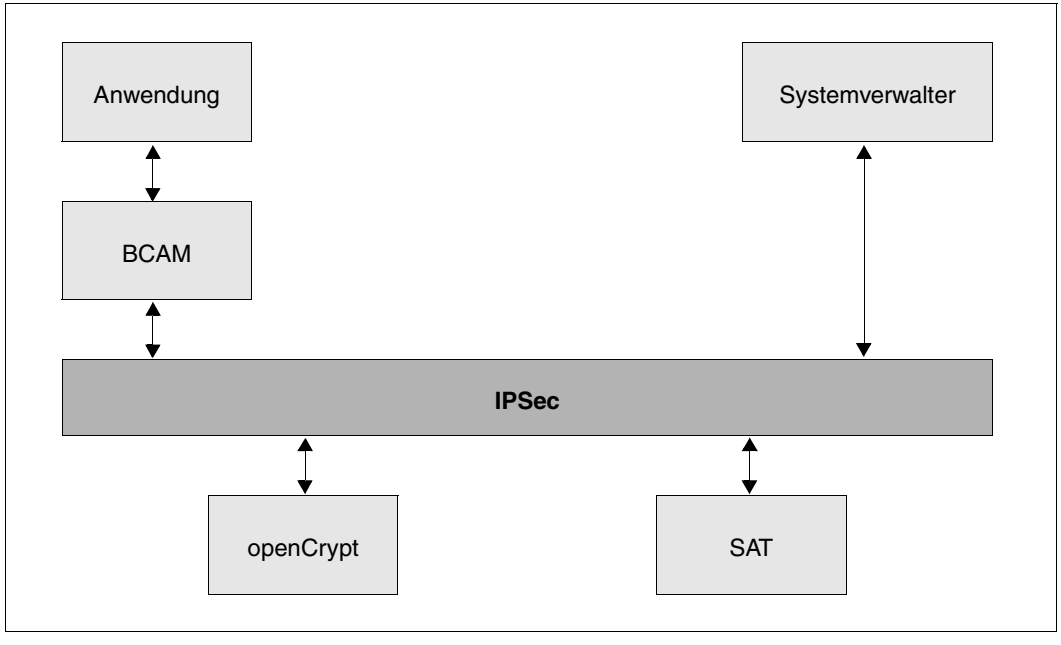


Bild 12: IPSec-Subsystem im BS2000/OSD-Umfeld

Security Audit Trail (SAT)

Die BS2000/OSD-Funktionseinheit SAT (Security Audit Trail) protokolliert alle sicherheitsrelevanten Ereignisse von IPSec. SAT ist ein Bestandteil von SECOS und dient als Protokollierungskomponente des BS2000/OSD für sicherheitsrelevante Ereignisse. Diese Ereignisse werden in SAT-Protokolldateien (SATLOG) abgelegt und können mit dem Dienstprogramm SATUT ausgewertet werden.

Ereignisse, die besonders sicherheitskritisch sind, können ohne Verzögerung mithilfe der SAT-Alarmfunktion überwacht werden. Die Operator-Konsole zeigt die Alarmmeldungen an, so dass entsprechende Maßnahmen eingeleitet werden können. Weitere Informationen zu SAT finden Sie im Benutzerhandbuch „SECOS (BS2000/OSD)“.

IPSec protokolliert die folgenden Ereignisse:

- Laden der IPSec Data Base war erfolgreich
- Laden der IPSec Data Base war fehlerhaft
- Verstoß gegen die Security Policy beim Datentransfer

Beim Verstoß gegen eine Security Policy werden folgende Daten an SAT übergeben:

- Eigene IPv4-Adresse
- Partner-IPv4-Adresse
- Eigene IPv6-Adresse
- Partner-IPv6-Adresse
- Fehlercode

Die Zuordnung der gemeldeten Daten zu den Ereignissen finden Sie im Benutzerhandbuch „SECOS (BS2000/OSD)“.

Die Verstöße gegen die Security Policy beim Datentransfer werden neben der Protokollierung an SAT auch als Meldung an der Operator-Konsole angezeigt.

openCrypt

Innerhalb des IPSec-Subsystems sind keine kryptografischen Algorithmen implementiert. Die notwendigen kryptografischen Leistungen werden durch das Produkt openCRYPT erbracht (siehe Handbuch „openCRYPT V1.2 (BS2000/OSD)“).

Bestandteile von IPSec

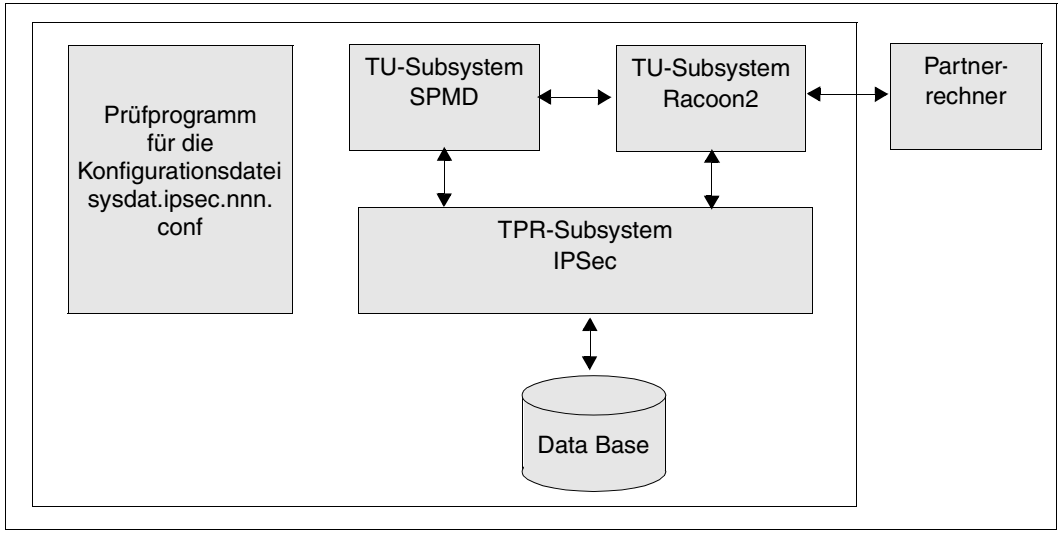


Bild 13: Wesentliche Bestandteile von IPSec

Data Base

Die IPSec Data Base wird im virtuellen Adressraum gehalten. Sie umfasst die Security Associations Data Base (SAD) und die Securitiy Policy Data Base (SPD). SAD und SPD sind in den Abschnitten „Sicherheitsrichtlinie (Security Policy)“ auf Seite 40 und „Sicherheitsverbindung (Security Association)“ auf Seite 42 ausführlich beschrieben.

Aus den Einträgen der IPSec-Konfigurationsdateien werden beim Starten des Subsystems IPSec zum einen die SPD und zum anderen der statische (manuell konfigurierte) Teil der SAD erzeugt. Im laufenden IPSec-Betrieb kann die IPSec Data Base jederzeit mithilfe von Kommandos, die ein Neueinlesen der Konfigurationsdateien bewirken, verändert werden.

Die Security Association Data Base wird durch den IKE-Daemon verwaltet. Die Security Policy Data Base wird durch das Programm SPMD verwaltet.

Subsystem IPSec

Über das Subsystem IPSec werden zum einen alle Zugriffe auf SAD und SPD vorgenommen. Zum anderen werden hier alle Sicherheitsprotokolle abgewickelt.

Subsystem Racoon2

Racoon2 ist ein Programm, das die Internet Key Exchange Protocols Version 1 und Version 2 realisiert, siehe Abschnitt „Automatisierte Verwaltung“ auf Seite 59. Racoon2 automatisiert den Aufbau von SAs und den Austausch von Schlüsseln zwischen dem eigenen und Partnerrechnern.

Subsystem SPMD

SPMD ist ein Programm, das die lokale Security Data Base verwaltet. Beim Start des Subsystems IPSec werden die in der Konfigurationsdatei SYSDAT.IPSEC.nnn.RAC2 definierten SP dem TPR-Subsystem IPSec bekannt gemacht. Über eine Kommunikationsschnittstelle bietet SPMD dem IKE-Daemon Informationen über auszuführende Aktionen.

6 IPSec - Konfiguration und Betrieb

6.1 IPSec-Installation

IPSec wird mithilfe des Produkts IMON (**I**nstallations **Mon**itor) installiert. Es empfiehlt sich, IPSec im Standard-Installationsmodus, d.h. auf die System-Standardkennung, zu installieren.

Nach der Installation mit IMON sind folgende Dateien installiert:

Subsystemkatalog:	$TSOS.SYSSSC.IPSEC.*nnn*
Subsystembibliothek:	$TSOS.SKMLNK.IPSEC.*nnn* $TSOS.SPMLNK.IPSEC.*nnn* $TSOS.SYSLNK.IPSEC.*nnn* $TSOS.SYSLNK.IPSEC.*nnn*.RAC2
Programmbibliothek:	$TSOS.SYSPRG.IPSEC.*nnn*
Kommandosyntaxdatei:	$TSOS.SYSSDF.IPSEC.*nnn*
Meldungsdatei:	$TSOS.SYSMES.IPSEC.*nnn*
SSINFO-Datei:	$TSOS.SYSSSI.IPSEC.*nnn*
IMON-Datei:	$TSOS.SYSSII.IPSEC.*nnn*
IPSec-Konfigurationsdatei:	$TSOS.SYSDAT.IPSEC.*nnn*.CONF
IPSec-Konfigurationsdatei:	$TSOS.SYSDAT.IPSEC.*nnn*.RAC2

nnn steht hier für die Version der BS2000/OSD-Installationseinheit IPSEC (z.B. 014 für die Version 1.4).

6.2 Inbetriebnahme - Kurzanleitung

Nach der erfolgreichen Installation kann das Subsystem IPSec gestartet werden. IPSec liest dabei die Default Konfigurationsdatei (SYSDAT.IPSEC.nnn.CONF) ein und startet den IKE-Daemon Racoon2 sowie den Security Policy Management Daemon SPMD. Die beiden Programme lesen ihre gemeinsame Konfigurationsdatei (SYSDAT.IPSEC.nnn.RAC2) ein. Auf Grund der implementierten Standardregel, die gesamte Kommunikation ungeschützt passieren zu lassen, bleibt IPSec unsichtbar.

Die Dienste von IPSec werden erst dann wirksam, wenn einige vorbereitende Maßnahmen ergriffen werden. Zunächst muss festgelegt werden, für welche Kommunikationsbeziehungen welche IPSec Dienste eingesetzt werden sollen. Mit anderen Worten, es müssen Policies definiert werden. Die Policy legt u.a. fest, wie IPSec zu Sicherheitsverbindungen kommt (automatisch oder manuell).

Im folgenden Beispiel wird demonstriert, was zu tun ist, um IPSec Dienste nutzen zu können. Ausführliche Informationen zu jedem Schritt finden Sie in den folgenden Kapiteln.

1. Festlegung der Policy
   Die $DIALOG-Verbindungen eines PCs sollen durch Verschlüsselung und Authentifizierung abgesichert werden.
   In die statische Konfigurationsdatei wird die Default-Policy explizit eingetragen. Mit dieser Policy wird der der Datentransfer zu anderen Partnern erlaubt.

2. Verfahrensauswahl
   Die Wahl des einzusetzenden Sicherheitsprotokolls und der Verschlüsselungsalgorithmen ist abhängig davon, was die beiden IPSec Partnersysteme unterstützen. Es ist also erforderlich, dass sich die Systemverwalter miteinander abstimmen.
   Für das Beispiel wird angenommen, dass beide Systeme automatischen Schlüsselaustausch, 3DES-CBC als Verschlüsselungsverfahren und HMAC-SHA-1 als Authentifizierungsverfahren unterstützen.
   Automatischer Schlüsselaustausch ist dem manuellem Austausch immer vorzuziehen, da ein deutlich geringerer Administrationsaufwand anfällt und die Übermittlung der Schlüsselinformation über einen abgesicherten Kommunikationsweg erfolgt.

3. Erstellen der IPSec-Konfigurationsdateien
   Die Beispielkonfigurationsdatei SYSDAT.IPSEC.*nnn*.CONF sieht wie folgt aus:

```
*
* definition of POLICY records
*
POLICY NAME=THEOTHERS -
,POLICY-RANGE=GLOBAL -
,OWN-ADDRESS=ANY -
,OWN-PORTNUMBER=ANY -
,DIRECTION=ANY -
,PROTOCOL=ANY -
,ICMP-TYPE=ANY -
,ICMP-CODE=ANY -
,MODE = BYPASS
```

Die Beispielkonfigurationsdatei SYSDAT.IPSEC.*nnn*.RAC2 sieht wie folgt aus:

```
#
# include of the default configuration file with the  default parameter
#
include "SYSDAT.IPSEC.nnn.RAC2.MANUAL-BSP-DEFAULT" ;

# PC-1
# Describes the KMP capabilities of PC-1
# Parameters of the default section are overriden.
#
remote PC-1 {
   acceptable_kmp { ikev1; };
   ikev1 {
      my_id ipaddr 172.25.92.72;
      peers_id ipaddr 172.25.83.42;
      peers_ipaddr 172.25.83.42 port 500;
      kmp_enc_alg { 3des_cbc;};
      kmp_hash_alg { sha1; md5;};
      kmp_dh_group { modp1024; };
      kmp_auth_method { psk; };
      pre_shared_key "PC-1.psk";
   };

};
#
# $DIALOG
# Selector and policy definition for the communication to $DIALOG
#
selector DIAL-in {
   direction inbound;
   src 172.25.83.42;
   dst 172.25.92.72 port 1110;
   upper_layer_protocol "tcp";
   policy_index PC-1;
```

```
};
selector DIAL-out {
   direction outbound;
   dst 172.25.83.42 ;
   src 172.25.92.72 port 1110;
   upper_layer_protocol "tcp";
   policy_index PC-1;
};
#
# Requested is IPsec protection with protocol ESP in end-to-end mode.
# Encryption and integrity protection is required.
#
policy PC-1 {
   action auto_ipsec;
   remote_index PC-1;
   ipsec_mode transport;
   ipsec_index { ipsec_esp; };
   ipsec_level require;
};

ipsec ipsec_esp {
   ipsec_sa_lifetime_time 28800 sec;
   sa_index esp_01;
};

sa esp_01 {
   sa_protocol esp;
#  esp_enc_alg { aes128_cbc; 3des_cbc; };
#  esp_auth_alg { hmac_sha1; hmac_md5; };
   esp_enc_alg {  3des_cbc; };
   esp_auth_alg { hmac_sha1; };
};
```

Die Konfigurationsdatei kann unter dem Namen SYSDAT.IPSEC.nnn.RAC2 gespeichert werden.

IPSec kann nur durch Selektoren konfiguriert werden, die aus den Protokollen der IP Suite abgeleitet werden können. Für $DIALOG ist es erforderlich, die TCP Portnummer 1110 anzugeben. Diese Information benötigt auch der PC Administrator. Die RFC1006 Portnummer 102 ermöglicht den Zugang zum System nach wie vor unverschlüsselt.

4. Erstellen der PSK-Datei für den Partnerrechner
   Der automatische Schlüsselaustausch wird durch das IKE-Protokoll realisiert. Die beiden IKE-Instanzen (Programme) auf den beteiligten Partnerrechnern müssen sich gegenseitig authentifizieren. Im BS2000 wird hierfür das Verfahren der preshared secret keys (PSK) verwendet. Für jeden Partnerrechner sollte eine eigene Datei angelegt werden, in die der PSK eingetragen wird. Für dieses Beispiel sieht der Eintrag in der Datei PC-1.psk so aus:

   ```
   =)(shbwldqU&$RQA
   ```

   Die Information, dass PSK als Authentifikationsverfahren verwendet werden muss und welcher Schlüssel zu verwenden ist, muss dem PC Administrator zur Verfügung gestellt werden.

5. Partnerrechner konfigurieren.
   Analoge Konfigurationsschritte müssen natürlich auch auf dem Partnerrechner vorgenommen werden. Dabei ist darauf zu achten, dass identische Konfigurationen erstellt werden.

6. IPSec starten
   Das IPSec-Subsystem wird mit folgendem Kommando gestartet:

   ```
   /START-SUBSYSTEM IPSEC
   ```

   Während des Startens wird die Datei SYSDAT.IPSEC.nnn.CONF eingelesen. Diese Datei enthält die statischen Policy und SA Definitionen. Das Programm SPMD wird gestartet. SPMD liest die Konfigurationsdatei SYSDAT.IPSEC.nnn.RAC2 und ergänzt die Security Policy Database um die in der Datei beschriebenen Security Policies.
   SPMD startet danach das Programm Racoon2, das ebenfalls die Datei SYSDAT.IPSEC.nnn.RAC2 einliest. SPMD und Racoon2 haben also die gleiche SPD.

7. Verbindung aufbauen
   Eine verschlüsselte Dialogverbindung wird dann aufgebaut, wenn in der Emulation die Zielportnummer 1110 verwendet wird.

8. Überprüfen der Verbindung
   Zum Überprüfen, ob für die $DIALOG-Verbindung zum PC tatsächlich IPSec verwendet wird, eignet sich das Kommando SHOW-CONNECTION mit geeigneten Selektionsparametern IPSEC und der Angabe des Prozessornamens.

   Beispielsweise muss nach der Eingabe des Kommandos

   ```
   /SHOW-CONNECTION SELECT=*BY-ATTRIBUTES(IPSEC=*YES)
   ```

   eine Meldung `BCA08A2` erscheinen, die eine Verbindung zum Partnerrechner beschreibt und die Information IPSEC=YES enthält.

6.3 IPSec-Konfiguration

Der Funktionsumfang von IPSec, die Überwachung des Datentransfers und gegebenenfalls die Anwendung von Sicherheitsdiensten auf einzelne IP-Segmente, wird durch die Security Policy Database (SPD) und die Security Association Database (SAD) kontrolliert und gesteuert. Der Aufbau und die Verwaltung dieser Datanbasis erfolgt im BS2000/OSD mit Hilfe mehrerer Konfigurationsdateien. Während der Installation von IPSec wird die leere Defaultdatei SYSDAT.IPSEC.*nnn*.CONF erzeugt, die beim Starten des Subsystems eingelesen wird. Die Manipulation der Datenbasis im laufenden IPSec Betrieb erfolgt ebenfalls durch Konfigurationsdateien.

Details finden Sie Abschnitt „IPSec-Konfiguration ändern“ auf Seite 111.

ACHTUNG!

Die IPSec-Konfigurationsdateien enthalten sicherheitsrelevante Informationen und sind deshalb vor unberechtigtem Zugriff zu schützen. Die Dateizugriffsrechte sollten auf USER-ACCESS = *OWNER-ONLY gesetzt werden.
SPMD und Racoon2 arbeiten mit derselben Konfigurationsdatei. Die Manipulation der Datenbasis im laufenden IPSec Betrieb muss deshalb immer für beide Programme erfolgen.

6.3.1 Verarbeitung der Dateien für die statische Konfiguration

IPSec-Konfigurationsdateien des BS2000/OSD enthalten in der Regel Anweisungen und Objektdefinitionen. Anweisungen steuern die Reihenfolge und die Art der Objektverarbeitung. Objektdefinitionen erzeugen Identifikatoren, stellen Beziehungen untereinander her und beschreiben somit, was innerhalb der Datenbasis manipuliert wird. Objekte werden durch Namen identifiziert.

Die Verarbeitung einer Menge von Konfigurationsdateien unterliegt einigen **Regeln**.

- Anweisungen werden sofort ausgeführt.
- Der Geltungsbereich einer Anweisung ist auf eine einzelne Konfigurationsdatei beschränkt.
- Die Objektnamen müssen eindeutig sein, da Objekte außerhalb der definierenden Konfigurationsdateien sichtbar sein müssen.
- Innerhalb der Objekte sind nur Rückwärtsbezüge zulässig. D.h. referenziert ein Objekt ein anderes Objekt, muss dieses bereits bekannt sein. Das angesprochene Objekt kann in einer anderen, aber früher eingelesenen Konfigurationsdatei definiert sein.
- Die Reihenfolge der Objektdefinitionen ist ansonsten beliebig. Sie beeinflusst die Ordnung der Datenbasis nicht.

Folgende **Sprachmittel** stehen zum Aufbau und zur Manipulation der IPSec Datenbasis zur Verfügung.

- Die Anweisungen zur Verarbeitung des Inhalts einer Konfigurationsdatei sind:
  - FLUSH
  - INCLUDE
  - ADD
  - DELETE
- Für die Definition von Verschlüsselungsalgorithmen, Signiermethoden, Security Associations und Security Policies stehen fünf Objekte zur Verfügung, die synonym als Konfigurationssätze bezeichnet werden:
  - KEY-Satz
  - SIGNATURE-Satz
  - SECURITY-ASSOCIATION-Satz
  - POLICY-Satz
  - PARTNER-SAS-Satz

Das POLICY Objekt ist konzeptionell das wichtigste Sprachmittel zum Aufbau der IPSec Datenbasis im BS2000/OSD. Die POLICY bestimmt, was mit einzelnen IP-Segmenten zu tun ist. Dazu dienen Verweise im Definitionssatz auf die entsprechenden SECURITY-ASSOCIATION-Sätze. Durch die POLICY werden die **Selektoren** für die IP-Segmentfilterung festgelegt. Aus den Selektoren der Policy werden bei der administrativen Schlüsselverwaltung die Selektoren für die referenzierten Security Associations abgeleitet. Die Security Policy Database bezieht die vorgeschriebene Ordnung aus den definierten Selektoren der Policies.

Für die Verarbeitung der definierten Security Policies einer Menge von Konfigurationsdateien sind im BS2000/OSD **weitere Regeln** zu beachten:

- Es werden keine Security Labels und Namensselektoren wie DNS- oder X.500 Namen unterstützt.
- Wildcard- und Bereichsangaben werden unterstützt.
- Bereichsüberlappungen werden unterstützt.
- Policies, mit genauerer Beschreibung werden so in die SPD einsortiert, dass sie bei Suchvorgängen vor weniger exakt definierten Policies gefunden werden. Als Maß für die Genauigkeit wird die Anzahl der explizit oder implizit vorhandenen Bereichsangaben herangezogen.
- Policies mit unterschiedlichen Namen, aber mit identischen Selektordefinitionen, werden erkannt und abgewiesen.

Security Associations, die durch administrative Schlüsselverwaltung gekennzeichnet sind, werden bereits beim Start des Subsystems IPSec erzeugt und in die SAD eingetragen. Die Ordnung der SAD ist durch Selektoren bestimmt. Als ergänzende Unterscheidungskriterien dienen der Typ des Sicherheitsprotokolls (AH bzw. ESP) und der Security Parameter Index (SPI). Dies ist aus folgenden Gründen notwendig:

- Ein POLICY Objekt kann mehr als ein SECURITY-ASSOCIATION Objekt referenzieren, um ein SA Bundle zu definieren. Würden nur die Selektordefinitionen berücksichtigt, ließen sich die Security Associations nicht hinreichend unterscheiden.
- Automatisiertes Schlüsselmanagement legt selbständig Security Associations an, bevor die definierte Lebensdauer einer genutzten SA abgelaufen ist. Dadurch werden unnötige Verzögerungen beim Datentransfer vermieden. Diese SAs unterscheiden sich in ihrem SPI.
- Zur Suche einer SA für ein ankommendes und mit IPSec geschütztes IP-Segment sind lediglich die IP-Adressen, das Sicherheitprotokoll und der SPI auswertbar. Die Selektoren, die die End-to-End Beziehung der Kommunikationspartner beschreiben, sind zum Empfangszeitpunkt nicht erkennbar.

BS2000/OSD unterstützt die von der Sicherheitsarchitektur geforderten Profile, d.h. End-to-End Sicherheitsverbindungen im Transport- und Tunnelmodus, sowie Tunnelmodus zu Security Gateways. IPSec führt keine Überprüfung der Adressangaben durch, kann also für eine Tunnelmode SA nicht erkennen, ob sie ein Endsystem oder Security Gateway als Endpunkt hat.

i

- Die INCLUDE Anweisung ermöglicht es, die Beschreibung der IPSec Datenbasis in überschaubaren Bausteinen zu halten. So können die für administrative Schlüsselverwaltung notwendigen Geheimnisse in eigenen Dateien abgelegt werden. Die Sicherheitsrichtlinien für einzelne Adressen oder Adressbereiche können in eigenen Dateien gehalten werden.
- Die DISCARD Policy eignet sich für den Aufbau einer Endsystem Firewall, die bis auf die Genauigkeit einzelner Portnummern und IP-Adressen definiert werden kann und die den Zugang zum System sperrt bzw. die Übertragung aus dem System verhindert.
- Wegen der fehlenden Unterstützung von Namensselektoren lassen sich keine Sicherheitsrichtlinien für Anwendungen definieren, die über TCP-Portnummer 102 adressiert werden. BS2000/OSD bietet durch das BCMAP-Kommando eine Umgehungsmöglichkeit (vgl. vorangehendes Beispiel).

6.3.2 Steueranweisungen für die IPSec-Konfigurationsdatei

Die IPSec-Konfigurationsdatei ist eine SAM-Datei mit folgendem Satzformat:

- Die maximale Satzlänge beträgt 255 Zeichen.
- Das Zeichen * leitet Kommentare ein.
- Die letzte Position einer Zeile innerhalb eines Satzes ist für das Fortsetzungszeichen "-" reserviert.

FLUSH: Löschen einer vorhandenen Konfiguration

Durch die FLUSH-Anweisung wird die vorhandene Konfiguration gelöscht. Die FLUSH-Anweisung und eine Konfigurationsdatei ohne Objektdefinition und ohne Anweisung liefern dasselbe Ergebnis.

Da die FLUSH-Anweisung auch dann ausgeführt wird, wenn sie nicht am Anfang einer IPSec-Konfigurationsdatei steht, sollte sie wohl überlegt angewendet werden.

Die FLUSH-Anweisung muss alleine in einer Zeile stehen!

FLUSH

INCLUDE: Einlesen einer weiteren Konfigurationsdatei

Mt Hilfe der INCLUDE-Anweisung wird eine weitere IPSec-Konfigurationsdatei eingelesen. Die Bearbeitung der definierten Datensätze erfolgt synchron, d.h. Datensätze, die von einzufügenden Datensätzen referenziert werden, müssen bereits eingelesen worden sein. Die INCLUDE-Anweisung kann rekursiv angewendet werden. Sie erlaubt die Verwaltung der Konfigurationsbeschreibung in Dateien mit überschaubarer Größe. Insbesondere ist es durch die INCLUDE-Anweisung möglich, Schlüsselwerte und Policies getrennt zu verwalten.

INCLUDE <filename 1..54>

Beschreibung der Operanden

<filename 1..54>
Name der einzufügenden Konfigurationsdatei.

ADD: Ändern des Verarbeitungsmodus in Hinzufügen

Die ADD-Anweisung ändert den Verarbeitungsmodus. Die auf die ADD-Anweisung folgenden Objektdefinitionen erzeugen Objekte in der IPSec-Datenbasis. Die ADD-Anweisung ist auf die Konfigurationsdatei beschränkt, in der sie steht. Sie gilt bis zur nächsten DELETE-Anweisung in der gleichen Konfigurationsdatei bzw. bis an das Ende dieser Datei.

Der Verarbeitungsmodus ADD ist die Defaulteinstellung für jede Konfigurationsdatei. Konfigurationsdateien können nacheinander oder mit Hilfe der INCLUDE Anweisung geschachtelt eingelesen werden.

Die ADD-Anweisung muss alleine in einer Zeile stehen!

ADD

DELETE: Ändern des Verarbeitungsmodus in Löschen

Die DELETE-Anweisung ändert den Verarbeitungsmodus. Die auf die DELETE-Anweisung folgenden Objektdefinitionen löschen Objekte aus der IPSec-Datenbasis. Die DELETE-Anweisung ist auf die Konfigurationsdatei beschränkt, in der sie steht. Sie gilt bis zur nächsten ADD-Anweisung in der gleichen Konfigurationsdatei bzw. bis an das Ende dieser Datei.

Die Objektdefinitionen der zu löschenden Objekte benötigen lediglich das Schlüsselwort NAME. (Siehe Beschreibung der Konfigurationssätze KEY, SIGNATURE, SECURITY-ASSOCIATION und POLICY.)

Objekte mit einer gültigen NAME Definition werden auch dann gelöscht, wenn andere Parameterdefinitionen des Objekts syntaktisch fehlerhaft sind.

Die DELETE-Anweisung muss alleine in einer Zeile stehen!

DELETE

KEY: Verschlüsselungsalgorithmus definieren

Der KEY-Satz definiert eine neue Verschlüsselungsmethode innerhalb der IPSec Data Base.

KEY
KEY NAME = <alphanum-name 1 .. 32> **,KEY-ALGORITHM = DES-CBC / 3DES-CBC / AES-CBC** **,KEY-VALUE =** <x-string 16 .. 16> / <x-string 32 .. 32> / <x-string 64 .. 64>

KEY
KEY NAME = <alphanum-name 1 .. 32> **,KEY-ALGORITHM = NULL**

Beschreibung der Operanden

NAME= <alphanum-name 1 .. 32>
Name des Schlüssels. Der Schlüsselname kann aus Buchstaben und Ziffern bestehen. Groß-/Kleinschreibung der Buchstaben wird berücksichtigt. Das erste Zeichen des Namens muss ein Buchstabe sein.

KEY-ALGORITHM =
spezifiziert den zu verwendenden Verschlüsselungsalgorithmus.

KEY-ALGORITHM = DES-CBC
Es wird der DES-Algorithmus im CBC-Modus verwendet.

KEY-ALGORITHM = 3DES-CBC
Es wird der TripleDES-Algorithmus im CBC-Modus verwendet.

KEY-ALGORITHM = AES-CBC
Es wird der AES-Algorithmus im CBC-Modus verwendet.

KEY-ALGORITHM = NULL
Es wird kein Verschlüsselungsalgorithmus verwendet.

KEY-VALUE =
spezifiziert den zu verwendenden Schlüssel.

KEY-VALUE = <x-string 16 .. 16> / <x-string 32 .. 32> / <x-string 48 .. 48> / <x-string 64 .. 64>
In Hochkommata (‘) eingeschlossener String aus Sedezimal-Zeichen mit vorangestelltem x oder X (z.B. X’01’, x’12AB’).

ACHTUNG!

Die zulässige Schlüssellänge ist abhängig vom verwendeten Verschlüsselungsalgorithmus:

DES-CBC:	16 Sedezimalzeichen
3DES-CBC:	32 / 48 Sedezimalzeichen
AES-CBC:	32 / 48 / 64 Sedezimalzeichen

Falls zur Darstellung des Schlüssels eine Zeile nicht ausreicht, können Sie eine oder mehrere Fortsetzungszeilen verwenden. Hierfür schreiben Sie in der fortzusetzenden Zeile bis zum Zeilenende und beginnen die Fortsetzungszeile in Spalte 1 mit dem nächsten Zeichen des Schlüssels.

SIGNATURE: Signiermethode definieren

Der SIGNATURE-Satz definiert eine neue Signiermethode innerhalb der IPSec Data Base .

SIGNATURE
SIGNATURE NAME = <alphanum-name 1 .. 32> **,SIGNATURE-ALGORITHM = HMAC-MD5** / **HMAC-SHA-1** **,SIGNATURE-VALUE =** <x-string 32 .. 32> / <x-string 40 .. 40>

SIGNATURE
SIGNATURE NAME = <alphanum-name 1 .. 32> **,SIGNATURE-ALGORITHM = NULL**

Beschreibung der Operanden

NAME = <alphanum-name 1 .. 32>
Name der Signiermethode. Der Name kann aus Buchstaben und Ziffern bestehen. Groß-/Kleinschreibung der Buchstaben wird berücksichtigt. Das erste Zeichen des Namens muss ein Buchstabe sein.

SIGNATURE-ALGORITHM =
spezifiziert den zu verwendenden Signatur-Algorithmus.

SIGNATURE-ALGORITHM = HMAC-MD5
Es wird der MD5-Algorithmus verwendet.

SIGNATURE-ALGORITHM = HMAC-SHA-1
Es wird der SHA-1-Algorithmus verwendet.

SIGNATURE-ALGORITHM = NULL
Es wird kein Signatur-Algorithmus verwendet.

SIGNATURE-VALUE =
spezifiziert den Schlüssel, der zur Bildung der Signatur verwendet werden soll.

SIGNATURE-VALUE = <x-string 32 .. 32> / <x-string 40 .. 40>
In Hochkommata (‘) eingeschlossener String aus Sedezimal-Zeichen mit vorangestelltem x oder X (z.B. X’01’, x’12AB’).

ACHTUNG!

Die zulässige Schlüssellänge ist abhängig vom verwendeten Signaturverfahren:

HMAC-MD5: 32 Sedezimalzeichen

HMAC-SHA-1: 40 Sedezimalzeichen

Falls zur Darstellung des Schlüssels eine Zeile nicht ausreicht, können Sie eine oder mehrere Fortsetzungszeilen verwenden. Hierfür schreiben Sie in der fortzusetzenden Zeile bis zum Zeilenende und beginnen die Fortsetzungszeile in Spalte 1 mit dem nächsten Zeichen des Schlüssels.

SECURITY-ASSOCIATION: Security Association definieren

Durch den SECURITY-ASSOCIATION-Satz werden Eigenschaften einer Security-Association definiert.

SECURITY-ASSOCIATION

SECURITY-ASSOCIATION NAME = < alphanum-name 1 .. 32>

,TYPE = AUTHENTICATION (...) / **ENCRYPTION (...)**

AUTHENTICATION (...)
SIGNATURE = <alphanum-name 1 .. 32>

ENCRYPTION (...)
SIGNATURE= <alphanum-name 1 .. 32> / ***NONE**
, KEY=<alphnum-name 1 .. 32> / ***NONE**

,**MODE = MANUAL (...) / IKE [(...)]**

MANUAL (...)
INDEX = <integer 256 .. 4294967295>

[**,ECN-TUNNEL = ALLOWED** / **FORBIDDEN**]

[**,ECN-TUNNEL-NEGOTIATION = YES** / **NO**]

Beschreibung der Operanden

NAME = <alphanum-name 1 .. 32>
Name des Security-Association-Satzes. Der Name kann aus Buchstaben und Ziffern bestehen. Groß-/Kleinschreibung der Buchstaben wird berücksichtigt. Das erste Zeichen des Namens muss ein Buchstabe sein. Über den Namen wird im SECURITY-POLICY-Satz auf den zu verwendenden SECURITY-ASSOCIATION-Satz verwiesen.

TYPE = AUTHENTICATION (...) / ENCRYPTION (...)
spezifiziert den Typ der Security Association.

AUTHENTICATION (SIGNATURE = ...)
Es wird eine Authentifizierung mit einem Authentication-Header (AH) durchgeführt.

SIGNATURE = <alphanum-name 1 .. 32>
Name der verwendeten Signiermethode. Diese Signiermethode muss bereits definiert sein.

ENCRYPTION (SIGNATURE = ... , KEY= ...)
Es wird eine Verschlüsselung und/oder eine Authentifizierung mit einem Encapsulated Security Payload Header (ESP) durchgeführt.

SIGNATURE = <alphanum-name 1 .. 32> / *NONE
Name des verwendeten SIGNATURE-Satzes. Dieser SIGNATURE-Satz muss bereits definiert sein.
*NONE bedeutet, dass keine Authentifizierung durchgeführt wird.

KEY = <alphanum-name 1 .. 32> / *NONE
Name des verwendeten KEY-Satzes. Dieser KEY-Satz muss bereits definiert sein.
*NONE bedeutet, dass nicht verschlüsselt wird.

MODE = MANUAL (INDEX = ...)
Es wird eine statische SECURITY-ASSOCIATION beim Einlesen der Konfigurationsdatei erzeugt.

INDEX = <integer 256 .. 4294967295>
spezifiziert den Security Parameter Index (SPI), der bei eingehenden (incoming) AH- und ESP-Headern die Security Association identifiziert, die mit dem vorliegenden SECURITY-ASSOCIATION-Satz definiert wird. Der Bereich <integer 0 .. 255> ist für interne Zwecke reserviert (RFC 2406).

Die beiden folgenden Operanden müssen genau dann definiert werden, wenn die Security Association für einen IP-Tunnel verwendet wird.

ECN-TUNNEL = ...
steuert die Verwendung von ECN-Tunneln gemäß RFC 3168. Dieser Parameter wird nur bei Verwendung der Security Association für einen IPSec-Tunnel benötigt.

ECN-TUNNEL = ALLOWED
Die Verwendung von ECN-Tunneln gemäß RFC 3168 ist erlaubt.

ECN-TUNNEL = FORBIDDEN
Die Verwendung von ECN-Tunneln gemäß RFC 3168 ist verboten.

ECN-TUNNEL-NEGOTIATION = ...
legt fest, ob die Verwendung von ECN-Tunneln mit dem IPSec-Partner ausgehandelt werden kann. Dieser Parameter wird nur bei Verwendung der Security Association für einen IPSec-Tunnel benötigt.

ECN-TUNNEL-NEGOTIATION = YES
Die Verwendung von ECN-Tunneln gemäß RFC 3168 kann mit dem IPSec-Partner ausgehandelt werden.

ECN-TUNNEL-NEGOTIATION = NO
Die Verwendung von ECN-Tunneln gemäß RFC 3168 kann nicht mit dem IPSec-Partner ausgehandelt werden.

POLICY: Security Policy definieren

Der POLICY-Satz definiert eine neue Security Policy.

POLICY

POLICY NAME = <alphanum-name 1 .. 32>

,POLICY-RANGE = GLOBAL / **PARTNERSYSTEM (...)** / **PARTNER (...)**

PARTNERSYSTEM (IP-ADDRESS = ... / **IPV6-ADDRESS = ...)**

IP-ADDRESS = <ipv4addr> / <ipv4addr/prefix-len> / <ipv4addr-low> - <ipv4addr-high> / **ANY**

IPV6-ADDRESS = <ipv6addr> / <ipv6addr/prefix-len> / <ipv6addr-low> - <ipv6addr-high> / **ANY**

PARTNER (IP-ADDRESS = ... , PORTNUMBER = ... / **IPV6-ADDRESS = ... , PORTNUMBER = ...)**

IP-ADDRESS = <ipv4addr> / <ipv4addr/prefix-len> / <ipv4addr-low> - <ipv4addr-high> / **ANY**

IPV6-ADDRESS = <ipv6addr> / <ipv6addr/prefix-len> / <ipv6addr-low> - <ipv6addr-high> / **ANY**

,PORTNUMBER = ANY / <port#> / <port#-low> - <port#-high>

,FIRST-SECURITY-ASSOCIATION = <alphanum-name 1 ..32> / ***NONE**

,SECOND-SECURITY-ASSOCIATION = <alphanum-name 1 .. 32> / ***NONE**

[**,COMPRESSION = DEFLATE** / ***NONE**]

,MODE = BYPASS / DISCARD / TRANSPORT, ... / **TUNNEL (...), ...**

TUNNEL (...), ...

INNER-TUNNEL = (END-OF-TUNNEL (...) [, **START-OF-TUNNEL = (...)]**

,SECURITY-ASSOCIATION = ... [,COMPRESSION = ...])

END-OF-TUNNEL (IP-ADDRESS = <ipv4addr > / **IPV6-ADDRESS =** <ipv6addr>**)**

[**,START-OF-TUNNEL (IP-ADDRESS =** <ipv4addr > / **IPV6-ADDRESS =** <ipv6addr>**)**]

,SECURITY-ASSOCIATION = <alphanum-name 1 ..32> / ***NONE**

[**,COMPRESSION = DEFLATE** / ***NONE]**

OUTER-TUNNEL = *NONE / (END-OF-TUNNEL (...) [, **START-OF-TUNNEL = (...)]**

,SECURITY-ASSOCIATION = ...[,COMPRESSION = ...])

END-OF-TUNNEL (IP-ADDRESS = <ipv4addr > / **IPV6-ADDRESS =** <ipv6addr>**)**

[**,START-OF-TUNNEL (IP-ADDRESS =** <ipv4addr > / **IPV6-ADDRESS =** <ipv6addr>**)**]

,SECURITY-ASSOCIATION = <alphanum-name 1 ..32> / ***NONE**

[**,COMPRESSION = DEFLATE** / ***NONE**]

,OWN-ADDRESS = ANY / **IP-ADDRESS =** <ipv4addr> / **IPV6-ADDRESS =** <ipv6addr>

,OWN-PORTNUMBER = ANY / <port#> / <port#> - <port#>

,DIRECTION = IN / **OUT** / **ANY**

,PROTOCOL = TCP / **UDP / ICMP, ...** / **ANY, ...**

,ICMP-CODE = ANY / <code> / <code-low> - <code-high>

,ICMP-TYPE = ANY / <type> / <type-low> - <type-high>

Beschreibung der Operanden

NAME = <alphanum-name 1 .. 32>
Name der Security Policy. Der Name kann aus Buchstaben und Ziffern bestehen. Groß-/Kleinschreibung der Buchstaben wird berücksichtigt. Das erste Zeichen des Namens muss ein Buchstabe sein.

POLICY-RANGE = ...
spezifiziert den Gültigkeitsbereich der Security Policy.

POLICY-RANGE = GLOBAL
Die Security Policy kann für den Datentransfer zu allen Partneranwendungen in allen Partnersystemen angewendet werden.

POLICY-RANGE = PARTNERSYSTEM (IP-ADDRESS = ... / IPV6-ADDRESS = ...)
Die Security Policy kann für den Datentransfer zu den angegebenen Partnersystemen angewendet werden.

IP-ADDRESS = <ipv4addr> / <ipv4addr/prefix-len> /
<ipv4addr-low> - <ipv4addr-high> / ANY
IPv4-Adresse, IPv4-Adresse mit Präfix-Angabe oder Bereich von IPv4-Adressen. Diese spezifiziert / spezifizieren das Partnersystem bzw. die Partnersysteme, auf die die Policy angewendet werden kann. Die IPv4-Adresse(n) muss / müssen in der üblichen „decimal dotted“-Schreibweise angegeben werden.
ANY bedeutet eine beliebige IPv4-Adresse.

IPV6-ADDRESS = <ipv6addr> / <ipv6addr/prefix-len> /
<ipv6addr-low> - <ipv6addr-high> / ANY
IPv6-Adresse, IPv6-Adresse mit Präfix-Angabe oder Bereich von IPv6-Adressen. Diese spezifiziert / spezifizieren das Partnersystem bzw. die Partnersysteme, auf die die Policy angewendet werden kann. Die IPv6-Adresse(n) muss / müssen in der üblichen Sedezimaldarstellung mit Doppelpunkt (:) angegeben werden.
ANY bedeutet eine beliebige IPv6-Adresse.

POLICY-RANGE = PARTNER (IP-ADDRESS = ... , PORTNUMBER = ... /
IPV6-ADDRESS = ... , PORTNUMBER = ...)
Die Security Policy kann für den Datentransfer zu den angegebenen Partneranwendungen in den angegebenen Partnersystemen angewendet werden.

IP-ADDRESS = <ipv4addr> / <ipv4addr/prefix-len> /
<ipv4addr-low> - <ipv4addr-high> / ANY
IPv4-Adresse, IPv4-Adresse mit Präfix-Angabe oder Bereich von IPv4-Adressen. Diese spezifiziert / spezifizieren das Partnersystem bzw. die Partnersysteme, auf die die Policy angewendet werden kann. Die IPv4-Adresse(n) muss / müssen in der üblichen „decimal dotted“-Schreibweise angegeben werden.
ANY bedeutet eine beliebige IPv4-Adresse.

IPV6-ADDRESS = <ipv6addr>/<ipv6addr/prefix-len> /
<ipv6addr-low> - <ipv6addr-high> / ANY
IPv6-Adresse, IPv6-Adresse mit Präfix-Angabe oder Bereich von IPv6-Adressen. Diese spezifiziert / spezifizieren das Partnersystem bzw. die Partnersysteme, auf die die Policy angewendet werden kann. Die IPv6-Adresse(n) muss / müssen in der üblichen Sedezimaldarstellung mit Doppelpunkt (:) angegeben werden.
ANY bedeutet eine beliebige IPv6-Adresse.

PORTNUMBER = ANY / <port#> / <port#-low> - <port#-high>
Portnummer oder Bereich von Portnummern. Diese spezifiziert / spezifizieren die Partneranwendung(en), auf die die Policy angewendet werden kann. Gültige Portnummern liegen im Bereich von 0 bis 65535 .
Durch das Schlüsselwort ANY wird der Wertebereich von 0 bis 65535 festgelegt.

FIRST-SECURITY-ASSOCIATION = <alphanum-name 1 .. 32> / *NONE
Name der Security-Association, die auf den Inhalt des IP-Paketes anzuwenden ist. Der entsprechende SECURITY-ASSOCIATION-Satz muss bereits definiert sein.

SECOND-SECURITY-ASSOCIATION = <alphanum-name 1 .. 32> / *NONE
Name der Security-Association, die auf den Inhalt des eventuell bereits durch FIRST-SECURITY-ASSOCIATION geschützten IP-Paketes anzuwenden ist.
Der Security Policy darf nur dann eine zweite Security Association zugeordnet werden, wenn dieser Security Policy mit FIRST-SECURITY-ASSOCIATION = ... bereits eine erste Security Association zugeordnet wurde.
Bei Verwendung von 2 SA's muss die 1.SA das Sicherheitsprotokoll ESP, die 2. SA das Sicherheitsprotokoll AH verwenden (siehe Abschnitt „Sicherheitskonzepte auf der Basis von Security Associations“ auf Seite 158)
Das Schlüsselwort *NONE legt fest, dass keine zweite Security Association verwendet wird.

COMPRESSION = DEFLATE / *NONE
Über den optionalen Operanden COMPRESSION kann die Komprimierung der Daten gefordert werden. Die Komprimierung erfolgt vor der Anwendung eines IPsec-Protokolls.
Der Parameterwert DEFLATE gibt den Komprimierungsalgorithmus an.
Der Defaultwert ist *NONE.

MODE = BYPASS / DISCARD / TRANSPORT, ... / TUNNEL(...), ...
gibt an, wie mit den Paketen der spezifizierten Kommunikationsbeziehung zu verfahren ist.

MODE = BYPASS
Auf Pakete, die von der Policy betroffen sind, werden keine IPSec Funktionen angewendet. Sie werden unverschlüsselt und ohne weitere Überprüfung übertragen. Es muss keine Security-Association angegeben werden.

MODE = DISCARD
Pakete, die von der Policy betroffen sind, werden verworfen. Dadurch ist es möglich, Kommunikationsbeziehungen zu verhindern. Es muss keine Security-Association angegeben werden.

MODE = TRANSPORT, ...
Die Security Policy sichert den Datentransfer im Transportmodus.

MODE = TUNNEL (...), ...
Die Security Policy sichert den Datentransfer im Tunnelmodus.

INNER-TUNNEL = (END-OF-TUNNEL (...) [,START-OF-TUNNEL(...)] ,SECURITY-ASSOCIATION = ...)
spezifiziert den IPSec-Tunnel der Security Policy. Er wird definiert durch die Operanden END-OF-TUNNEL, START-OF-TUNNEL (optional) sowie SECURITY-ASSOCIATION. Falls der START-OF-TUNNEL-Parameter nicht angegeben ist, ist der lokale Endpunkt des Tunnels die bei OWN-ADDRESS angegebene Adresse.

END-OF-TUNNEL (IP-ADDRESS = <ipv4addr > / IPV6-ADDRESS = <ipv6addr>)
spezifiziert die remote Adresse des inneren Tunnel-Endpunkts. Der lokale Endpunkt des Tunnels wird via START-OF-TUNNEL-Parameter (siehe unten) bestimmt.

IP-ADDRESS = <ipv4addr>
Remote IPv4-Adresse des Tunnel-Endpunkts. Die IPv4-Adresse muss in der üblichen „decimal dotted“-Schreibweise angegeben werden.

IPV6-ADDRESS = <ipv6addr>
Remote IPv6-Adresse des Tunnel-Endpunkts. Die IPv6-Adresse muss in der üblichen Sedezimaldarstellung mit Doppelpunkt (:) angegeben werden.

START-OF-TUNNEL (IP-ADDRESS = <ipv4addr > / IPV6-ADDRESS = <ipv6addr>)
spezifiziert die lokale Adresse des Tunnel-Endpunkts.

IP-ADDRESS = <ipv4addr>
Lokale IPv4-Adresse des Tunnel-Endpunkts. Die IPv4-Adresse muss in der üblichen „decimal dotted“-Schreibweise angegeben werden.

IPV6-ADDRESS = <ipv6addr>
Lokale IPv6-Adresse des Tunnel-Endpunkts. Die IPv6-Adresse muss in der üblichen Sedezimaldarstellung mit Doppelpunkt (:) angegeben werden.

SECURITY-ASSOCIATION = <alphanum-name 1 .. 32> / *NONE
Name der Security Association, die im Security Tunnel verwendet wird. Der SECURITY-ASSOCIATION-Satz muss bereits definiert sein.

COMPRESSION = DEFLATE / *NONE
Über den optionalen Operanden COMPRESSION kann die Komprimierung der Daten gefordert werden. Die Komprimierung erfolgt vor der Anwendung eines IPsec Protokolls. Der Parameterwert DEFLATE gibt den Komprimierungsalgorithmus an.
Der Defaultwert ist *NONE.

OUTER-TUNNEL = *NONE / (END-OF-TUNNEL (...) [,START-OF-TUNNEL(...)] ,SECURITY-ASSOCIATION = ...)
spezifiziert den IPSec-Tunnel der Security Policy. Er wird definiert durch die Operanden END-OF-TUNNEL, START-OF-TUNNEL (optional) sowie SECURITY-ASSOCIATION. Falls der START-OF-TUNNEL-Parameter nicht angegeben ist, ist der lokale Endpunkt des Tunnels die bei OWN-ADDRESS angegebene Adresse.
Das Schlüsselwort *NONE legt fest, dass kein OUTER-TUNNEL verwendet wird.

END-OF-TUNNEL (IP-ADDRESS = <ipv4addr > /
IPV6-ADDRESS = <ipv6addr>)
spezifiziert die remote Adresse des inneren Tunnel-Endpunkts. Der lokale Endpunkt des Tunnels wird via START-OF-TUNNEL-Parameter (siehe unten) bestimmt.

IP-ADDRESS = <ipv4addr>
Remote IPv4-Adresse des Tunnel-Endpunkts. Die IPv4-Adresse muss in der üblichen „decimal dotted“-Schreibweise angegeben werden.

IPV6-ADDRESS = <ipv6addr>
Remote IPv6-Adresse des Tunnel-Endpunkts. Die IPv6-Adresse muss in der üblichen Sedezimaldarstellung mit Doppelpunkt (:) angegeben werden.

START-OF-TUNNEL (IP-ADDRESS = <ipv4addr > /
IPV6-ADDRESS = <ipv6addr>)
spezifiziert die lokale Adresse des Tunnel-Endpunkts.

IP-ADDRESS = <ipv4addr>
Lokale IPv4-Adresse des Tunnel-Endpunkts. Die IPv4-Adresse muss in der üblichen „decimal dotted“-Schreibweise angegeben werden.

IPV6-ADDRESS = <ipv6addr>
Lokale IPv6-Adresse des Tunnel-Endpunkts. Die IPv6-Adresse muss in der üblichen Sedezimaldarstellung mit Doppelpunkt (:) angegeben werden.

SECURITY-ASSOCIATION = <alphanum-name 1 .. 32> / *NONE
Name der Security Association, die im Security Tunnel verwendet wird. Der SECURITY-ASSOCIATION-Satz muss bereits definiert sein.

COMPRESSION = DEFLATE / *NONE
Über den optionalen Operanden COMPRESSION kann die Komprimierung der Daten gefordert werden. Die Komprimierung erfolgt vor der Anwendung eines IPsec Protokolls. Der Parameterwert DEFLATE gibt den Komprimierungsalgorithmus an.
Der Defaultwert ist *NONE.

OWN-ADDRESS = ANY/ IP-ADDRESS = <ipv4addr> / IPV6-ADDRESS = <ipv6addr>
spezifiziert, auf welche eigenen Adressen die Security Policy angewendet werden kann.

OWN-ADDRESS = ANY
legt fest, dass die Security Policy auf alle eigenen Adressen angewendet werden kann.

OWN-ADDRESS = IP-ADDRESS = <ipv4addr>
spezifiziert eine eigene IPv4-Adresse, auf die die Security-Policy angewendet werden kann. Die IPv4-Adresse muss in der üblichen „decimal dotted“-Schreibweise angegeben werden.

OWN-ADDRESS = IPV6-ADDRESS = <ipv6addr>
spezifiziert eine eigene IPv6-Adresse, auf die die Security-Policy angewendet werden kann. Die IPv6-Adresse muss in der üblichen Sedezimaldarstellung mit Doppelpunkt (:) angegeben werden.

OWN-PORTNUMBER = ANY / <port#> / <port#-low - port#-high>
spezifiziert eine Portnummer oder einen Bereich von Portnummern. Auf die zugehörige(n) eigene(n) Anwendung(en) kann die Security Policy angewendet werden.
Der gültige Wertebereich für eine Portnummer liegt zwischen 0 und 65535.

OWN-PORTNUMBER = ANY
spezifiziert den Portnummernbereich von 0 bis 65535.

OWN-PORTNUMBER = <port#>
spezifiziert eine Portnummer.

OWN-PORTNUMBER = <port#-low> - <port#-high>
spezifiziert einen Portnummernbereich.

DIRECTION = IN / OUT / ANY
spezifiziert die Richtung des Datentransfers, der durch die Security Policy gesichert wird.

DIRECTION = IN
Die Policy sichert den eingehenden (incoming) Datentransfer.

DIRECTION = OUT
Die Policy sichert den ausgehenden (outgoing) Datentransfer.

DIRECTION = ANY
Die Policy sichert sowohl den eingehenden (incoming) als auch den ausgehenden (outgoing) Datentransfer.

PROTOCOL = TCP / UDP / ICMP, ... / ANY, ...
spezifiziert das Schicht4-Protokoll, das durch die Security Policy gesichert wird.

PROTOCOL = TCP
Die Policy sichert den TCP-Datentransfer.

PROTOCOL = UDP
Die Policy sichert den UDP-Datentransfer.

PROTOCOL = ICMP, ...
Die Policy sichert den ICMP-Datentransfer.

PROTOCOL = ANY, ...
Die Policy sichert sowohl den TCP- , den UDP als auch den ICMP-Datentransfer.
Bei der Angabe von PROTOCOL=ANY werden folgende PARAMETER gesetzt:
PARTNER-PORTNUMBER=ANY
OWN-PORTNUMBER=ANY
ICMP-CODE=ANY
ICMP-TYPE=ANY

ICMP-CODE =
ICMP-Code oder Bereich von ICMP-Codes, auf die die Policy anwendbar ist.
Der gültige Wertebereich für einen ICMP-Code liegt zwischen 0 und 255.
Dieser Parameter ist nur anzugeben, wenn PROTOCOL = ICMP oder ANY ist.

ICMP-CODE = ANY
Der ICMP-Typ ist beliebig.

ICMP-CODE = <code>
spezifiziert einen bestimmten ICMP-Code.

ICMP-CODE = <code-low> - <code-high>
spezifiziert einen Bereich von ICMP-Codes.

ICMP-TYPE =
ICMP-Typ oder Bereich von ICMP-Typen, auf die die Policy anwendbar ist.
Der gültige Wertebereich für ICMP-Typen liegt zwischen 0 und 255.
Dieser Parameter ist nur anzugeben, wenn PROTOCOL = ICMP oder ANY ist.

ICMP-TYPE = ANY
Der ICMP-Typ ist beliebig.

ICMP-TYPE = <type>
spezifiziert einen bestimmten ICMP-Typ.

ICMP-TYPE = <type-low> - <type-high>
spezifiziert einen Bereich von ICMP-Typen.

PARTNER-SAS: Automatisch erzeugte SAs löschen

Durch den PARTNER-SAS-Satz wird im DELETE-Modus definiert, welche dynamisch erzeugten SAs (MODE = IKE) gelöscht werden sollen. Im ADD-Modus hat der Satz keine Wirkung.

PARTNER-SAS
IP-ADDRESS = < Ipv4-addr > / IPV6-ADDRESS = < Ipv6-addr >

Beschreibung der Operanden

IP-ADDRESS = <Ipv4-addr>
IPv4-Adresse, die das Partnersystem angibt, dessen SAs gelöscht werden sollen.

IPV6-ADDRESS = <Ipv6-addr>
IPv6-Adresse, die das Partnersystem angibt, dessen SAs gelöscht werden sollen.

6.3.3 Überprüfen der IPSec-Konfiguration

Nach dem Erstellen oder Modifizieren einer IPSec-Konfigurationsdatei ist es ratsam die syntaktische Korrektheit einer einzelnen geänderten Datei und/oder die zu erwartende IP-Sec-Konfiguration zu prüfen. Dazu bietet IPSec ein Programm an.

START-IPSEC-DB-CHECK

Der Aufruf erfolgt mit:

► `/START-IPSEC-DB-CHECK`

Danach wird die folgende Startmeldung ausgegeben:

```
IPSEC-DB-CHECK Version n.n loaded
```

Sie werden nun aufgefordert, den Namen der zu überprüfenden Konfigurationsdatei einzugeben.

```
IPSEC-DB-CHECK Enter filename of configurationfile or *none or *end
```

– Bei Eingabe eines Dateinamens wird der Syntax-Check für diese Konfigurationsdatei durchgeführt. Wenn die angegebene Konfigurationsdatei nicht vorhanden ist, wird die folgende Fehlermeldung ausgegeben:

  `IPSEC-DB-CHECK could not open configurationfile` *`dateiname`*

  Anschließend werden Sie erneut aufgefordert, den Namen der zu überprüfenden Konfigurationsdatei anzugeben.

– Bei Eingabe von `*none` werden die zu prüfenden Konfigurationssätze von *SYSDTA gelesen.

  ► Beenden Sie die Eingabe von *SYSDTA mit `END`.

– Bei Eingabe von `*end` wird die Ausführung von START-IPSEC-DB-CHECK ohne weitere Aktionen beendet.

Das Ergebnis der Syntaxüberprüfung wird in eine Logging-Datei geschrieben.

Mit der folgenden Ausgabe werden Sie aufgefordert, die Logging-Datei zu spezifizieren:

```
IPSEC-DB-CHECK Enter filename of loggingfile or *none or *end
```

– Bei Eingabe eines Dateinamens werden die Loggingsätze in eine Datei dieses Namens geschrieben:

  – Wenn die Datei noch nicht existiert, wird sie neu angelegt.

  – Wenn die Datei bereits vorhanden ist, wird sie überschrieben.

– Bei Eingabe von *none werden die Loggingsätze nach *SYSOUT geschrieben.
– Bei Eingabe von *end wird die Ausführung von START-IPSEC-DB-CHECK ohne weitere Aktionen beendet.

Mit der folgenden Meldung wird angezeigt, dass mit der Überprüfung der Konfigurationsdatei begonnen wurde:

```
IPSEC-DB-CHECK starting
```

Jede INCLUDE-Anweisung wird wie folgt protokolliert:

```
IPSEC-DB-CHECK including file dateiname
IPSEC-DB-CHECK including file dateiname terminated
```

Wenn die Konfigurationsdatei fehlerfrei gelesen werden konnte, werden die folgenden Meldungen ausgegeben:

IPSEC configurationfile *dateiname* read without error and without warning.

bzw., falls Warnungen ausgegeben werden:

IPSEC configurationfile *dateiname* read with n warning(s).
IPSEC-DB-CHECK terminated

Wenn beim Lesen der Konfigurationsdatei Fehler festgestellt wurden, werden die folgenden Meldungen ausgegeben:

IPSEC configuration dateiname read with *x* error(s) and n warning(s)..
See logging file *logging-dateiname* for more information.
IPSEC-DB-CHECK terminated

Struktur der Logging-Datei

Am Anfang der Logging-Datei stehen die beiden folgenden Informationssätze:

```
IPSEC-DB-CHECK: Logging for test of IPSEC configurationfile file dateiname
IPSEC-DB-CHECK: Test performed at datum uhrzeit
```

Darauf folgen die einzelnen Zeilen der gelesenen Konfigurationsdatei. Jeder gelesenen Zeile ist die folgende Meldung vorangestellt:

```
Read line zeilennummer
```

Nachdem ein Konfigurationssatz erfolgreich eingelesen werden konnte, wird die folgende Meldung ausgegeben:

```
New X Record created
```

(*X* steht für Key / Signature / Security Association)

Konnte ein POLICY-Satz erfolgreich eingelesen werden, wird die folgende Meldung ausgegeben:

```
New Policy created
```

Für jede aus der Policy abgeleitete SA wird die folgende Meldung ausgegeben:

```
New i1 i2 SA created
```

(i1 steht für inbound oder outbound. i2 steht für First, Second, inner tunnel oder outer tunnel)

Führt eine SA-Ableitung zu einem Duplikat wird folgender Hinweis ausgegeben:

```
Warning: i1 i2 SA according to POLICY record already in database (i3)
```

(i1, i2 wie oben; i3 Name des Duplikats)

Wenn bei der Überprüfung der Konfigurationsdatei ein Fehler festgestellt wird, wird in der nächsten Zeile die folgende Meldung ausgegeben:

```
ERROR: erläuternder Text
```

Kann der Fehler einer bestimmten Spalte (in der aktuellen Zeile der Konfigurationsdatei) zugeordnet werden, dann wird diese Spalte mit dem Zeichen „^'“markiert.

Die gemeldeten Fehler müssen behoben werden.

In die Logging-Datei wird anschließend ausgegeben, welche SPD und SAD aus der geprüften Konfigurationsdatei erzeugt worden wäre.

6.3.4 IKE-Konfiguration

Im BS2000/OSD erbringt der IKE-Daemon Racoon2 in Zusammenarbeit mit dem Security Policy-Daemon SPMD die Funktionen der automatisierten SA- und Schlüsselverwaltung. Beide Programme können über Anweisungen in einer Konfigurationsdatei gesteuert werden.

Bei der Installation des Subsystems IPSec wird unter dem Namen $TSOS.SYSDAT.IPSEC.*nnn*.RAC2 eine Defaultdatei mit entsprechenden Anweisungen erzeugt. Dort finden sich neben Defaultwerten für Protokoll-Parameter auch Angaben, mit welchen Portnummern Racoon2 und SPMD arbeiten sollen und in welcher Datei z.B. das SPMD-Passwort steht. Diese Datei muss ebenso wie die PSK-Dateien vom Systemverwalter erstellt werden. Das Subsystem IPsec und die Programme SPMD und Racoon2 können mit der Defaultdatei gestartet werden.

Auf diese Datei wird im vorliegenden Handbuch unter dem Namen $TSOS.SYSDAT.IPSEC.*nnn*.RAC2.MANUAL-BSP-DEFAULT Bezug genommen. Sie kann mit Hilfe der INCLUDE-Anweisung in zu erstellenden Konfigurationsdateien eingebunden werden. Es ist daher empfehlenswert, die installierte Defaultdatei zu sichern.

Defaultdatei SYSDAT.IPSEC.*nnn*.RAC2

```
# The SPMD/Racoon2 default configuration file.
setval
{
   PSKDIR   " ";
};
# interface info
# Which address should the daemons use.
interface
{
# Racoon2 listens with UDP port 500 at every system's IP address.
#
   ike
  {
      MY_IP;
   };
# A specific IP address and/or an alternate port number must be definied
# here.
# Several instructions are possible concurrently.
# The support of NAT-Traversal requires an additional instruction of the
# form  MY_IP port 4500;
#
# SPMD listens  with TCP Portnumber 3333 at the loopback address.
# This is BS2000/OSD special and is needed for communication between
# Racoon2 and SPMD.
    spmd
```

```
    {
       127.0.0.1 port 3333;
    };

# SPMD's logon password is located in the file $(PSKDIR)SPMD.PWD
#
   spmd_password "spmd.pwd";
};
#
# default section
#
default
{
#
# remote describes the KMP peer.
   remote
  {
      ikev1
     {
         proposal_check obey;
         logmode normal;
         kmp_sa_lifetime_time 3600 sec;
         kmp_sa_lifetime_byte 0;
         interval_to_send 10 sec;
         times_per_send 1;
         ipsec_sa_nego_time_limit 40 sec;
         kmp_enc_alg
   {
        3des_cbc;
     };
         kmp_hash_alg
     {
       md5;
     };
         kmp_dh_group
     {
       modp1536;
       modp1024;
       modp768;
     };
         kmp_auth_method
     {
       psk;
     };
         random_pad_content on;
      };
      ikev2
   {
```

```
            logmode normal;
            kmp_sa_lifetime_time 0;
            kmp_sa_lifetime_byte 0;
            max_retry_to_send 3;
            interval_to_send 10 sec;
            times_per_send 1;
            ipsec_sa_nego_time_limit 40 sec;
            kmp_enc_alg
    {
         3des_cbc;
       };
            kmp_prf_alg
       {
         hmac_md5;
        };
            kmp_hash_alg
       {
         hmac_md5;
       };
            kmp_dh_group
       {
         modp1536;
         modp2048;
         modp3072;
       };
            kmp_auth_method
       {
         psk;
       };
            random_pad_content on;
            random_padlen on;
            max_padlen 50 bytes;
        };
   };
# kmp_prf_alg is not needed for IKEv1.
# The notation of the kmp_hash_alg algorithms are distinct in IKEv1 and
# IKEv2.
# The _lifetime value 0 means infinite.
# Authentication method must be preshared secret key. The key must be
# included  in a file. The file name should be specified in a remote
# directive.
#
# The following specifies a minimal default policy directive.
#
   policy
  {
      ipsec_mode transport;
      ipsec_level require;
```

```
    };
 # The following specifies a minimal default ipsec directive.
#
    ipsec
   {
        ipsec_sa_lifetime_time 0;
        ipsec_sa_lifetime_byte 0;
    };

# The following specifies a minimal default sa directive.
#
    sa
    {
        esp_enc_alg
    {
          aes128_cbc;
          3des_cbc;
    };

        ah_auth_alg
    {
          hmac_sha1;
          hmac_md5;
    };
    };
};
```

Eine weiterführende Dokumentation ist auf der Racoon2 homepage zu finden:

http://www.racoon2.wide.ad.jp/w/

6.4 IPSec-Subsystem Bedienung

In den voran gegangenen Kapitel wurden ausführlich die für den IPSec und IKE-Betrieb erforderlichen Dateien im BS2000/OSD beschrieben. Vor dem Start des Subsystems IPSec sollte überprüft werden, ob die Dateien installiert sind, und welchen Inhalt sie haben. Für einen sinnvollen IPSec-Betrieb ist Folgendes zu erstellen:

- IPSec-Konfigurationsdateien
- Preshared Secret Key Dateien (PSK) für die einzelnen Partner.
- SPMD.PWD (Datei, die das LOGON-Password für SPMD enthält).

SYSSSI-Datei

Beim Start des Subsystems IPSec werden die Anweisungen der Datei SYSSSI.IPSEC.*nnn* ausgeführt. Im Auslieferungszustand enthält die Datei folgendes Kommando:

```
LOAD-IPSEC-DB FROM-FILE=$TSOS.SYSDAT.IPSEC.nnn.CONF.
```

IPSec wird angewiesen, die IPSec-Datenbasis beginnend mit der Standardkonfigurationsdatei aufzubauen. Der Name der Datei kann an dieser Stelle geändert werden, um die Generierung der IPSec Datenbasis mit einer anderen Datei zu starten.

Wenn bei der Subsystem-Aktivierung festgestellt wird, dass keine IPSec-Konfigurationsdatei mit dem in der SSINFO-Datei spezifizierten Namen vorhanden ist, wird der Start des Subsystems mit Fehler abgebrochen.

Sollte die Defaultportnummer 3500 für die Kommunikation zwischen IPSec und dem IKE-Daemon bereits durch ein anderes Systemprogramm belegt sein, kann IPSec durch folgende Anweisung in der SYSSSI-Datei veranlasst werden, eine alternative UDP-Portnummer zu verwenden:

```
SET-PFKEY-PARAMETER PORT=<port number>
```

IPSec-Subsystem aktivieren und deaktivieren

Das IPSec-Subsystem wird mit folgendem BS2000/OSD-Kommando aktiviert:

```
/START-SUBSYSTEM IPSEC
```

Das Programm SPMD wird dabei mitgestartet. SPMD startet das Programm Racoon2.

Das IPSec-Subsystem wird mit folgendem BS2000/OSD Kommando deaktiviert (angehalten):

```
/STOP-SUBSYSTEM IPSEC
```

Die Programme SPMD und Racoon2 werden ebenfalls beendet.

6.5 IPSec-Subsystem Administration

6.5.1 IPSec-Konfiguration ändern

Die IPSec-Konfiguration kann im laufenden Betrieb verändert werden. Normalerweise müssen Sicherheitsrichtlinien neu definiert, verändert oder gelöscht werden. In Ausnahmefällen, z.B. wenn ein IPSec Partnersystem temporär nicht erreichbar ist oder neu gestartet wurde, kann es erforderlich sein Sicherheitsverbindungen zu löschen. Die einzige dafür vorgesehene Möglichkeit ist eine Konfigurationsdatei zu erstellen, die die Veränderung der Datenbasis beschreibt, und diese Datei einzulesen.

Beispiel

Innerhalb der Konfigurationsdatei werden die Anweisungen ADD und DELETE eingesetzt.

Die Datei MODIFY.DEFAULTPOLICY ändert die Default-Policy von BYPASS auf DISCARD.

```
*
DELETE
*
POLICY NAME=THEOTHERS
*
ADD
*
*
* definition of POLICY records
*
POLICY NAME=THEOTHERS -
,POLICY-RANGE=GLOBAL -
,OWN-ADDRESS=ANY -
,OWN-PORTNUMBER=ANY -
,DIRECTION=ANY -
,PROTOCOL=ANY -
,ICMP-TYPE=ANY -
,ICMP-CODE=ANY -
,MODE = DISCARD
```

Ein anschließendes Kommando

```
/LOAD-IPSEC-DB FROM-FILE=MODIFY.DEFAULTPOLICY
```

modifiziert die IPSec-Konfiguration.

LOAD-IPSEC-DB: IPSec Database laden

Mit dem Kommando LOAD-IPSEC-DB wird eine IPSec-Konfigurationsdatei eingelesen und dadurch die IPSec Datenbasis verändert.

Als SDF-Kurzname existiert für LOAD-IPSEC-DB der Aliasname LDIPSDB.

BS2000-Konsole	BS2000-SDF-Kommando	Kommando-/SOF-Datei	SNMP-Management	Parameter-service
	x			

Auflistung zulässiger Kommandoquellen

LOAD-IPSEC-DB/LDIPSDB IPSEC Data Base laden
FROM-FILE = **<u>*UNCHANGED</u>** / ***STD** / **<full-filename 1 .. 54>**

Beschreibung der Operanden

FROM-FILE =
spezifiziert den Namen der IPSec-Konfigurationsdatei.

FROM-FILE = <u>*UNCHANGED</u>
Der aktuell eingestellte Dateiname wird verwendet (Defaultwert).

FROM-FILE = *STD
Es wird eine IPSec-Konfigurationsdatei mit Standard-Dateinamen $TSOS.SYSDAT.IPSEC.*nnn*.CONF verwendet (*nnn* spezifiziert die IPSec Version, z.B. 014 für IPSec V1.4).

FROM-FILE = <full-filename 1 .. 54>
Es wird eine IPSec-Konfigurationsdatei mit einem vom Standard-Dateinamen abweichenden Namen verwendet.

Kommando-Protokollierung

- Eine korrekte Ausführung des LOAD-IPSEC-DB-Kommandos wird mit der Meldung YIS0211 quittiert. Zusätzlich kann die Meldung YIS0219 ausgegeben werden.
- Eine fehlerhafte Ausführung des LOAD-IPSEC-DB-Kommandos wird mit der Meldung YIS0210 quittiert. Zusätzlich kann die Meldung YIS0218 ausgegeben werden.

Kommando-Returncodes

Die folgende Tabelle fasst die bei der Kommandoverarbeitung möglichen Returncodes zusammen.

(SC2)	SC1	Maincode	Bedeutung
0	0	CMD0001	Kommando erfolgreich abgearbeitet
0	1	CMD0202	Operandenfehler
0	32	CMD0221	Systemfehler
0	64	CMD0216	Privileg nicht ausreichend

Beispiele

Die IPSec Data Base soll bei bereits generiertem IPSec-Subsystem mit den Daten der IPSec-Konfigurationsdatei SYSDAT.IPSEC.TEST.CONF geladen werden:

```
/LOAD-IPSEC-DB FROM-FILE=SYSDAT.IPSEC.TEST.CONF
```

Die IPSec Data Base soll bei bereits generiertem IPSec-Subsystem mit den Daten einer IPSec-Konfigurationsdatei mit Standard-Dateinamen geladen werden:

```
/LOAD-IPSEC-DB FROM-FILE=*STD
```

Automatisierte Konfiguration ändern

Die Änderung von Sicherheitsrichtlinien einer automatisierten Konfiguration erfolgt analog. Lediglich die Anweisungen für das Löschen und Hinzufügen einer Sicherheitsrichtlinie können nicht in eine Konfigurationsdatei geschrieben werden. Die Anweisungen müssen Sie als Eingabeparameter von Dienstprogrammen angeben.

```
/START-PROG *M(SYSLNK.IPSEC.nnn.RAC2,CTRLPROG,A,A)

%  BLS0523 ELEMENT 'CTRLPROG', VERSION '300', TYPE 'L' FROM LIBRARY
':BO7A:$TSOS.SYSLNK.IPSEC.nnn.RAC2' IN PROCESS
%  BLS0524 LLM 'CTRLPROG', VERSION '300' OF '2009-02-09 13:07:29' LOADED
%  CCM0001 ENTER OPTIONS:
*policy delete DIAL-in

/START-PROG *M(SYSLNK.IPSEC.nnn.RAC2,CTRLPROG,A,A)
*policy delete DIAL-out
```

Mithilfe dieses Steuerungsprogrammes löschen Sie die ursprünglich definierte Policy, die den Zugang zu der Anwendung $DIALOG von dem PC mit der IP-Adresse 172.25.83.42 ermöglicht (siehe Seite 77).

Soll nun der Dialogzugang einer größeren Anzahl von PCs ermöglicht werden, können Sie z.B. eine Konfigurationsdatei MODIFY.DIALOG mit folgendem Inhalt erstellen:

```
# PC-1
# Describes the KMP capabilities of PC-1
# Parameters of the default section are overridden.
#
remote PC-1 {
   acceptable_kmp { ikev1; };
   ikev1 {
      peers_ipaddr 172.25.83.0/24 port 500;
      kmp_enc_alg { 3des_cbc;};
      kmp_hash_alg { sha1; md5;};
      kmp_dh_group { modp1024; };
      kmp_auth_method { psk; };
      pre_shared_key "PC-1.psk";
   };
};
#
# $DIALOG
# Selector and policy definition for the communication to $DIALOG from
# a range of partners.
#
selector DIAL-in {
   direction inbound;
   src 172.25.83.0/24;
   dst 172.25.92.72 port 1110;
```

```
   upper_layer_protocol "tcp";
   policy_index PC-1;
};
selector DIAL-out {
   direction outbound;
   dst 172.25.83.0/24;
   src 172.25.92.72 port 1110;
   upper_layer_protocol "tcp";
   policy_index PC-1;
};
```

Die Konfigurationsdatei, die den Dialogzugang erweitert, wird dem Security Policy-Daemon SPMD bzw. dem IKE-Daemon Racoon2 mit folgendem Kommando bekannt gemacht:

```
/MODIFY-SERVICE-PARAMETER SERVICE-NAME=$YISSPMD,START-PARAMETER= -
/  '-A -f  MODIFY.DIALOG'

/MODIFY-SERVICE-PARAMETER SERVICE-NAME=$YISRAC2,START-PARAMETER= -
/  '-A -f  MODIFY.DIALOG'
```

Die Option -A bewirkt, dass der Inhalt der Datei, die in der Option -f angegeben wurde, in die aktuelle Konfiguration eingefügt wird.

Alternativ dazu können Sie die Konfigurationsdatei SYSDAT.IPSEC.nnn.RAC2 entsprechend anpassen und mit folgenden Kommandos neu einlesen:

```
/MODIFY-SERVICE-PARAMETER SERVICE-NAME=$YISSPMD,START-PARAMETER= -
/  '-R -f SYSDAT.IPSEC.nnn.RAC2'

/MODIFY-SERVICE-PARAMETER SERVICE-NAME=$YISRAC2,START-PARAMETER= -
/  '-R -f SYSDAT.IPSEC.nnn.RAC2'
```

SPMD löscht dann alle aktuellen Security Policies aus dem IPSec-Kernel und baut die Konfiguration gemäß der neuen Beschreibung auf.

6.5.2 IPSec-Monitoring

Das IPSec-Monitoring informiert über den IPSec-Betrieb via Protokollierung in die aktuelle CONSLOG-Datei.

Die Protokollierung umfasst die folgenden IPSec-Meldungen:

- YIS0310 (Datenelemente)
- YIS0311 (openCRYPT)
- YIS0312 (Fehler in IPSec)

START-IPSEC-MONITORING / SRIPSMN: IPSec-Monitoring einschalten

Als SDF-Kurzname existiert für START-IPSEC-MONITORING der Aliasname SRIPSM.

BS2000-Konsole	BS2000-SDF-Kommando	Kommando-/SOF-Datei	SNMP-Management	Parameter-service
	x			

Auflistung zulässiger Kommandoquellen

START-IPSEC-MONITORING / **SRIPSMN IPSEC Monitoring starten**
TIMER = **<u>*STD</u>** / <integer 5..32765>

TIMER = ...
spezifiziert das Timer-Intervall für die Aktualisierung der Monitoring Ausgabe.

TIMER = <u>*STD</u>
Als Timer-Intervall wird der Standardwert (5 Sekunden) eingestellt.

TIMER = <integer 5 .. 32765>
Als Timer-Intervall wird der angegebene ganzzahlige Wert (Einheit: Sekunden) eingestellt.

Kommando-Protokollierung

- Eine korrekte Ausführung des START-IPSEC-MONITORING-Kommandos wird mit der Meldung `YIS0223` quittiert.
- Eine fehlerhafte Ausführung des Kommandos wird mit der Meldung `YIS0222` quittiert.

Kommando-Returncodes

Die folgende Tabelle fasst die bei der Kommandoverarbeitung möglichen Returncodes zusammen.

(SC2)	SC1	Maincode	Bedeutung
0	0	CMD0001	Kommando erfolgreich abgearbeitet
0	1	CMD0202	Operandenfehler
0	32	CMD0221	Systemfehler
0	64	CMD0216	Privileg nicht ausreichend

Beispiel

Das IPsec-Monitoring soll eingeschaltet werden. Die Anzeige soll in Abständen von 10 Minuten (600 Sekunden) aktualisiert werden:

```
/START-IPSEC-MONITORING TIMER=600
```

STOP-IPSEC-MONITORING / SPIPSMN: IPSec-Monitoring ausschalten

Als SDF-Kurzname existiert für STOP-IPSEC-MONITORING der Aliasname SPIPSMN.

BS2000-Konsole	BS2000-SDF-Kommando	Kommando-/SOF-Datei	SNMP-Management	Parameter-service
	x			

Auflistung zulässiger Kommandoquellen

STOP-IPSEC-MONITORING / **SPIPSMN IPSEC Monitoring beenden**

Kommando-Protokollierung

- Eine korrekte Ausführung des STOP-IPSEC-MONITORING-Kommandos wird mit der Meldung `YIS0225` quittiert.
- Eine fehlerhafte Ausführung des Kommandos wird mit der Meldung `YIS0224` quittiert.

Kommando-Returncodes

Die nachfolgende Tabelle fasst die bei der Kommandoverarbeitung möglichen Returncodes zusammen

(SC2)	SC1	Maincode	Bedeutung
0	0	CMD0001	Kommando erfolgreich abgearbeitet
0	1	CMD0202	Operandenfehler
0	32	CMD0221	Systemfehler
0	64	CMD0216	Privileg nicht ausreichend

Beispiel

Das IPSec-Monitoring soll beendet werden:

```
/STOP-IPSEC-MONITORING
```

6.5.3 Diagnoseunterlagen erstellen

Sollte es im IPSec-Betrieb zu einer Fehlersituation kommen, die Sie selbst nicht beheben können, wenden Sie sich bitte an Ihren Ansprechpartner.

Für eine effiziente Fehlersuche benötigt Ihr Ansprechpartner folgende Informationen:

- genaue Beschreibung der Fehlersituation sowie Angaben, ob der Fehler reproduzierbar ist
- Beschreibung der Hardware-Konfiguration
- Informationen zur Software-Konfiguration mit Angaben zu Art und Umfang der eingesetzten Betriebssystem- und IPSec-Software

  Hier müssen auch die entsprechenden Versionsnummern und eventuell verwendete Rep-Korrekturen dokumentiert werden.
- vollständiges Blattschreiberprotokoll bzw. die $SYSAUDIT.SYS.CONSLOG-Datei der BS2000/OSD-Session
- Dumps
- $TSOS.SYS.SERSLOG-Datei
- ggf. IPSec-Monitoring-Unterlagen
- IPSec-Traces (siehe unten)
- IPSECDIA-Unterlagen (siehe unten)
- Racoon2-Traces
- SPMD-Traces
- BCAM-Diagnose-Unterlagen. Diese erzeugen Sie mithilfe des Dienstprogramms ASTRID (siehe Handbuch „BCAM V21.0A“).

IPSec Traces erstellen

Die IPSec-Traces erstellen Sie wie folgt:

► Schalten Sie den IPSec-Trace mit dem Kommando /DCDIAG ein.

  – Der Name für den IPSec-Benutzerschnittstellen-Trace lautet IPSEC.COM.

  – Weitere Einzelheiten zum Kommando /DCDIAG entnehmen Sie bitte dem Handbuch „BCAM V21.0A“.

Racoon2-Traces erstellen

Die Racoon2-Traces erstellen Sie wie folgt:

- Geben Sie das folgende Kommando ein

```
/MODIFY-SERVICE-PARAMETER SERVICE-NAME=
        $YISRAC2, START-PARAMETER= '-D 7 -l <dateiname>'
```

  Für <dateiname> geben Sie einen Namen ihrer Wahl an.

- Sind die Diagnoseinformationen geschrieben, geben Sie das Kommando erneut mit folgenden Parametern ein:

```
/MODIFY-SERVICE-PARAMETER SERVICE-NAME=$YISRAC2, START-PARAMETER= '-D 0'
```

SPMD-Traces erstellen

Die SPMD-Traces erstellen Sie wie folgt:

- Geben Sie das folgende Kommando ein:

```
/MODIFY-SERVICE-PARAMETER SERVICE-NAME=$YISSPMD,
        START-PARAMETER= '-D 10 -l <dateiname>'
```

  Für <dateiname> geben Sie einen Namen ihrer Wahl an.

- Sind die Diagnoseinformationen geschrieben, geben Sie das Kommando erneut mit folgenden Parametern ein:

```
/MODIFY-SERVICE-PARAMETER SERVICE-NAME=$YISSPMD, START-PARAMETER= '-D 0'
```

IPSECDIA-Unterlagen erstellen

Die IPSECDIA-Informationen erstellen Sie wie folgt:

- Weisen Sie SYSLST auf eine Datei zu.
- Starten Sie das IPSec-Diagnoseprogramm: `IPSECDIA`.
- Geben Sie die Anweisung `TOTAL` ein.
- Geben Sie die Anweisung `END` ein.
- Nehmen Sie die Umlenkung der SYSLST-Ausgabe zurück.

6.6 Konfigurationsbeispiele

Im Folgenden wird die Anwendung der IPSec-Sicherheitsmechanismen anhand von Beispielkonfigurationen veranschaulicht.

6.6.1 Konfiguration mit manuell und automatisch definierten Schlüsseln

In diesem Bespiel sollen folgende Sicherheitsrichtlinien gelten, die durchgesetzt werden sollen:

1. Kommunikationsbeziehungen ohne IPSec-Sicherheitsmechanismen werden nicht zugelassen.
2. Jeglicher Datenverkehr innerhalb des eigenen Subnetzes 172.25.92.0/24 ist zu verschlüsseln.
3. Der Datenverkehr zwischen den IP-Adressen 172.25.92.72 und 172.25.92.74 ist mit einem eigenen Schlüssel zu sichern.
4. Der FTP-Datenverkehr zwischen den Systemen mit den IP-Adressen 172.25.92.72 und 172.25.92.74 ist zu authentifizieren und zu verschlüsseln. Dazu ist es notwendig den FTP-Server-Port 21 im Partnersystem zu schützen. Im Gegensatz dazu muss der FTP-Daten-Port 20 im eigenen System geschützt werden. Die Konfiguration erfolgt also aus der Sicht des FTP-Client.
5. Es soll der Transportmodus verwendet werden.
6. Verschlüsselungsverfahren ist AES.
7. Signaturverfahren ist HMAC-SHA1.
8. Die Schlüssel werden manuell vergeben. Sie sind in einer eigenen Datei zu verwalten.

6.6.1.1 Manuell definierte Schlüssel in einer IPSec-Konfiguration

Die Konfigurationsdateien IPSEC.KEYS und IPSEC.CONF sehen dann wie folgt aus:

Datei IPSEC.KEYS

```
KEY NAME=KEYAES1 -
,KEY-ALGORITHM=AES-CBC -
,KEY-VALUE=X'344B69566B3742535466433756457832'
_____________________________________________________________________(7)
*
KEY NAME=KEYAES11 -
,KEY-ALGORITHM=AES-CBC -
,KEY-VALUE=X'5077776952585966476e386934553766'
_____________________________________________________________________(6)
*
KEY NAME=KEYAES2 -
,KEY-ALGORITHM=AES-CBC -
,KEY-VALUE=X'73786472580a72476252537158384e4d'
_____________________________________________________________________(5)
*
KEY NAME=KEYAES3 -
,KEY-ALGORITHM=AES-CBC -
,KEY-VALUE=X'2f6b2f364964567163624b44386e4238'
_____________________________________________________________________(4)
*
SIGNATURE NAME=KEYSHA1 -
,SIGNATURE-ALGORITHM=HMAC-SHA-1 -
,SIGNATURE-VALUE=X'396767586A4435464830473848543647 4B64652F'
_____________________________________________________________________(6,7)
```

Datei IPSEC.CONF

```
FLUSH
_____________________________________________________________________(1)
*
INCLUDE IPSEC.KEYS
_____________________________________________________________________(2)
*
* definition of SA records
*
SECURITY-ASSOCIATION NAME=SA1 -
,TYPE=ENCRYPTION( -
 SIGNATURE=KEYSHA1 -
,KEY=KEYAES1) -
```

```
,MODE=MANUAL(INDEX=257) -
,ECN-TUNNEL= ALLOWED -
,ECN-TUNNEL-NEGOTIATION= YES
_____________________________________________________________________(7)
*
SECURITY-ASSOCIATION NAME=SA11 -
,TYPE=ENCRYPTION( -
 SIGNATURE=KEYSHA1 -
,KEY=KEYAES11) -
,MODE=MANUAL(INDEX=258) -
,ECN-TUNNEL= ALLOWED -
,ECN-TUNNEL-NEGOTIATION= YES
_____________________________________________________________________(6)
*
SECURITY-ASSOCIATION NAME=SA2 -
,TYPE=ENCRYPTION( -
 SIGNATURE=*NONE -
,KEY=KEYAES2) -
,MODE=MANUAL(INDEX=300) -
,ECN-TUNNEL= ALLOWED -
,ECN-TUNNEL-NEGOTIATION= YES
_____________________________________________________________________(5)
*
SECURITY-ASSOCIATION NAME=SA3 -
,TYPE=ENCRYPTION( -
 SIGNATURE=*NONE -
,KEY=KEYAES3) -
,MODE=MANUAL(INDEX=400) -
,ECN-TUNNEL= ALLOWED -
,ECN-TUNNEL-NEGOTIATION= YES
_____________________________________________________________________(4)
*
* definition of POLICY records
*
POLICY NAME=THEOTHERS -
,POLICY-RANGE=GLOBAL -
,OWN-ADDRESS=ANY -
,OWN-PORTNUMBER=ANY -
,DIRECTION=ANY -
,PROTOCOL=ANY -
,ICMP-TYPE=ANY -
,ICMP-CODE=ANY -
,MODE = DISCARD
_____________________________________________________________________(3)
*
POLICY NAME=MYFRIENDS -
,POLICY-RANGE=PARTNERSYSTEM (IP-ADDRESS=172.25.92.0-172.25.92.255) -
,OWN-ADDRESS=ANY -
```

```
,OWN-PORTNUMBER=ANY -
,DIRECTION=ANY -
,PROTOCOL=ANY -
,ICMP-TYPE=ANY -
,ICMP-CODE=ANY -
,MODE = TRANSPORT -
,FIRST-SECURITY-ASSOCIATION=SA3 -
,SECOND-SECURITY-ASSOCIATION=*NONE
_____________________________________________________________________(4)
*
POLICY NAME=VM4 -
,POLICY-RANGE=PARTNERSYSTEM (IP-ADDRESS=172.25.92.74) -
,OWN-ADDRESS=IP-ADDRESS = 172.25.92.72 -
,OWN-PORTNUMBER=ANY -
,DIRECTION=ANY -
,PROTOCOL=ANY -
,ICMP-TYPE=ANY -
,ICMP-CODE=ANY -
,MODE = TRANSPORT -
,FIRST-SECURITY-ASSOCIATION=SA2 -
,SECOND-SECURITY-ASSOCIATION=*NONE
_____________________________________________________________________(5)
*
POLICY NAME=FTPDATA -
,POLICY-RANGE=PARTNER (IP-ADDRESS=172.25.92.74,PORTNUMBER = 20) -
,OWN-ADDRESS=IP-ADDRESS = 172.25.92.72 -
,OWN-PORTNUMBER=ANY -
,DIRECTION=ANY -
,PROTOCOL=TCP -
,MODE = TRANSPORT -
,FIRST-SECURITY-ASSOCIATION=SA11 -
,SECOND-SECURITY-ASSOCIATION=*NONE
_____________________________________________________________________(6)
*
POLICY NAME=FTPSERVER -
,POLICY-RANGE=PARTNER (IP-ADDRESS=172.25.92.74,PORTNUMBER = 21) -
,OWN-ADDRESS=IP-ADDRESS = 172.25.92.72 -
,OWN-PORTNUMBER=ANY -
,DIRECTION=ANY -
,PROTOCOL=TCP -
,MODE = TRANSPORT -
,FIRST-SECURITY-ASSOCIATION=SA1 -
,SECOND-SECURITY-ASSOCIATION=*NONE
_____________________________________________________________________(7)
```

Erläuterungen

(1) Eventuell vorhandene Einträge in der SPD und SAD werden gelöscht.

(2) Die Datei mit den Schlüsseln wird eingelesen. Hiermit ist Punkt 8. realisiert.

(3) Alle Datenpakete von allen eigenen Adressen zu beliebigen Partneradressen werden verworfen (Firewall outbound, Punkt 1.).

(4) Der Datenverkehr im eigenen Subnetz wird im Transportmodus gesichert. Verwendet wird das Verschlüsselungsverfahren AES (Punkte 2., 5. und 6.).

(5) Der Datenverkehr von der eigenen IP-Adresse 172.25.92.72 zum Rechner mit der IP-Adresse 172.25.92.74 wird im Transportmodus gesichert. Verwendet wird das Verschlüsselungsverfahren AES (Punkte 3., 5. und 6.).

(6) Der Datenverkehr von beliebigen Ports der eigenen IP-Adresse 172.25.92.72 zum FTP-Datenport auf der Anlage 172.25.92.74 wird im Transportmodus gesichert. Verwendet wird das Verschlüsselungsverfahren AES und das Authentifizierungsverfahren HMAC-SHA1 (Punkte 4., 5., 6. und 7.).

(7) Der Datenverkehr von beliebigen Ports der eigenen IP-Adresse 172.25.92.72 zum FTP-Serverport auf der Anlage 172.25.92.74 wird im Transportmodus gesichert. Verwendet wird das Verschlüsselungsverfahren AES und das Authentifizierungsverfahren HMAC-SHA1 (Punkte 4., 5., 6. und 7.).

Eine Prüfung mit START-IPSEC-DB-CHECK liefert in der Logging-Datei eine Ausgabe der SPD und SAD:

```
      S E C U R I T Y   P O L I C Y   D A T A B A S E


outbound policies in ascending order


FTPDATA
  nr of ranges:    1
  ranges set:      own-port
  partner address: IPv4   172.25.92.74
  own address:     IPv4   172.25.92.72
  protocol:        TCP
  partner port:    20
  own port:        0 - 65535
  mode of use:     transport
FTPSERVER
  nr of ranges:    1
```

```
  ranges set:      own-port
  partner address: IPv4   172.25.92.74
  own address:     IPv4   172.25.92.72
  protocol:        TCP
  partner port:    21
  own port:        0 - 65535
  mode of use:     transport

VM4
  nr of ranges:    3
  ranges set:      prot part-port own-port type code
  partner address: IPv4   172.25.92.74
  own address:     IPv4   172.25.92.72
  protocol:        ANY
    ICMP type:       0 - 255
    ICMP code:       0 - 255
  partner port:    0 - 65535
  own port:        0 - 65535
  mode of use:     transport

MYFRIENDS
  nr of ranges:    5
  ranges set:      part-addr own-addr prot part-port own-port type code
  partner address: IPv4   172.25.92.0 - 172.25.92.255
  own address:     ANY
  protocol:        ANY
    ICMP type:       0 - 255
    ICMP code:       0 - 255
  partner port:    0 - 65535
  own port:        0 - 65535
  mode of use:     transport

THEOTHERS
  nr of ranges:    5
  ranges set:      part-addr own-addr prot part-port own-port type code
  partner address: ANY
  own address:     ANY
  protocol:        ANY
    ICMP type:       0 - 255
    ICMP code:       0 - 255
  partner port:    0 - 65535
  own port:        0 - 65535


 inbound policies in ascending order


 FTPDATA
  nr of ranges:    1
```

```
  ranges set:      own-port
  partner address: IPv4   172.25.92.74
  own address:     IPv4   172.25.92.72
  protocol:        TCP
  partner port:    20
  own port:        0 - 65535
  mode of use:     transport

FTPSERVER
  nr of ranges:    1
  ranges set:      own-port
  partner address: IPv4   172.25.92.74
  own address:     IPv4   172.25.92.72
  protocol:        TCP
  partner port:    21
  own port:        0 - 65535
  mode of use:     transport

VM4
  nr of ranges:    3
  ranges set:      prot part-port own-port type code
  partner address: IPv4   172.25.92.74
  own address:     IPv4   172.25.92.72
  protocol:        ANY
    ICMP type:       0 - 255
    ICMP code:       0 - 255
  partner port:    0 - 65535
  own port:        0 - 65535
  mode of use:     transport

MYFRIENDS
  nr of ranges:    5
  ranges set:      part-addr own-addr prot part-port own-port type code
  partner address: IPv4   172.25.92.0 - 172.25.92.255
  own address:     ANY
  protocol:        ANY
    ICMP type:       0 - 255
    ICMP code:       0 - 255
  partner port:    0 - 65535
  own port:        0 - 65535
  mode of use:     transport

THEOTHERS
  nr of ranges:    5
  ranges set:      part-addr own-addr prot part-port own-port type code
  partner address: ANY
  own address:     ANY
  protocol:        ANY
    ICMP type:       0 - 255
```

```
    ICMP code:       0 - 255
  partner port:    0 - 65535
  own port:        0 - 65535
  mode of use:     transport


outbound ESP security associations in ascending order

SA11
  nr of ranges:    1
  ranges set:      own-port
  partner address: IPv4   172.25.92.74
  own address:     IPv4   172.25.92.72
  protocol:        TCP
  partner port:    20
  own port:        0 - 65535
  SPI:             258 00000102
  mode of use:     transport

SA1
  nr of ranges:    1
  ranges set:      own-port
  partner address: IPv4   172.25.92.74
  own address:     IPv4   172.25.92.72
  protocol:        TCP
  partner port:    21
  own port:        0 - 65535
  SPI:             257 00000101
  mode of use:     transport

SA2
  nr of ranges:    2
  ranges set:      part-port own-port type code
  partner address: IPv4   172.25.92.74
  own address:     IPv4   172.25.92.72
  protocol:        ANY
    ICMP type:       0 - 255
    ICMP code:       0 - 255
  partner port:    0 - 65535
  own port:        0 - 65535
  SPI:             300 0000012C
  mode of use:     transport

SA3
  nr of ranges:    4
  ranges set:      part-addr own-addr part-port own-port type code
  partner address: IPv4   172.25.92.0 - 172.25.92.255
  own address:     ANY
  protocol:        ANY
```

```
    ICMP type:       0 - 255
    ICMP code:       0 - 255
  partner port:    0 - 65535
  own port:        0 - 65535
  SPI:             400 00000190
  mode of use:     transport


inbound ESP security associations in ascending order

SA1
  nr of ranges:    0
  own address:     IPv4   172.25.92.72
  SPI:             257 00000101
  partner address: IPv4   172.25.92.74
  protocol:        TCP
  partner port:    21
  own port:        0 - 65535
  mode of use:     transport

SA11
  nr of ranges:    0
  own address:     IPv4   172.25.92.72
  SPI:             258 00000102
  partner address: IPv4   172.25.92.74
  protocol:        TCP
  partner port:    20
  own port:        0 - 65535
  mode of use:     transport

SA2
  nr of ranges:    0
  own address:     IPv4   172.25.92.72
  SPI:             300 0000012C
  partner address: IPv4   172.25.92.74
  protocol:        ANY
    ICMP type:       0 - 255
    ICMP code:       0 - 255
  partner port:    0 - 65535
  own port:        0 - 65535
  mode of use:     transport

SA3
  nr of ranges:    1
  ranges set:      own-addr
  own address:     ANY
  SPI:             400 00000190
  partner address: IPv4   172.25.92.0 - 172.25.92.255
  protocol:        ANY
```

```
    ICMP type:       0 - 255
    ICMP code:       0 - 255
  partner port:    0 - 65535
  own port:        0 - 65535
  mode of use:     transport


No outbound AH security associations!
No inbound AH security associations!
No outbound IPCOMP security associations!
No inbound IPCOMP security associations!
```

6.6.1.2 Automatischer Schlüsselaustausch in einer IPSec-Konfiguration

Die im Abschnitt „Manuell definierte Schlüssel in einer IPSec-Konfiguration“ auf Seite 122 gezeigte Beispielkonfiguration mit manuell vergebenen Schlüsseln und Security Policy Indexes (SPIs) lässt sich äquivalent durch folgende Konfigurationsdatei darstellen.

Die Security Associations werden durch IKE aufgebaut, sobald eine Datenkommunikation erfolgt, die durch die beschriebenen Security Policies erfasst ist. Die Konfigurationsdatei "SYSDAT.IPSEC.*nnn*.RAC2.MANUAL-BSP-DEFAULT" muss ebenso existieren wie die SPMD-Passwortdatei und die PSK-Dateien für die IKE-Partnerinstanzen. Die Kommentare zu obigem Beispiel gelten analog, wobei berücksichtigt werden muss, dass die SPI einzelner Security Associations vom IPSec Kernel "zufällig" vergeben werden.

```
include "SYSDAT.IPSEC.nnn.RAC2.MANUAL-BSP-DEFAULT" ;
#
# VM4
# The system with IP address 172.25.92.74 (VM4)
# prefers IKEv1 and requires a different policy.
# Parameters of the default section are overriden.
#
remote VM4 {
   acceptable_kmp { ikev1; };
   ikev1 {
      my_id ipaddr 172.25.92.72;
      peers_id ipaddr 172.25.92.74;
      peers_ipaddr 172.25.92.74 port 500;
      kmp_enc_alg { 3des_cbc;};
      kmp_hash_alg { sha1; md5;};
      kmp_dh_group { modp1536; modp2048; };
      kmp_auth_method { psk; };
      pre_shared_key "VM4.psk";
   };
#
# Selector and policy definition for the peer VM4.
#
selector VM4-in {
```

```
   direction inbound;
   src 172.25.92.74;
   dst 172.25.92.72;
   upper_layer_protocol "any";
   policy_index VM4;
};
selector VM4-out {
   direction outbound;
   dst 172.25.92.74;
   src 172.25.92.72;
   upper_layer_protocol "any";
   policy_index VM4;
};
#
# All IP pakets must be ESP secured.
#
policy VM4 {
   action auto_ipsec;
   remote_index VM4;
   ipsec_mode transport;
   ipsec_index { ipsec_esp-2; };
   ipsec_level require;
};

ipsec ipsec_esp-2 {
   ipsec_sa_lifetime_time 86400 sec;
   ipsec_sa_lifetime_byte 10 MB;
   sa_index esp_01;
};

sa esp_01 {
   sa_protocol esp;
   esp_enc_alg { aes128_cbc; };
   esp_auth_alg { non_auth; };
};
# Selector and policy definition for FTP from and to VM4.
# This is an example for a application/port based policy.
#
selector FTPSERVER-in {
   direction inbound;
   src 172.25.92.74;
   dst 172.25.92.72 port 21;
   upper_layer_protocol "tcp";
   policy_index FTPSERV;
};
selector FTPSERVER-out {
   direction outbound;
   dst 172.25.92.74 port 21;
```

```
   src 172.25.92.72;
   upper_layer_protocol "tcp";
   policy_index FTPSERV;
};
#
# All TCP pakets must be AH+ESP secured.
#
policy FTPSERV {
   action auto_ipsec;
   remote_index VM4;
   ipsec_mode transport;
   ipsec_index { ftp_server; };
   ipsec_level require;
};

ipsec ftp_server {
   ipsec_sa_lifetime_time 3600 sec;
   ipsec_sa_lifetime_byte 1 MB;
   sa_index{ esp_01; ah_01; };
};

sa ah_01 {
   sa_protocol ah;
   ah_auth_alg { hmac_sha1; };
};
# Selector and policy definition for the FTP data of VM4.
#
selector FTPDATA-in {
   direction inbound;
   src 172.25.92.74;
   dst 172.25.92.72 port 20;
upper_layer_protocol "tcp";
   policy_index FTPDATA;
};
selector FTPDATA-out {
   direction outbound;
   dst 172.25.92.74 port 20;
   src 172.25.92.72;
   upper_layer_protocol "tcp";
   policy_index FTPDATA;
};
#
# All IP pakets must be AH+ESP secured.
#
policy FTPDATA {
   action auto_ipsec;
   remote_index VM4;
   ipsec_mode transport;
```

```
   ipsec_index { ftp_data; };
   ipsec_level require;
};

ipsec ftp_data {
   ipsec_sa_lifetime_time 28800 sec;
   ipsec_sa_lifetime_byte 1 GB;
   sa_index{ esp_01; ah_01; };
};
#
# Selector and policy definition for the default policy DISCARD.
#
selector THEOTHERS-in {
   direction inbound;
   src IP_ANY;
   dst 172.25.92.72/0;
   upper_layer_protocol "any";
   policy_index THEOTHERS;
};
selector THEOTHERS-out {
   direction outbound;
   dst IP_ANY;
   src 172.25.92.72/0;
   upper_layer_protocol "any";
   policy_index THEOTHERS;
};
#
# All IP pakets are discarded unless there is a policy.
#
policy THEOTHERS {
   action discard;
};
```

6.6.1.3 Kommentare zu den Beispielen

SPs und SAs werden in der Datenbasis in einer aufsteigenden Reihenfolge bezüglich ihrer Selektoren abgelegt und auch in dieser Reihenfolge durchsucht.

Die "am genauesten" spezifizierten Sicherheitsrichtlinien werden vor den weniger genauen einsortiert. Das Maß der Genauigkeit ist die Anzahl der Bereichsangaben.

In unserer Beispielkonfiguration wurde für die SP FTPSERVER bei der eigenen Portnummer ANY und bei allen anderen Selektoren Einzelwerte angegeben. Die Anzahl der Bereichsangaben ist daher eins. Bei der SP VM4 wurden nur die eigene IP-Adresse und die Partner-IP-Adresse angegeben. Die Anzahl der Bereichsangaben ist daher drei und die SP VM4 wird daher nach den SPs FTPDATA und FTPSERVER einsortiert. Ist die Anzahl der

Bereichsangaben gleich, wird innerhalb eines Selektors lexikografisch sortiert. Die Partnerportnummer 20 der SP FTPDATA ist kleiner als die Partnerportnummer 21 der SP FTP-SERVER. FTPDATA ist deshalb vor FTPSERVER einsortiert.

Da für den ein- und ausgehenden Datenverkehr die gleichen Angaben gemacht wurden, sieht die SPD auch gleich aus.

Aus den SPs werden SAs abgeleitet.

Für die SP THEOTHERS gibt es logischerweise in der SAD weder für eingehende noch ausgehende Pakete Einträge, da für zu verwerfende Pakete keine SA gebraucht wird.

Bearbeitung von Datenpaketen

Für jedes ausgehende Datenpaket wird in der SPD nach einer passenden Sicherheitsrichtlinie gesucht. Wird keine gefunden, so wird keine Verarbeitung durch IPSec vorgenommen. Wird eine gefunden, so wird entsprechend der Angabe im Operanden MODUS verfahren:

- BYPASS: es erfolgt keine Verarbeitung durch IPSec.
- DISCARD: das Paket wird verworfen
- TRANSPORT/TUNNEL: die zugehörigen SAs werden gesucht und das Datenpaket wird entsprechend verschlüsselt und/oder signiert.

Das Vorgehen bei eingehenden Datenpaketen ist analog. Zunächst wird gesucht, ob es zutreffende SAs gibt und das Datenpaket entschlüsselt und/oder verifiziert. Abschließend wird nach einer Sicherheitsrichtlinie gesucht. Wird eine gefunden, werden die verwendeten Sicherheitsprotokolle mit den geforderten verglichen. Bei Übereinstimmung wird das Paket weiterverarbeitet, ansonsten wird es verworfen.

Ausgehend von der oben beschriebenen Konfiguration wird die Bearbeitung von Datenpaketen an Beispielen demonstriert. Auf den angesprochenen Partnerrechnern muss natürlich eine entsprechende IPSec-Konfiguration vorhanden sein.

1. Ein TCP-Paket soll von einer Anwendung mit der Portnummer 3069 auf dem Rechner mit der IP-Adresse 172.25.92.72 an eine Anwendung mit der Portnummer 3069 auf dem Rechner mit der IP-Adresse 172.25.92.74 geschickt werden:

   IPSec bildet die Suchselektoren

   ```
   partner address: IPv4   172.25.92.74
   own address:     IPv4   172.25.92.72
   protocol:        TCP
   partner port:    3069
   own port:        3069
   ```

   und sucht in der SPD für ausgehende Pakete. Gefunden wird die SP mit dem Namen VM4. Die SP gilt für alle Protokolle, die gesuchte Portnummer 3069 liegt im Bereich 0 - 65535. Diese fordert eine ESP-SA im Transport-Modus. Mit den für die SP-Suche

gebildeten Selektoren wird in der SAD für ausgehende Datenpakete gesucht. Die SA mit dem Namen SA2 wird gefunden. Mit den Angaben in SA2 wird das Datenpaket verschlüsselt, mit der SPI 300 versehen und anschließend verschickt.

Kommt vom Rechner 172.25.92.74 die Antwort als durch ESP abgesichertes Paket zurück, wird mit den Selektoren in der SAD für eingehende ESP-Pakete gesucht:

```
own address:     IPv4   172.25.92.74
SPI:             300
```

SA2 wird gefunden. Mit den Angaben in SA2 wird das Datenpaket entschlüsselt. Nun werden für die Suche der SP die Selektoren gebildet:

```
partner address: IPv4   172.25.92.74
own address:     IPv4   172.25.92.74
protocol:        TCP
partner port:    3069
own port:        3069
```

Die SP VM4 wird gefunden und dort ist eine ESP-SA gefordert. Das Paket wird daher akzeptiert.

2. Ein TCP-Paket soll von einer Anwendung mit der Portnummer 3069 auf dem Rechner mit der IP-Adresse 172.25.92.72 an eine Anwendung mit der Portnummer 3069 auf dem Rechner mit der IP-Adresse 172.25.92.75 geschickt werden:

   IPSec bildet die Suchselektoren

   ```
   partner address: IPv4   172.25.92.75
   own address:     IPv4   172.25.92.72
   protocol:        TCP
   partner port:    3069
   own port:        3069
   ```

   und sucht in der SPD für ausgehende Pakete. Gefunden wird die SP mit dem Namen MYFRIENDS. Das weitere Vorgehen ist wie im 1. Beispiel.

3. Ein TCP-Paket soll von einer Anwendung mit der Portnummer 3069 auf dem Rechner mit der IP-Adresse 172.25.92.72 an eine Anwendung mit der Portnummer 3069 auf dem Rechner mit der IP-Adresse 172.25.123.137 geschickt werden:

   IPSec bildet die Suchselektoren

   ```
   partner address: IPv4   172.25.123.137
   own address:     IPv4   172.25.92.72
   protocol:        TCP
   partner port:    3069
   own port:        3069
   ```

   und sucht in der SPD für ausgehende Pakete. Gefunden wird die SP mit dem Namen THEOTHERS. Diese führt zum Verwurf des Paketes.

4. Auf dem Rechner 172.25.92.72 wird ein durch ESP abgesichertes Paket empfangen. Mit den Selektoren

   ```
   own address:     IPv4   172.25.92.72
   SPI:             2771
   ```

   wird in der SAD für eingehende ESP-Pakete gesucht. Eine SA wird nicht gefunden. Daher wird das Paket verworfen.

6.6.2 VPN Tunnel mit IPSec

Das folgende Beispiel veranschaulicht die Verwendung eines IPSec-Tunnels.

Das BS2000-System (192.168.11.87) und ein Router (192.168.11.13) sind die Endpunkte des Tunnels. Das Subnetz 172.25.226.128 - 172.25.226.255 (in der Notation 172.25.226.128/25) wird über den Router erreicht, der selbst ein Board in diesem Subnetz hat.

Es sollen folgende Sicherheitsrichtlinien gelten:

1. Sämtlicher Datentransfer zwischen der eigenen IP-Adresse 192.168.11.87 und dem Subnetz 172.25.226.128/25 soll zwischen den IP-Adressen 192.168.11.87 und 192.168.11.13 verschlüsselt werden.
2. Es soll der Tunnelmodus verwendet werden.
3. Es werden Verschlüsselungsverfahren AES256, AES192 und 3DES über eine Auswahlliste angeboten bzw. akzeptiert.
4. Die Schlüssel werden automatisch ausgetauscht. Für die Authentifizierung der Key Management-Instanzen, den Aufbau der Security-Associations und den Schlüsselaustausch soll IKEv1 verwendet werden.

Die Konfigurationsdateien ROUTER8.PSK, die SPMD-Passwort-Datei und die Datei für die Default Section SYSDAT.IPSEC.*nnn*.RAC2.MANUAL-BSP-DEFAULT müssen vorhanden sein.

Mit folgender Konfigurationsdatei lässt sich dann die beschriebene Konfiguration darstellen:

SYSDAT.IPSEC.014.RAC2

```
include "SYSDAT.IPSEC.014.RAC2.MANUAL-BSP-DEFAULT";

#
# IPSec tunnel tunr8: tunnel to router8
# router8 speaks IKEv1 and the preshared key is stored
# in the file "ROUTER8.psk".
```

```
# Parameters of the default section are overridden.
#

remote tunr8 {
   acceptable_kmp { ikev1; };
   ikev1 {
      my_id ipaddr 192.168.11.87;
      peers_ipaddr 192.168.11.13 port 500;
      kmp_enc_alg { 3des_cbc;};
      kmp_hash_alg { sha1; };
      kmp_dh_group { modp1024; };
      kmp_auth_method { psk; };
      pre_shared_key "ROUTER8.psk";
   };
};
_____________________________________________________________________(1)


#
# Selector and policy definition for the subnet 172.25.226.128/25
#

selector tunr8-in {
   direction inbound;
   src 172.25.226.128/25;
   dst 192.168.11.87;
   upper_layer_protocol "any";
   policy_index tunr8;
};
selector tunr8-out {
   direction outbound;
   dst 172.25.226.128/25;
   src 192.168.11.87;
   upper_layer_protocol "any";
   policy_index tunr8;
};
_____________________________________________________________________(2)

policy tunr8 {
   action auto_ipsec;
   remote_index tunr8;
   ipsec_mode tunnel;
   ipsec_index { ipsec_esp_multi; };
   ipsec_level require;
   peers_sa_ipaddr 192.168.11.13;
   my_sa_ipaddr 192.168.11.87;
};
_____________________________________________________________________(3)
```

```
ipsec ipsec_esp_multi {
   ipsec_sa_lifetime_time 28800 sec;
   sa_index esp_multi;
};

sa esp_multi {
   sa_protocol esp;
   esp_enc_alg { aes256_cbc; aes192_cbc; 3des_cbc; };
   esp_auth_alg { non_auth; };
};
_____________________________________________________________________(4)
```

Erläuterungen:

(1) In der remote-Anweisung stehen die kmp-Parameter für die gegenseitige Authentifizierung der IKE-Instanzen. Es wird IKEv1 verwendet (Punkt 4.).

(2) Jeglicher Datentransfer (d.h. über alle Protokolle) zwischen dem System 192.168.11.87 und einem System im Subnetz 172.25.226.128/25 soll den Tunnel zwischen 192.168.11.87 und 192.168.11.13 nutzen (Punkt 1.).

(3) Die Policy zwischen den Systemen 192.168.11.87 und 192.168.11.13 verwendet den Tunnelmodus (Punkt 2.).

(4) Als Verschlüsselungsverfahren werden AES256, AES192 und 3DES angeboten bzw. akzeptiert (Punkt 3.).

6.6.3 IP Payload Compression Protocol in IPSec

Für das nachstehende Beispiel gelten folgende Sicherheitsrichtlinien:

1. Der Datenverkehr zwischen den Systemen mit den IP-Adressen 172.25.226.220 und 172.25.226.221 wird durch eine IPSec ESP SA im Transportmodus gesichert.
2. Als Verschlüsselungsverfahren werden AES und 3DES, als Authentifizierungsverfahren werden HMAC-SHA1 und HMAC-MD5 vorgeschlagen, bzw. akzeptiert.
3. Die Komprimierung des IP Payload mit dem Komprimierungsalgorithmus DEFLATE wird zugelassen.
4. Die gegenseitige Authentifizierung der Key Management Instanzen, der Aufbau der ESP SA und der Schlüsselaustausch müssen über das Internet-Key-Exchange-Protokoll (IKEv1) erfolgen.

Die Defaultwerte stehen in der Datei SYSDAT.IPSEC.*nnn*.RAC2.MANUAL-BSP-DEFAULT, welche per include-Anweisung eingelesen wird.

Die Konfigurationsdateien VM1.PSK und SYSDAT.IPSEC.*nnn*.RAC2 sehen wie folgt aus:

Datei VM1.PSK

```
GEHEIMNIS
```

Datei SYSDAT.IPSEC.*nnn*.RAC2

```
include "SYSDAT.IPSEC.nnn.RAC2.MANUAL-BSP-DEFAULT";

#
# my own configuration
#
remote VM1_IKE1 {
   acceptable_kmp { ikev1; };
   ikev1 {
      my_id ipaddr 172.25.226.220;
      peers_ipaddr 172.25.226.221 port 500;
      kmp_enc_alg { 3des_cbc;};
      kmp_hash_alg { sha1; md5;};
      kmp_dh_group { modp1024; };
      kmp_auth_method { psk; };
      pre_shared_key "VM1.psk";
   };
};
_____________________________________________________________________(1)

selector VM1_1 {
   direction inbound;
   src 172.25.226.221;
   dst 172.25.226.220;
   upper_layer_protocol "any";
   policy_index VM1;
};

selector VM1_2 {
   direction outbound;
   dst 172.25.226.221;
   src 172.25.226.220;
   upper_layer_protocol "any";
   policy_index VM1;
};
_____________________________________________________________________(2)

policy VM1 {
   action auto_ipsec;
   remote_index VM1_IKE1;
   ipsec_mode transport;
   ipsec_index { ipsec_IPCOMP_def_ESP_md5-sha1.aes-3des; };
   ipsec_level require;
};
```

```
_____________________________________________________________________(3)
ipsec ipsec_IPCOMP_def_ESP_md5-sha1.aes-3des {
   ipsec_sa_lifetime_time 28800 sec;
   sa_index { IPCOMP_def; ESP_md5-sha1.aes-3des; };
};

sa ESP_md5-sha1.aes-3des {
   sa_protocol esp;
   esp_auth_alg { hmac_md5; hmac_sha1; };
   esp_enc_alg { aes128_cbc; 3des_cbc; };
};
_____________________________________________________________________(4)
sa IPCOMP_def {
   sa_protocol ipcomp;
   ipcomp_alg { deflate; };
};
_____________________________________________________________________(5)
```

Erläuterungen:

(1) In der remote-Anweisung stehen die kmp-Parameter für die gegenseitige Authentifizierung der IKE-Instanzen. Es wird IKEv1 verwendet (Punkt 4.).

(2) Der Datentransfer zwischen den Systemen 172.25.226.220 und 172.25.226.221 wird durch eine IPSec ESP SA im Transportmodus gesichert (Punkt 1.).

(3) Die Policy verwendet IKEv1 (Punkt 4.).

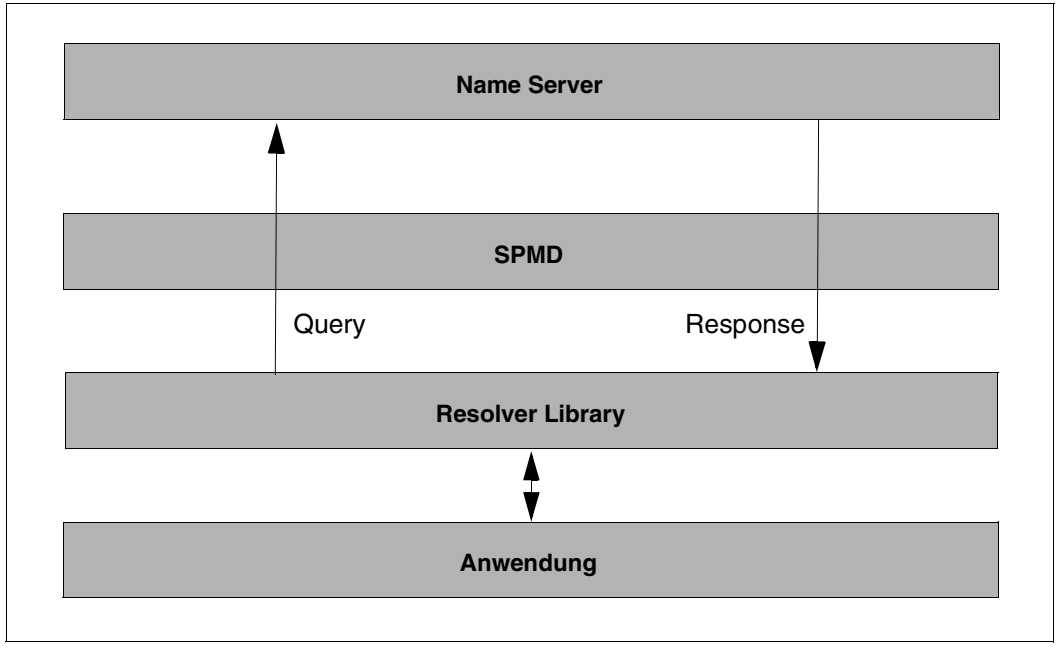
(4) Als Verschlüsselungsverfahren werden AES und 3DES und als Authentifizerungsverfahren werden HMAC-SHA1 und HMAC-MD5 vorgeschlagen bzw. akzeptiert (Punkt 2.).

(5) Für IP Payload wird DEFLATE als Komprimierungsalgorithmus zugelassen (Punkt 3.).

6.6.4 Unterstützung des Domain Name System (DNS)

DNS-Unterstützung bedeutet, dass FullyQualified Domain Names (FQDN) anstelle von IPv4- bzw. IPv6-Adressen verwendet werden.

Zur Unterstützung des Domain Name System (DNS) klemmt sich der SPMD als Proxy zwischen die Anwendungen und den Name Server:



Für FQDNs, die in der Konfigurationsdatei angegeben sind, setzt SPMD Queries ab. Ankommende Queries merkt sich der SPMD und leitet sie an den Name Server weiter. Der SPMD analysiert die Response des Name Server und verwendet diese zur Auflösung der in der Konfigurationsdatei angegebenen Namen. Dem TPR-Subsystem IPSEC werden die Security Policys mit IPv4- bzw. IPv6-Adressen bekannt gemacht.

Das folgende Beispiel zeigt, was zu tun ist, um IPSec mit FQDN nutzen zu können. Dabei wird von einer Client-Server-Situation ausgegangen. Der Client ist ein Rechner, der keine feste IP-Adresse hat. Er wird daher in der Konfigurationsdatei über seinen Namen spezifiziert. Die feste IP-Adresse des Servers hingegen ist den Clients bekannt.

- Ändern Sie den Name Server-Eintrag in der Resolver-Konfigurationsdatei auf:

```
nameserver 127.0.0.1
```

► Stoppen Sie anschließend den LWRESD und starten Sie ihn neu, damit die Änderung wirksam wird:

/STOP-LWRESD
/START-LWRESD

Beispielkonfigurationsdatei SYSDAT.IPSEC.014.RAC2

```
#
# include of the default configuration file with the default parameter
#
include SYSDAT.IPSEC.014.RAC2.MANUAL-BSP-DEFAULT ;
# resolver info

resolver
{
# spmd acts as a resolver
   resolver on;
# this is the DNS server
   nameserver {172.25.92.122  port 53;};
# spmd listens to dns queries on loopback
   dns_query {127.0.0.1 port 53;};
};

#
# VM2
# The client system VM2 must use IKEv2
# Parameters of the default section are overriden.
#

remote VM2 {
   acceptable_kmp { ikev2; };
   ikev2 {
      my_id ipaddr 172.25.92.74;
      peers_id fqdn "BCABLZO2.SO13.MCH.FTS.NET";
      kmp_enc_alg { 3des_cbc;};
      kmp_hash_alg { hmac_md5;};
      kmp_dh_group { modp1536; };
      kmp_prf_alg { hmac_md5;  };
      kmp_auth_method { psk; };
      pre_shared_key "VM2.psk";
   };
};

#
# Selector and policy definition for the peer VM4.
# All IP pakets must be ESP secured.
#

selector VM2-out {
```

```
   direction outbound;
   src 172.25.92.74;
   dst "BCABLZO2.SO13.MCH.FTS.NET";
   upper_layer_protocol "tcp";
   policy_index VM2;
};

selector VM2-in {
   direction inbound;
   src "BCABLZO2.SO13.MCH.FTS.NET";
   dst 172.25.92.74;
   upper_layer_protocol "tcp";
   policy_index VM2;
};

policy VM2 {
   action auto_ipsec;
   remote_index VM2;
   ipsec_mode transport;
   ipsec_index { ipsec_esp; };
   ipsec_level require;
   peers_sa_ipaddr IP_RW;
   my_sa_ipaddr 172.25.92.74;
};

ipsec ipsec_esp {
   ipsec_sa_lifetime_time 28800 sec;
   sa_index esp_01;
};

sa esp_01 {
   sa_protocol esp;
   esp_enc_alg { aes128_cbc;  };
   esp_auth_alg { hmac_sha1;  };
};
```

6.6.5 Unterstützung von Network Address Translation (NAT)

Damit IPSec-geschützte Daten NAT-Geräte passieren können, ist NAT-Traversal nötig.

Das folgende Beispiel zeigt, wie IPSec NAT-Traversal unterstützt. Dabei wird von einer Client-Server-Situation ausgegangen. Der Client ist ein Rechner, der keine feste IP-Adresse hat. Er wird daher in der Konfigurationsdatei über seinen Namen spezifiziert. Die feste IP-Adresse des Servers hingegen ist den Clients bekannt.

- Ändern Sie den Name Server-Eintrag in der Resolver-Konfigurationsdatei auf:

  ```
  nameserver 127.0.0.1
  ```

- Stoppen Sie anschließend den LWRESD und starten Sie ihn neu, damit die Änderung wirksam wird:

  /STOP-LWRESD
  /START-LWRESD

Beispielkonfigurationsdatei SYSDAT.IPSEC.014.RAC2

```
#
# include of the default configuration file with the default parameter
#
include SYSDAT.IPSEC.014.RAC2.MANUAL-BSP-DEFAULT ;

# interface info

interface
{
   ike {
      MY_IP port 500;
# The support of NAT-Traversal requires an additional instruction
# with port 4500;
      MY_IP port 4500;
   };
};

# resolver info
resolver
{
# spmd acts as a resolver
   resolver on;
# this is the DNS server
   nameserver {172.25.92.122  port 53;};
# spmd listens to dns queries on loopback
   dns_query {127.0.0.1 port 53;};
};

# VM2
```

```
# The client system VM2 must use IKEv2

remote VM2 {
   acceptable_kmp { ikev2; };
   ikev2 {
      my_id ipaddr 10.0.0.4;
      peers_id fqdn "BCABLZ02.SO13.MCH.FTS.NET";
      kmp_enc_alg { 3des_cbc;};
      kmp_hash_alg { hmac_md5;};
      kmp_dh_group { modp1536; };
      kmp_prf_alg { hmac_md5;  };
      kmp_auth_method { psk; };
      pre_shared_key "VM2.psk";
   };
};

#
# Selector and policy definition for the peer VM2.
# All IP pakets must be ESP secured.
#

selector VM2-out {
   direction outbound;
   src 10.0.0.4;
   dst "BCABLZ02.SO13.MCH.FTS.NET";
   upper_layer_protocol "tcp";
   policy_index VM2;
};

selector VM2-in {
   direction inbound;
   src "BCABLZ02.SO13.MCH.FTS.NET";
   dst 10.0.0.4;
   upper_layer_protocol "tcp";
   policy_index VM2;
};

policy VM2 {
   action auto_ipsec;
   remote_index VM2;
   ipsec_mode transport;
   ipsec_index { ipsec_esp; };
   ipsec_level require;
   peers_sa_ipaddr IP_RW;
   my_sa_ipaddr 10.0.0.4;
};

ipsec ipsec_esp {
   ipsec_sa_lifetime_time 28800 sec;
   sa_index esp_01;
```

```
};

sa esp_01 {
   sa_protocol esp;
   esp_enc_alg { aes128_cbc;  };
   esp_auth_alg { hmac_sha1;  };
};
```

7 IPSec-Meldungen

Die Meldungen von IPsec sind durch die Meldungsklasse YIS bestimmt, d.h alle Meldungen von IPSec beginnen mit YIS und sind von einer vierstelligen Dezimalzahl mit führenden Nullen gefolgt. Auf die Erstellung der kompletten Meldungsliste wird an dieser Stelle verzichtet.

Einzelne Meldungen, deren Nummer bekannt sind, können mit dem Kommando

HELP-MSG-INFORMATION MSG-IDENTIFICATION=YIS*nnnn*

angezeigt werden.

Eine komplette Liste aller Meldungen der Meldungsklasse YIS kann mit Hilfe des Programms MSGMAKER erstellt werden, die dann den aktuellen Inhalt der IPSec Meldungsdatei

SYSMES.IPSEC.*nnn*

widerspiegelt.

Das Programm MSGMAKER wird durch das Kommando

/START-MSGMAKER

aufgerufen. Die Erzeugung der Liste der IPSec Meldungen ergibt sich einfach aus der Programmführung von MSGMAKER.

8 Anhang

8.1 Position des IP Payload Compression Protocol (IPCOMP)

Das IP Payload Compression Protocol steht immer unmittelbar vor dem Transportprotokoll, das heißt vor dem ursprünglichen Payload eines IP Segments. IPCOMP ermöglicht es, eine eventuell auf Layer 2 erwünschte Datenkomprimierung auf Layer 3 zu verlagern. Ohne diese Maßnahme würde wegen einer Verschlüsselung durch IPSec keine Komprimierung erfolgen.

Da IPCOMP keine Transformation darstellt, die der Sicherung von Datenkommunikation dient, wird in den folgenden Kapiteln auf eine Darstellung verzichtet.

8.2 Position der Sicherheitsprotokolle

Position des AH relativ zum IPv4-Header

Im IPv4-Paket wird der Authentication Header wie folgt platziert:

- unmittelbar hinter dem Original-IP-Header und
- vor jedem Protokoll-Header (TCP, UDP, ICMP etc.) einer höheren Schicht des IP-Stack und
- vor jedem anderen IPSec-Header

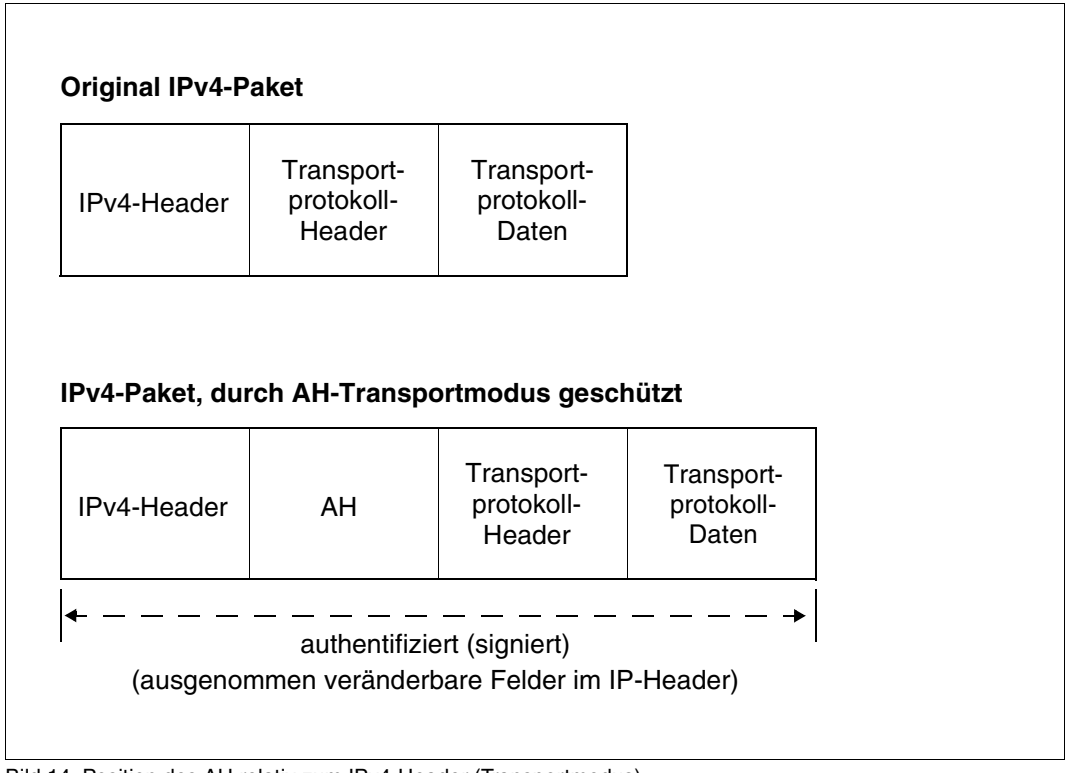


Bild 14: Position des AH relativ zum IPv4-Header (Transportmodus)

Position des AH relativ zum IPv6-Header

Im IPv6-Paket wird der AH hinter den Erweiterungsheadern für Hop-by-Hop, Routing und Fragmentierung eingefügt.

Für die Position des AH relativ zu dem oder den Destination Options-Header(n) gilt:

- Falls der/die Destination Options-Header von Security Gateways (Routern) verarbeitet werden soll(en), die im Destinations-Feld des IPv6-Headers spezifiziert sind, muss der AH nach dem/den Destination Options-Erweiterungsheader(n) platziert werden.
- Falls der Destination Options-Header erst vom Ziel-Host verarbeitet werden soll, muss der AH vor dem/den Destination Options-Header(n) eingefügt werden.

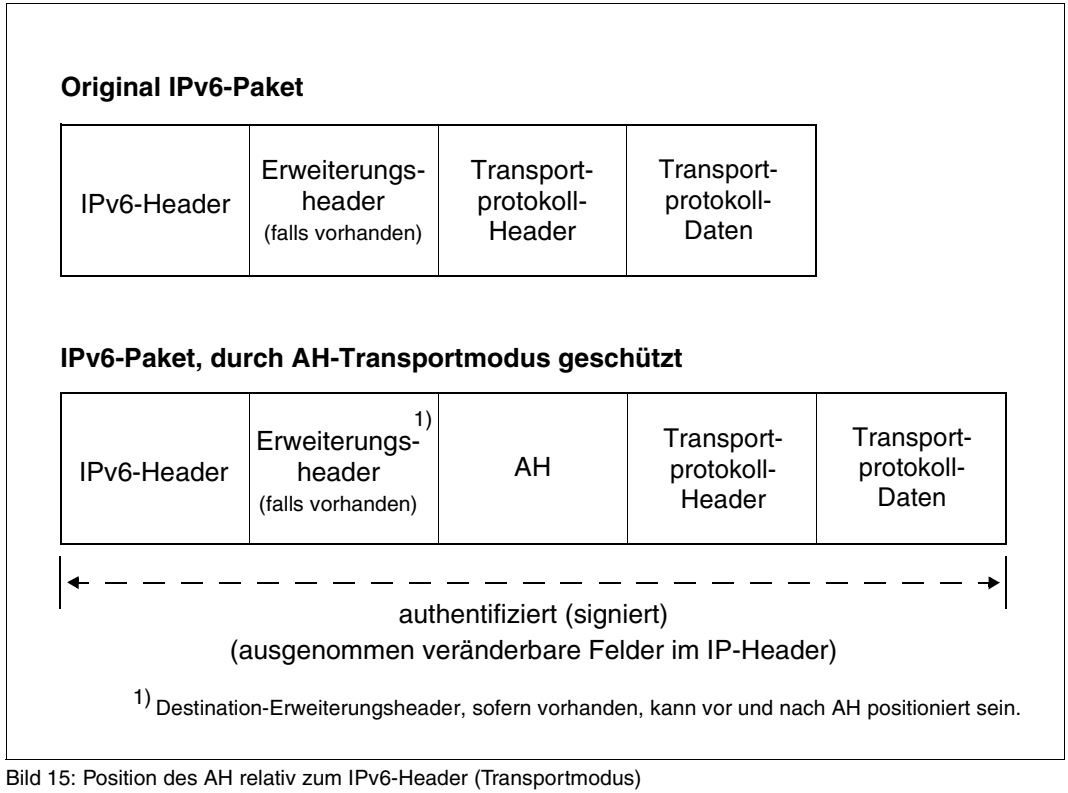


Bild 15: Position des AH relativ zum IPv6-Header (Transportmodus)

Position des AH relativ zum IPv4-Header

Im Tunnelmodus wird der AH unmittelbar vor dem ursprünglichen IPv4-Header platziert und ein neuer IPv4-Header wird dem AH vorangestellt.

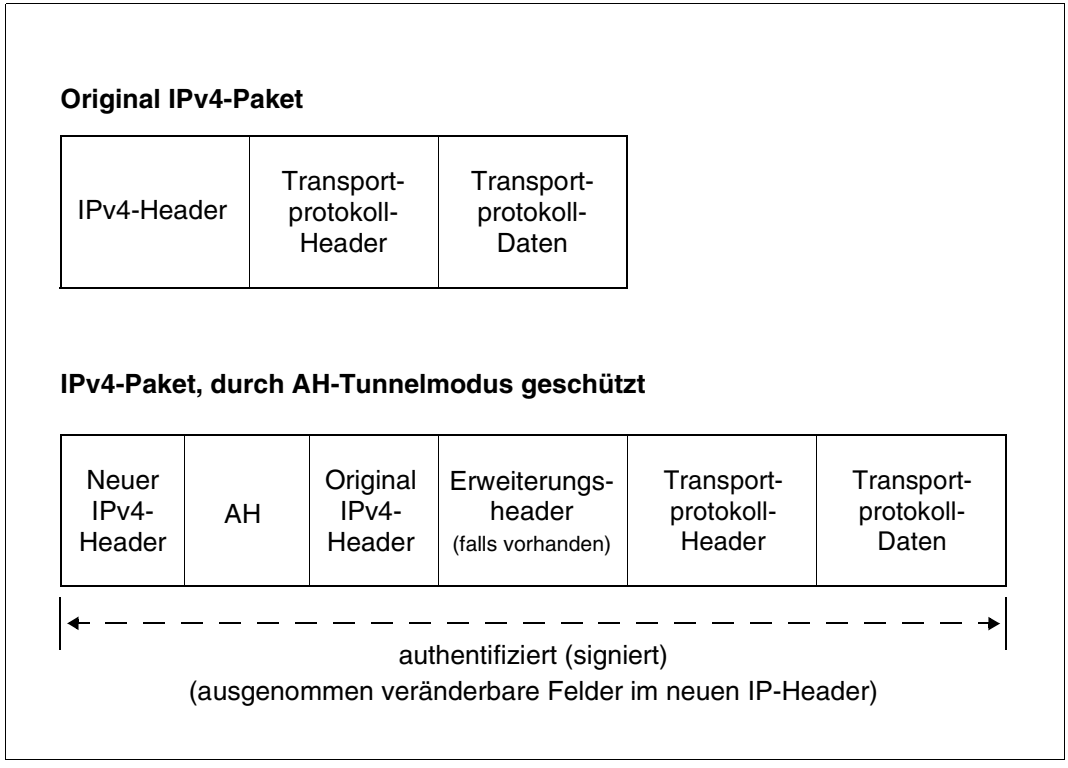


Bild 16: Position des AH relativ zum IPv4-Header (Tunnelmodus)

Position des AH relativ zum IPv6-Header

Im Tunnelmodus wird der AH unmittelbar vor dem ursprünglichen IPv6-Header platziert und dem AH wird ein neuer IPv6-Header plus eventuell vorhandene Erweiterungsheader vorangestellt.

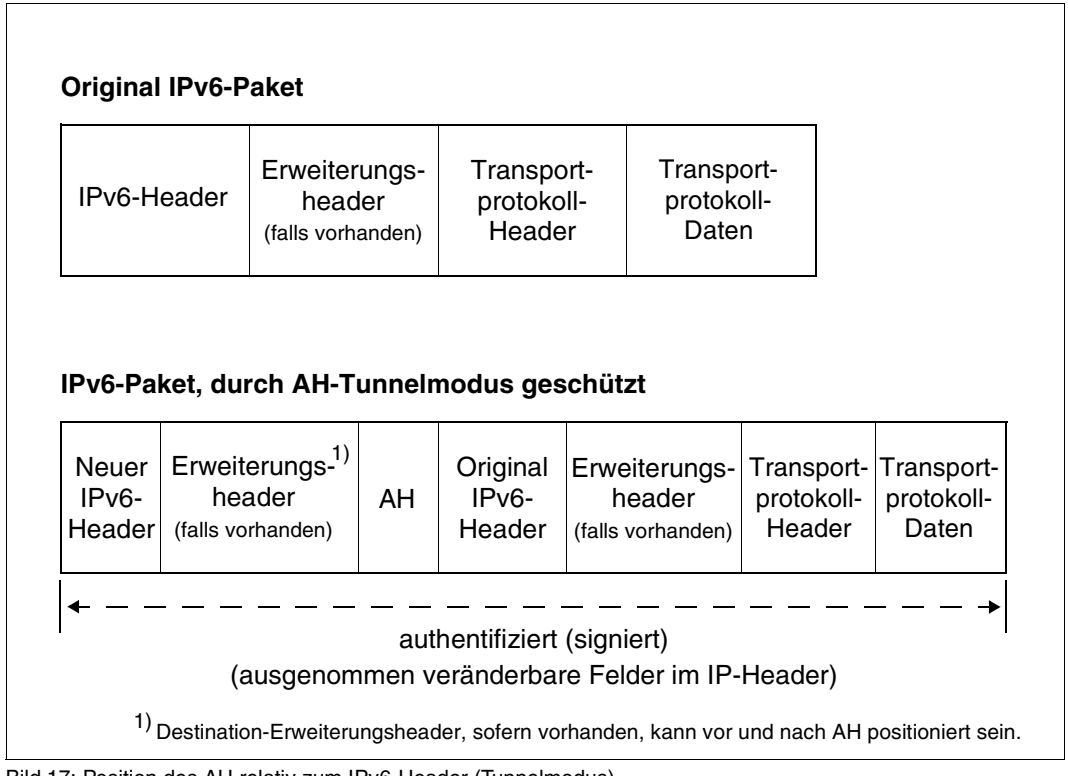


Bild 17: Position des AH relativ zum IPv6-Header (Tunnelmodus)

Position des ESP-Headers relativ zum IPv4-Header

Im IPv4-Header wird der ESP-Header unmittelbar hinter dem IPv4-Header (samt zugehöriger Optionen) und vor dem Protokoll-Header einer höheren Schicht eingefügt:

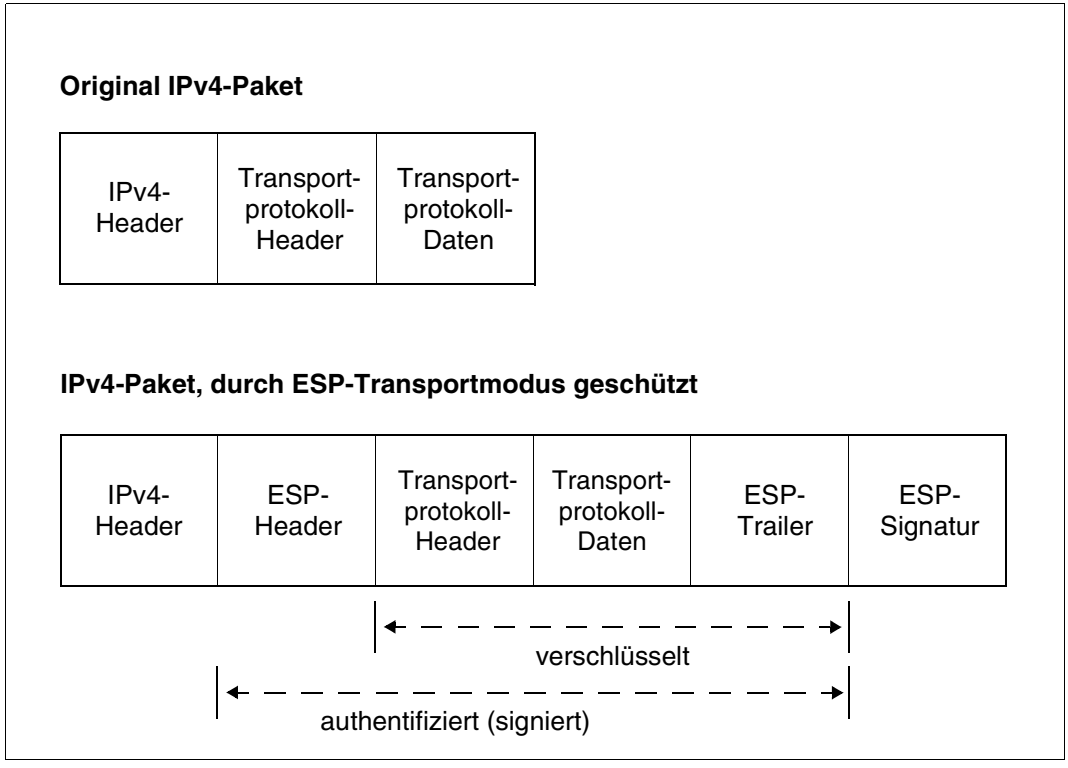


Bild 18: Position des ESP-Headers relativ zum IPv4-Header (Transportmodus)

Position des ESP-Headers relativ zum IPv6-Header

Im IPv6-Protokoll wird der ESP-Header hinter den Erweiterungsheadern für Hop-by-Hop, Routing und Fragmentierung eingefügt.

Für die Position des ESP-Headers relativ zu dem oder den Destination Options Header(n) gilt:

- Falls der/die Destination Options-Header von Systemen (Routern) verarbeitet werden soll(en), die im Destinations-Feld des IPv6-Headers spezifiziert sind, muss der ESP-Header nach dem/den Destination Options-Erweiterungsheader(n) platziert werden.
- Falls der Destination Options-Header erst vom Ziel-Host verarbeitet werden soll, kann der ESP-Header auch nach dem/den Destination Options-Header(n) eingefügt werden.

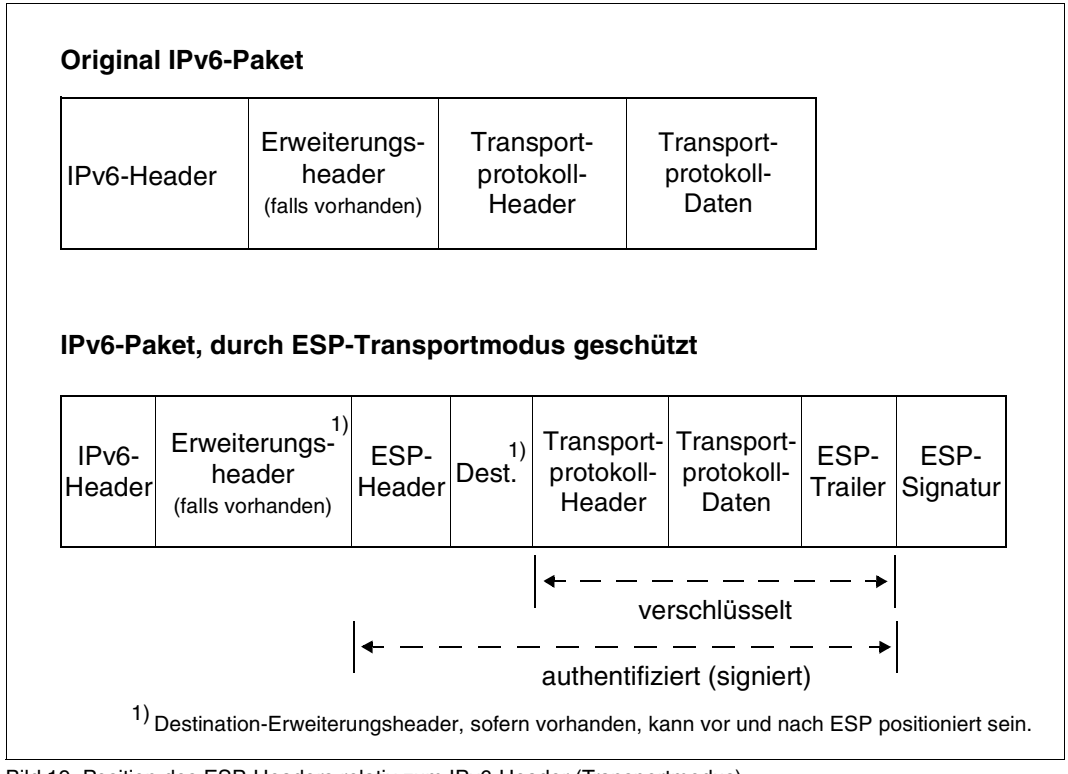


Bild 19: Position des ESP-Headers relativ zum IPv6-Header (Transportmodus)

Position des ESP-Headers relativ zum IPv4-Header

Im Tunnelmodus wird ESP vor dem ursprünglichen IPv4-Header platziert und der neue IPv4-Header wird dem ESP-Header vorangestellt.

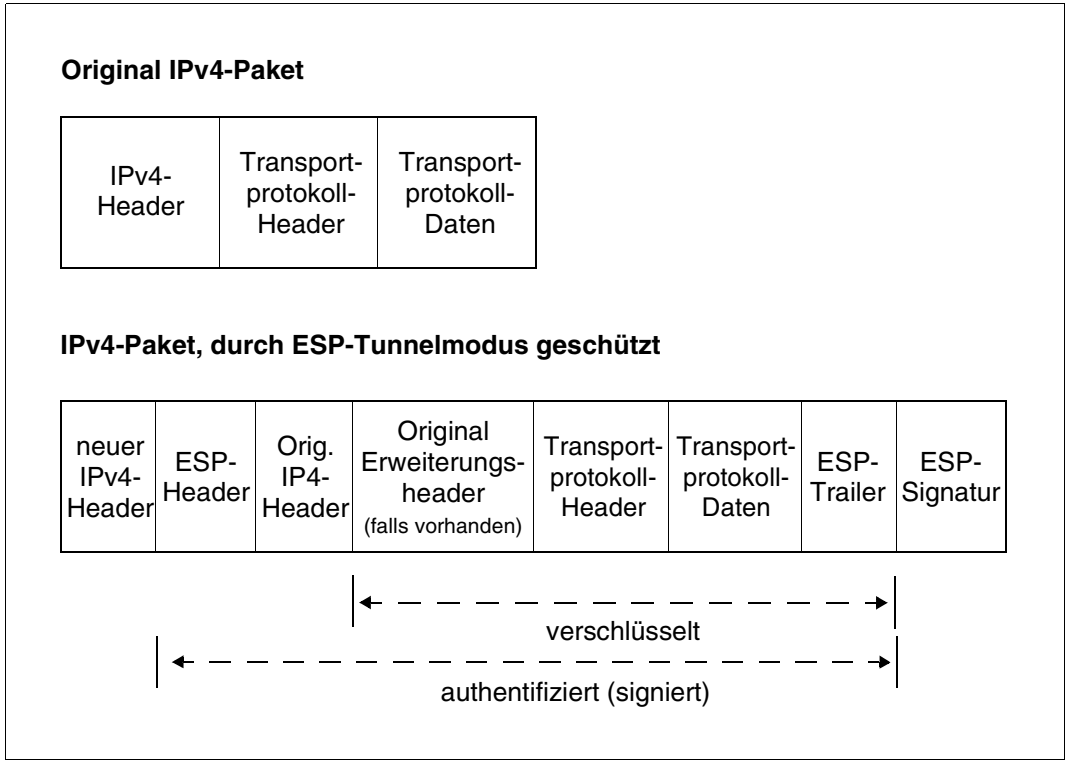


Bild 20: Position des ESP-Headers relativ zum IPv4-Header (Tunnelmodus)

Position des ESP-Headers relativ zum IPv6-Header

Im Tunnelmodus wird ESP vor dem ursprünglichen IPv6-Header platziert und der neue IPv6-Header plus eventuell vorhandene Erweiterungsheader werden ESP vorangestellt.

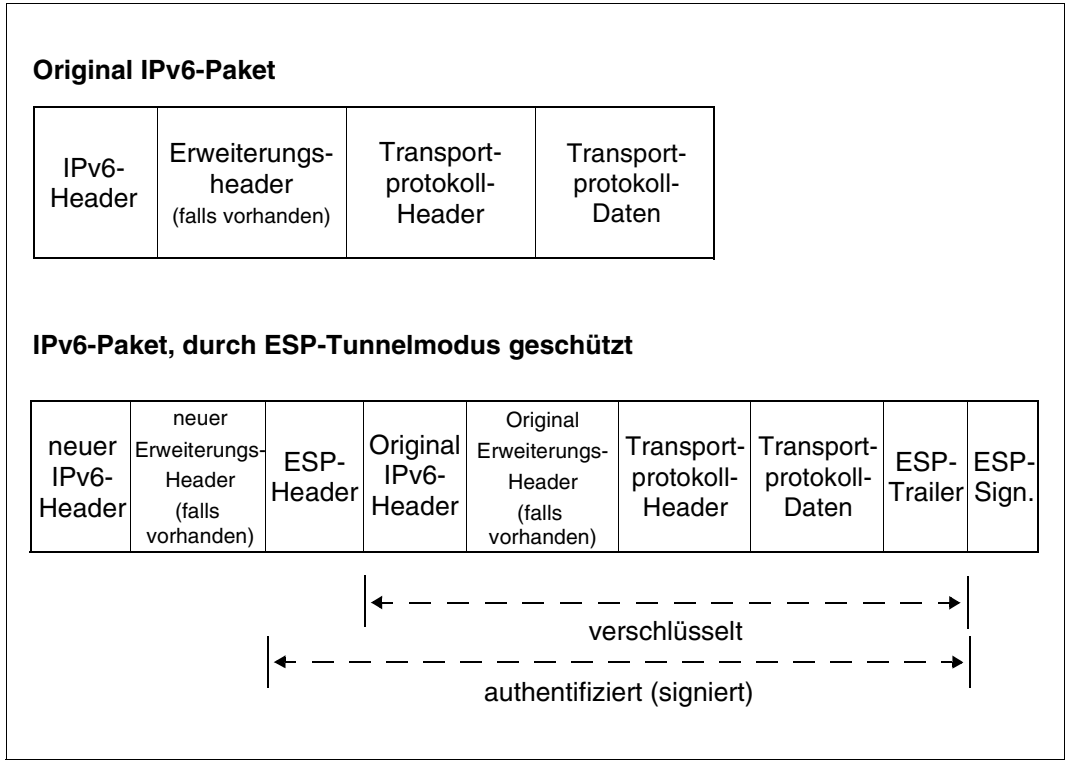


Bild 21: Position des ESP-Headers relativ zum IPv6-Header (Tunnelmodus)

8.3 Sicherheitskonzepte auf der Basis von Security Associations

Dieser Abschnitt beschreibt

- Sicherheitskonzepte, die IPSec bei der Verwendung von ESP- und AH-Headern in den Transfermodi Transport- und Tunnelmodus unterstützt.
- Kombinationsmöglichkeiten von AH- und ESP-Headern in IP-Paketen.
- Übersicht über die Bandbreite an Sicherheitspolitiken, die sich auf der Basis der Protokolle AH und ESP sowie der Transfermodi Transportmodus und Tunnelmodus realisieren lassen.
- Beispiele von Anwenderszenarien.

8.3.1 Sicherheitskonzepte auf der Basis von Transportmodus- und Tunnelmodus-SAs

Sicherheitskonzepte auf der Basis von Transportmodus-SAs

Nachfolgend sind die Sicherheitskonzepte skizziert, die durch die Verwendung von AH- und ESP-Headern im Transportmodus realisierbar sind.

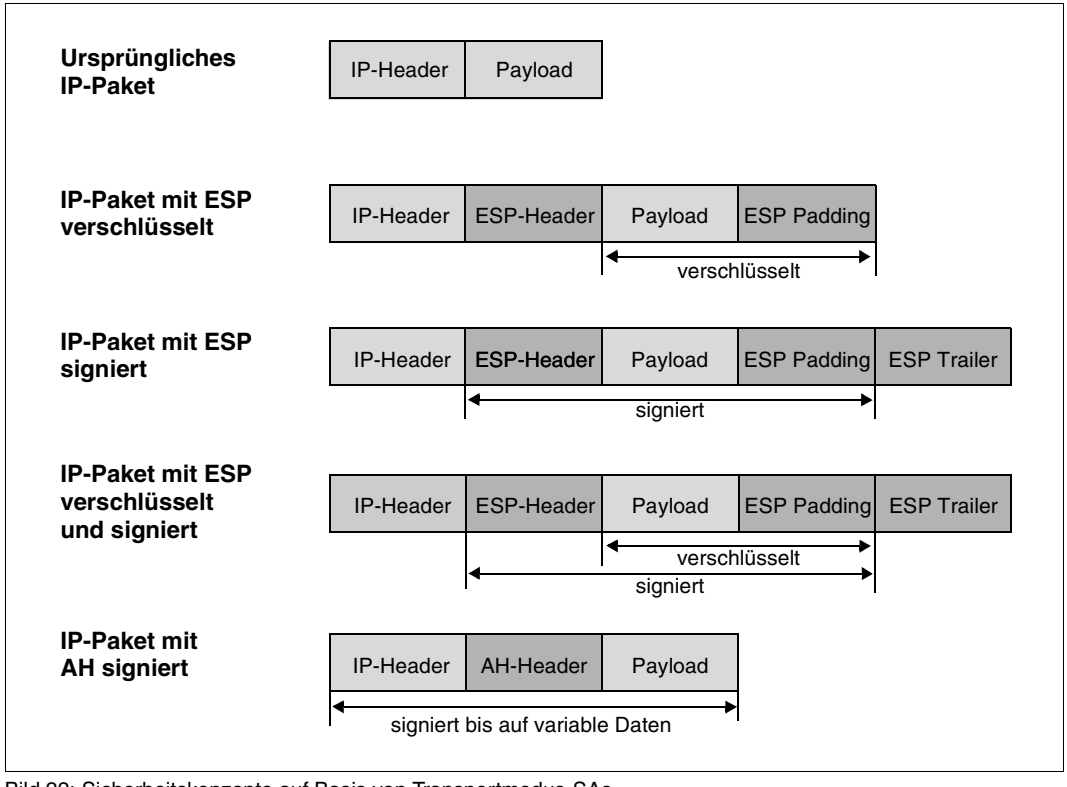


Bild 22: Sicherheitskonzepte auf Basis von Transportmodus-SAs

Sicherheitskonzepte auf der Basis von Tunnelmodus-SAs

Nachfolgend sind die Sicherheitskonzepte skizziert, die durch die Verwendung von AH- und ESP-Headern im Tunnelmodus realisierbar sind.

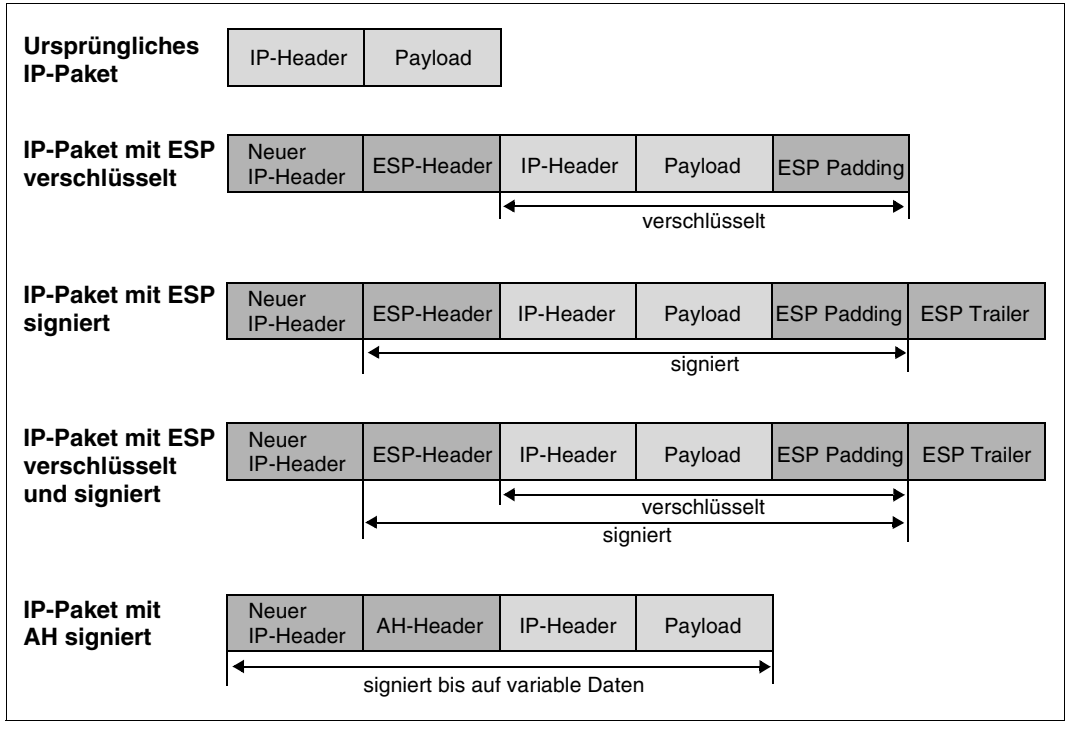


Bild 23: Sicherheitskonzepte auf Basis von Tunnelmodus-SA

8.3.2 Kombination von AH und ESP

Nachfolgend sind die von IPSec unterstützten Kombinationsmöglichkeiten von Authentication Header und Encapsulating Security Payload Header für eingehenden und ausgehenden IP-Verkehr dargestellt.

AH- und ESP-Kombinationsmöglichkeiten bei eingehendem IP-Verkehr

Bei eingehendem IP-Verkehr unterstützt IPSec die folgenden Kombinationen von AH und ESP.

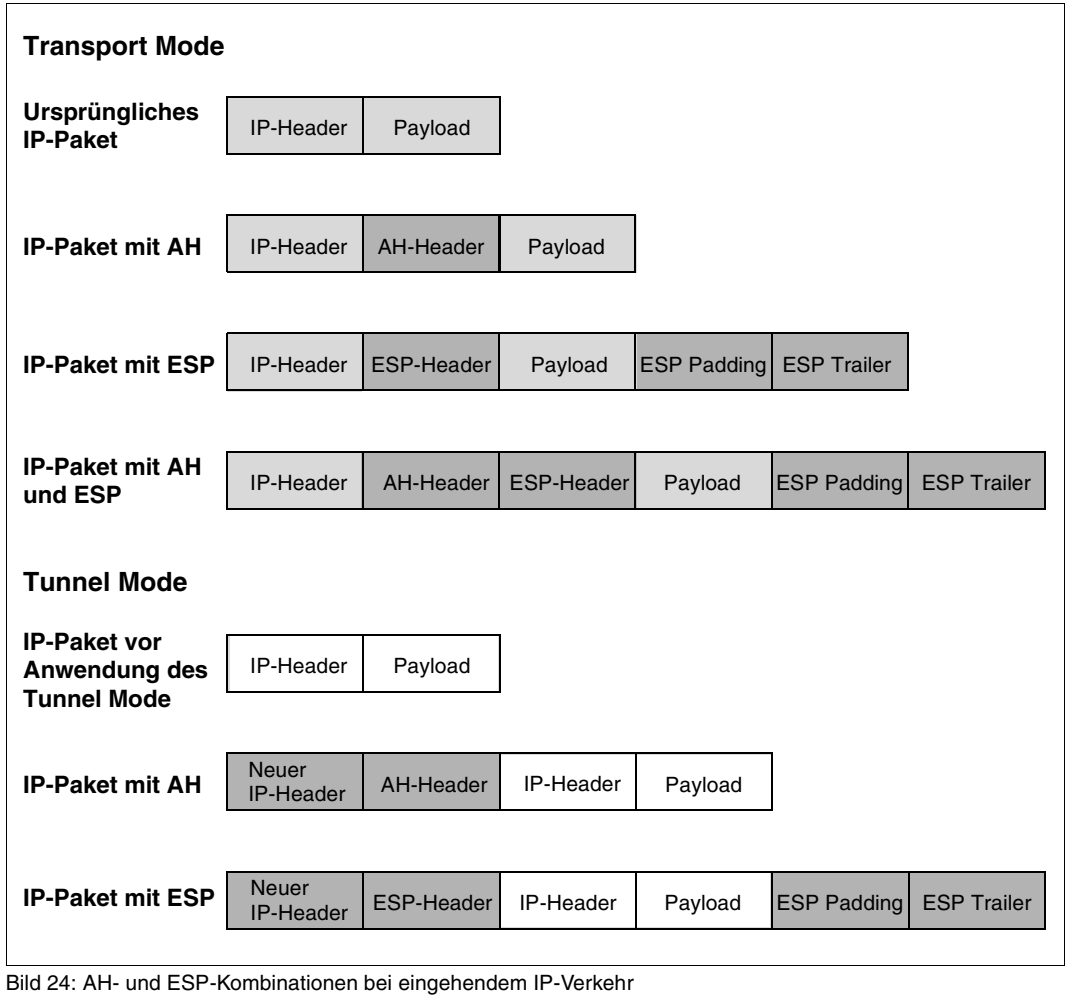


Bild 24: AH- und ESP-Kombinationen bei eingehendem IP-Verkehr

AH- und ESP-Kombinationsmöglichkeiten bei ausgehendem IP-Verkehr

Bei ausgehendem IP-Verkehr unterstützt IPSec auch andere Sequenzen von Authentication Headern und Encapsulating Security Payload Headern. Voraussetzung ist jedoch, dass eine zugehörige Security Policy für ausgehende Daten existiert.

8.3.3 Bandbreite der von IPSec unterstützten Sicherheitspolitiken

IPSec ermöglicht durch die Protokolle Authentication Header und Encapsulating Security Payload eine große Bandbreite an verschiedenen Sicherheitspolitiken.

Die auf den folgenden Seiten dargestellten Header-Layouts repräsentieren die verschiedenen Sicherheitsphilosophien.

Header-Layout im Transportmodus

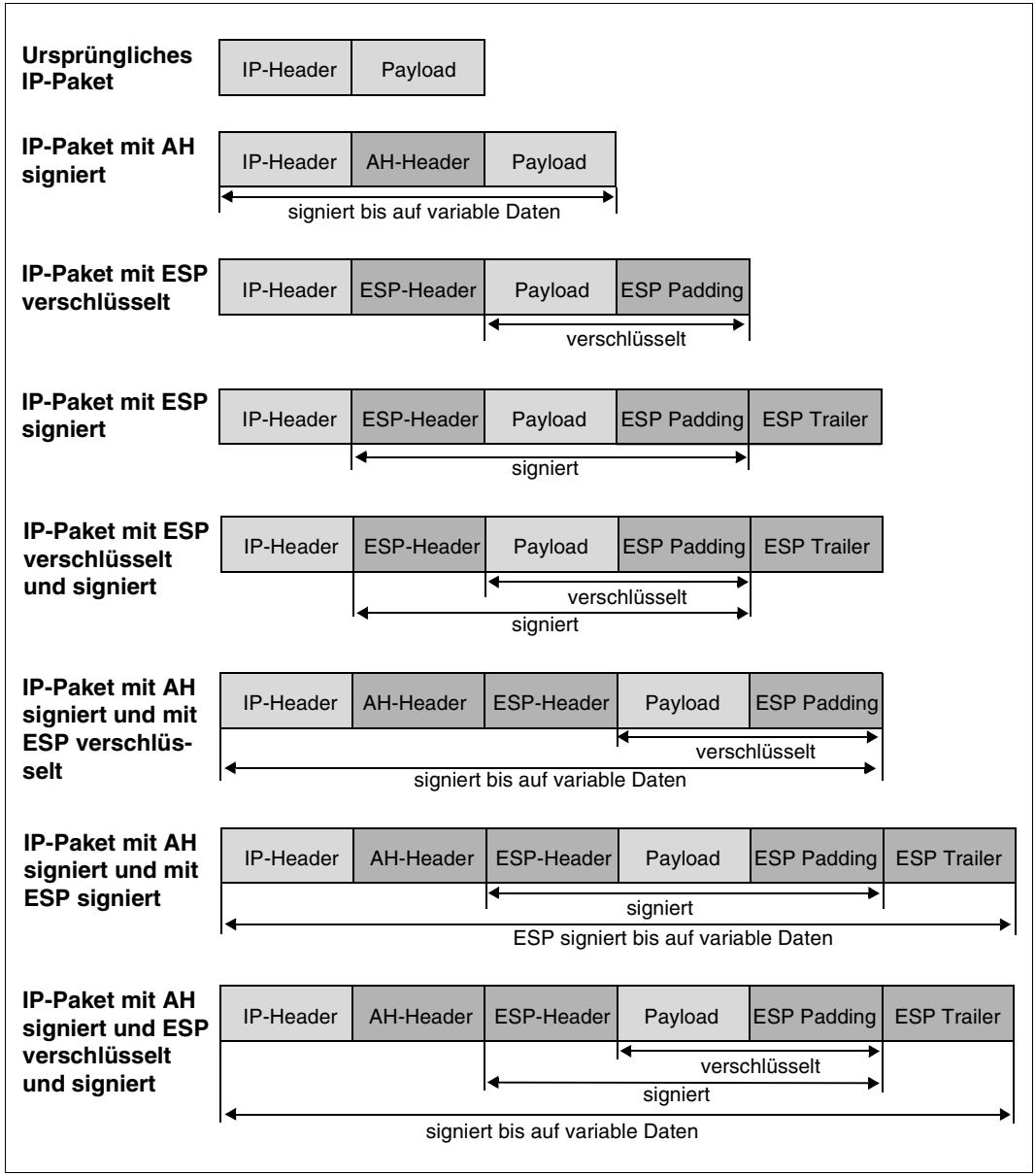


Bild 25: Header-Layout im Transportmodus

Header-Layout im Tunnelmodus mit Authentication Header im Tunnel

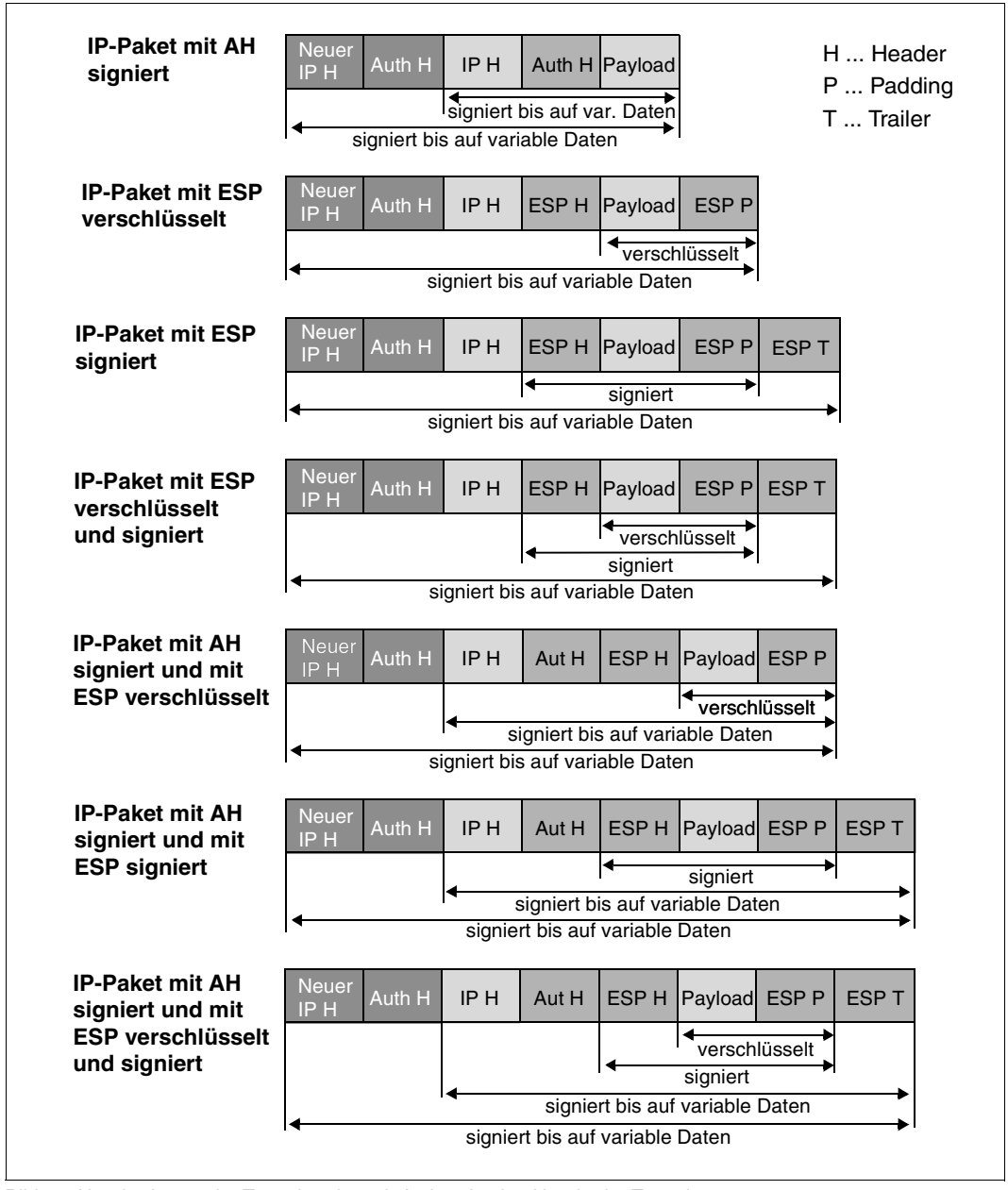


Bild 26: Header-Layout im Tunnelmodus mit Authentication Header im Tunnel

Header-Layout im Tunnelmodus mit ESP-Verschlüsselung im Tunnel

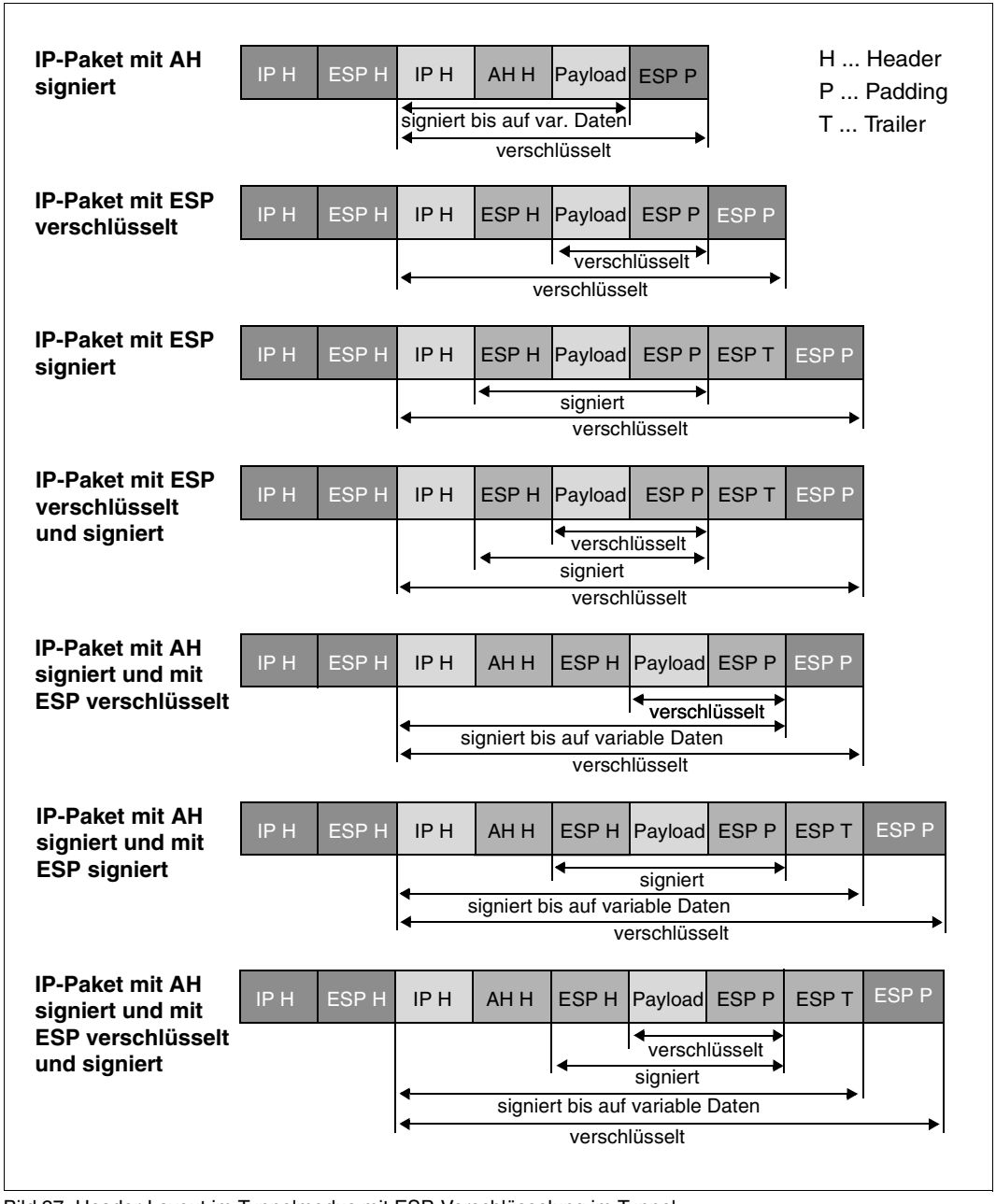


Bild 27: Header-Layout im Tunnelmodus mit ESP-Verschlüsselung im Tunnel

Header-Layout im Tunnelmodus mit ESP-Signierung im Tunnel

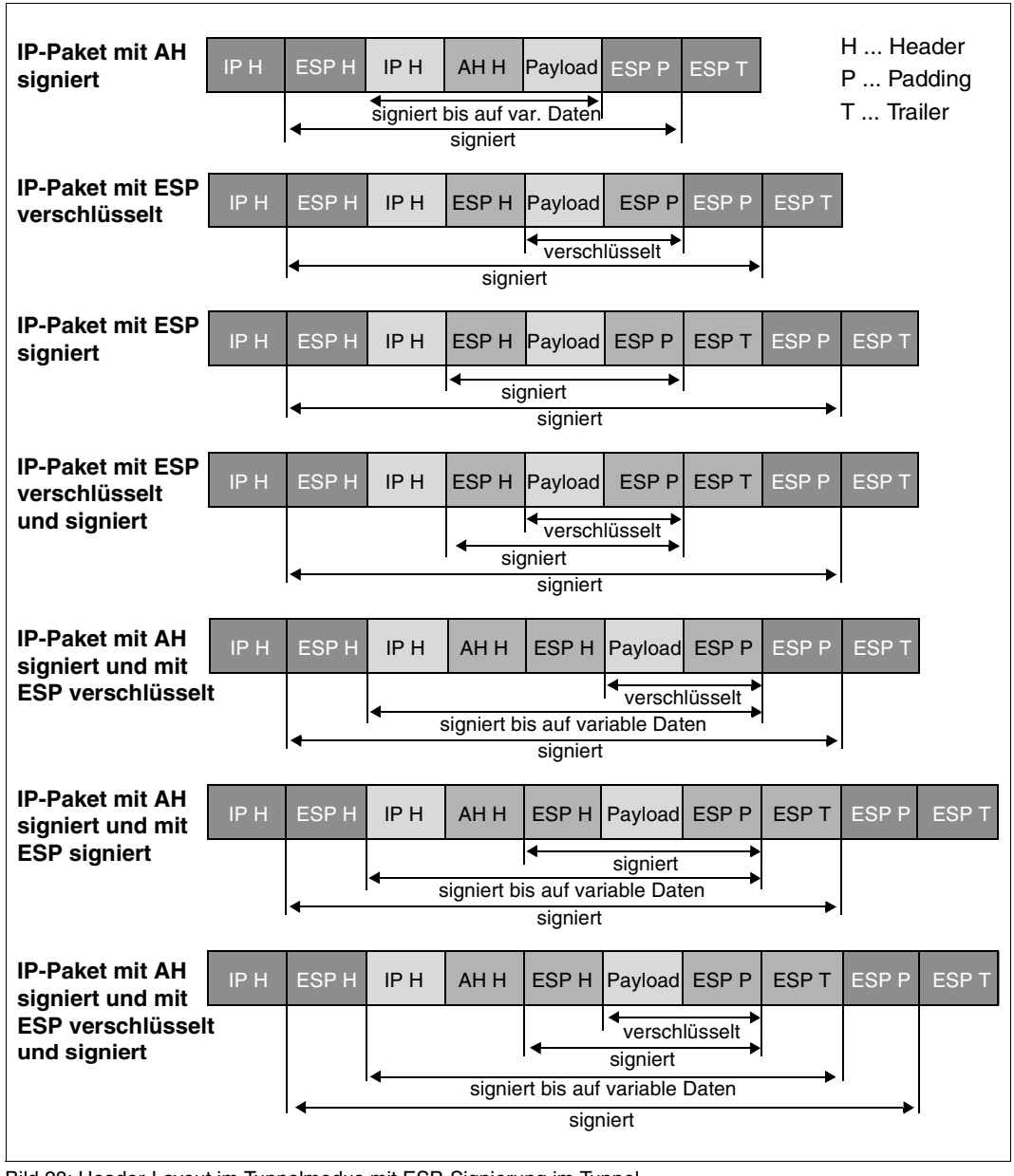


Bild 28: Header-Layout im Tunnelmodus mit ESP-Signierung im Tunnel

Header-Layout im Tunnelmodus mit ESP-Verschlüsselung und Signierung im Tunnel

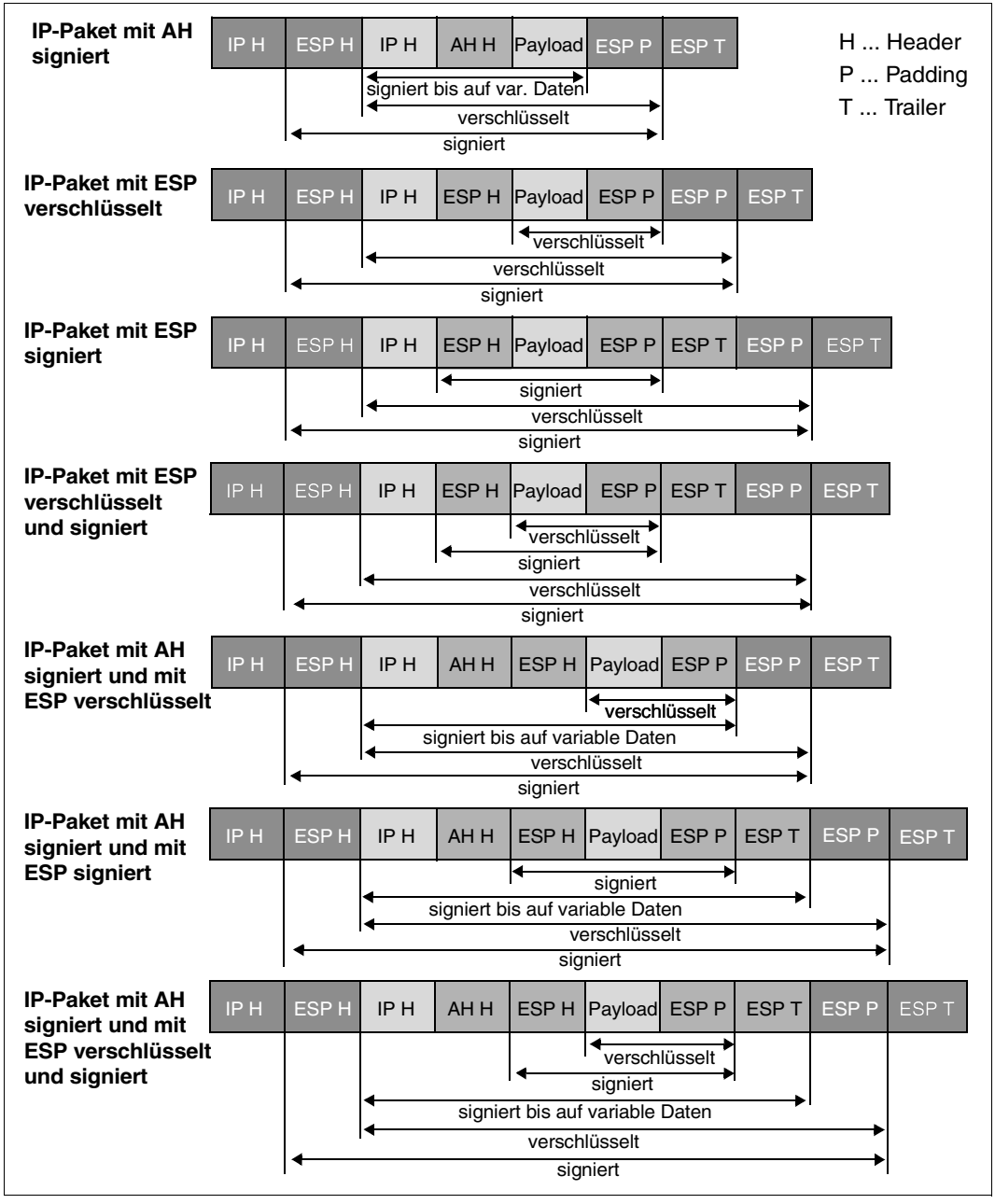


Bild 29: Header-Layout im Tunnelmodus mit ESP-Verschlüsselung im Tunnel

Header-Layout im Tunnelmodus ohne innere IPSec-Header

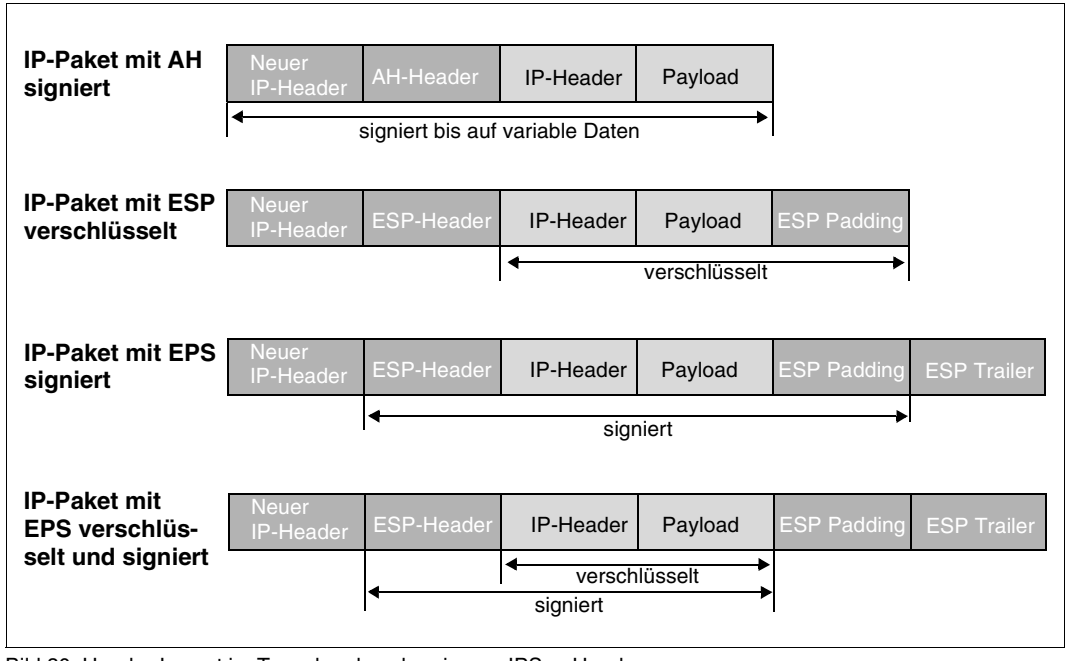


Bild 30: Header-Layout im Tunnelmodus ohne innere IPSec-Header

Abkürzungen

AH	Authentication Header
DNS	Domain Name System
ESP	Encapsulating Security Payload
FQDN	Fully Qualified Domain Name
HBCI	Homebanking Computer Interface
HTTP	Hypertext Transfer protocol
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IPCOMP	IP Payload Compression Protocol
ICV	Integrity Check Value
IETF	Internet Engeneering Task Force
IKE	Internet Key Exchange Protocol
IP	Internet Protocol
IPSec	Internet Security
ISAKMP	Internet Security Association and Key Management Protocol
IV	Initilization Vector
KMP	Key Management Protocol
MAC	Message Authentication Code
NAT	Network Address Translation
NAT-T	Network Address Translation - Traversal
NEA	Name der Transdata-Architektur
PFS	Perfect Forward Secrecy
PKI	Public Key Infrastructure
PSK	preshared secret key
RFC	Request for Comment
SA	Security Association
SAD	Security Association Data Base

SAT	Security Audit Trail
SET	Secure Electronic Transaction
SP	Security Policy
SPD	Security Policy Data Base
SPI	Security Parameter Index
SPMD	Security Policy Management Daemon
SSL	Secure Sockets Layer
TLS	Transport Layer Security
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ULP	Upper Layer Protocol

Fachwörter

AES (Advanced Encryption Standard)

Der Advanced Encryption Standard (AES) ist ein symmetrisches kryptografisches Verfahren, das als Nachfolger für DES bzw. 3DES im Oktober 2000 vom National Institute of Standards and Technology (NIST) als Standard bekannt gegeben wurde. Nach seinen belgischen Entwicklern Joan Daemen und Vincent Rijmen wird er auch Rijndael-Algorithmus genannt (gesprochen wie dt. "Reyndahl").

ANSI (American National Standards Institute)

Entwickelt Standards über verschiedene akkreditierte Normen-Gremien (Accredited
Standards Committee; ASC). Das X9-Komitee beschäftigt sich vorwiegend mit Sicherheitsstandards für Finanzdienstleistungen.

Asymmetrische Verschlüsselung

Bei einer asymmetrischen Verschlüsselung hat jeder Teilnehmer zwei Schlüssel: einen privaten (geheimen) und einen öffentlichen Schlüssel. Diese beiden Schlüssel stehen in einer mathematischen Beziehung zueinander.

Blockchiffre (Block cipher)

Die Blockverschlüsselung, auch Blockchiffre, ist ein Algorithmus der einen Datenblock mittels eines Schlüsselwerts verschlüsselt. Der verschlüsselte Block hat dabei die gleiche Länge.

CBC (Cipher Block Chaining)

Cipher Block Chaining Mode ist eine kryptografische Betriebsart, in der Blockchiffre betrieben werden können. Vor dem Verschlüsseln eines Klartextblocks wird dieser erst mit dem im letzten Schritt erzeugten Geheimtextblock per XOR (exklusives Oder) verknüpft.

Chiffrierter Text oder Chiffrat (Cipher text)

Das Ergebnis der Veränderung von Buchstaben oder Bits durch Ersetzung, Vertauschung oder beides.

Cryptoki

Programm-Schnittstelle zu Geräten, die kryptografische Informationen speichern und kryptografische Funktionen ausführen; spezifiziert vom PKSC#11-Standard.

DES (Data Encryption Standard; Datenverschlüsselungsstandard)

Ein 64-bit-Blockchiffrierer oder symmetrischer Algorithmus, der auch als Data Encryption Algorithm (DEA) (von ANSI) bzw. DEA-1 (von ISO) bezeichnet wird.

Diffie-Hellman

Der Diffie-Hellman-Schlüsselaustausch ist ein Protokoll aus dem Bereich der Kryptografie. Mit ihm erzeugen zwei Teilnehmer einen geheimen Schlüssel, den nur diese beiden kennen

Entschlüsselung (Decryption)

Der Prozess des Rück-Umwandelns von chiffriertem (verschlüsseltem) Text in Klartext.

FIPS (Federal Information Processing Standard)

Eine vom NIST veröffentlichte Norm der Regierung der USA.

Geheimer Schlüssel (Secret key)

Der „Sitzungsschlüssel“ in symmetrischen Algorithmen.

Hash-Funktion (Hash function)

Eine Einweg-Hash-Funktion ist eine Funktion, die aus einer i. a. großen Datenmenge eine i. a. kleinere Datenmenge erzeugt, die zur Erzeugung des Originals nicht umgekehrt werden kann.

HMAC (keyed-hash message authentication code)

Eine schlüsselabhängige Einweg-Hash-Funktion, die speziell für die Verwendung mit MAC (Message Authentication Code) gedacht ist und auf IETF RFC 2104 basiert.

IETF (Internet Engineering Task Force)

Die Internet Engineering Task Force (IETF) ist eine Organisation, die sich mit der technischen Weiterentwicklung des Internets befasst.

IKE (Internet Key Exchange)

Protokoll zur Einrichtung und Verwaltung von SAs

Initialisierungsvektor (Initialization Vector, IV)
Ein Block aus willkürlichen Daten, der unter Verwendung eines „Chaining Feedback Mode“ (siehe „CBC (Cipher Block Chaining)“) als Ausgangspunkt für einen Blockchiffrierer dient.

Integrität (Integrity)
Ein Beleg dafür, dass Daten bei der Speicherung oder Übertragung (durch unbefugte Personen) nicht verändert werden.

ISAKMP (Internet Security Association and Key Management Protocol)
Rahmen für die Schlüssel- und SA-Verwaltung.

ISO (International Organization for Standardization)
Diese Organisation ist für eine Vielzahl von Normen verantwortlich, wie das OSI-Modell sowie internationale Beziehungen mit dem ANSI bezüglich X. 509.

Klartext (Plain text oder clear text)
Daten oder Nachrichten in einer für den Menschen lesbaren Form vor dem Verschlüsseln – auch unverschlüsselter Text genannt.

MAC (Message Authentication Code)
Eine schlüsselabhängige Einweg-Hash-Funktion, bei der zur Verifizierung des Hash der identische Schlüssel benötigt wird.

MD5 (Message Digest 5)
MD5 (Message-Digest Algorithm 5) ist eine weit verbreitete kryptografische Hash-Funktion, die einen 128-Bit-Hashwert erzeugt. MD5 wurde 1991 von Ronald L. Rivest entwickelt

Mechanismus
Prozess zur Implementierung von kryptografischen Operationen.

Message Digest
Eine Prüfsumme, die aus einer Nachricht berechnet wird. Wenn Sie ein einziges Zeichen in der Nachricht verändern, hat die Nachricht einen anderen Message Digest.

NIST (National Institute for Standards and Technology)
Eine Abteilung des „U.S. Department of Commerce“ (Wirtschaftsministerium der USA). Veröffentlicht Normen bezüglich der Kompatibilität (FIPS).

NSA (National Security Agency)
Eine Abteilung des „U.S. Department of Defense“. Nimmt primär Aufgaben zur Informationssicherheit wahr.

Öffentlicher Schlüssel (Public key)

Die öffentlich verfügbare Komponente eines integrierten asymmetrischen Schlüsselpaares, die oft als Verschlüsselungsschlüssel bezeichnet wird.

PKCS (Public- Key Cryptography Standards)

Eine Reihe von De-facto-Normen zur Verschlüsselung mit öffentlichen Schlüsseln, die in Zusammenarbeit mit einem informellen Konsortium (Apple, DEC, Lotus, Microsoft, MIT, RSA und Sun) entwickelt wurden. Dazu gehören Algorithmen-spezifische und von Algorithmen unabhängige Implementierungsnormen. Spezifikationen zur Definition von Nachrichtensyntax und anderen Protokollen, die von RSA Data Security, Inc., gesteuert werden.

Privater Schlüssel (Private key)

Die im privaten Besitz befindliche „geheime“ Komponente eines integrierten asymmetrischen Schlüsselpaares, die oft als Entschlüsselungsschlüssel bezeichnet wird.

Pseudo-Zufallszahl (Pseudo-random number)

Eine Zahl, die aus der Anwendung von Algorithmen errechnet wird, die Zufallswerte auf Eingabewerte erzeugen, die aus der Computerumgebung abgeleitet sind (z.B. Maus-
koordinaten). Siehe „Zufallszahl (Random number)“.

RFC (Request for Comment)

Ein IETF-Dokument aus der Untergruppe FYI RFC, die Überblicke und Einführungen gibt, oder aus der Untergruppe STD RFC, die Internet-Normen angibt. Die Abkürzung FYI steht für „For Your Information“ – zu Ihrer Information. Jeder RFC hat zur Indizierung eine RFC-Nummer, anhand derer er abgerufen werden kann (www.ietf.org).

RSA

Abkürzung von RSA Security, Inc. Steht auch für die Firmenchefs Ron Rivest, Adi Shamir und Len Adleman oder bezieht sich auf den von ihnen erfundenen Algorithmus. Der RSA-Algorithmus wird in der Kryptographie mit öffentlichen Schlüsseln verwendet. Seine Funktionsweise beruht auf der Tatsache, dass zwei große Primzahlen zwar leicht miteinander zu multiplizieren sind, aber das Produkt nur schwer wieder in sie zu zerlegen ist.

Schlüssel (Key)

Ein Mittel zur Gewährung bzw. Verweigerung von Zugriff, Eigentumsrechten oder Steuerungsbefugnissen. Wird durch eine beliebige, große Anzahl von Werten dargestellt.

Schlüsselaustausch (Key exchange)
Ein Schema mit zwei oder mehreren Knoten zum Übertragen eines geheimen Sitzungsschlüssels über einen nicht gesicherten Kanal.

Schlüssellänge (Key length)
Die Anzahl der Bits zur Darstellung der Schlüsselgröße. Je länger der Schlüssel, desto stärker ist er.

Schlüsselverwaltung (Key management)
Das Verfahren zum sicheren Speichern und Verteilen kryptografischer Schlüssel. Der Gesamtprozess des sicheren Erstellens und Verteilens von kryptografischen Schlüsseln an befugte Empfänger.

SET (Secure Electronic Transaction)
Dient der sicheren Übertragung von Kreditkartennummern über das Internet.

SHA-1 (Secure Hash Algorithm)
Die 1994 vorgenommene Überarbeitung des vom NIST entwickelten SHA (FIPS 180-1). SHA-1 erzeugt einen 160-bit-Hash.

Sitzungsschlüssel (Session key)
Der geheime (symmetrische) Schlüssel zum Verschlüsseln aller Datensätze auf Transaktionsbasis. Für jede Kommunikationssitzung wird ein anderer Sitzungsschlüssel verwendet.

SSL (Secure Socket Layer)
Wurde von Netscape zur Gewährleistung von Sicherheit und zur Geheimhaltung im Internet entwickelt. Unterstützt die Server- und Client-Authentisierung und gewährleistet die Sicherheit und Integrität des Übertragungskanals. Wirkt auf der Übertragungsebene und dient als „Socket-Bibliothek“, wodurch eine anwendungsunabhängige Wirkungsweise ermöglicht wird. Verschlüsselt den gesamten Kommunikationskanal.

Stromchiffriercode (Stream cipher)
Eine Klasse der symmetrischen Schlüsselverschlüsselung, bei der die Umwandlung für jedes zu verschlüsselnde Symbol des Klartextes geändert werden kann; wird für Umgebungen mit geringer Speicherkapazität zum Puffern von Daten empfohlen.

Symmetrischer Algorithmus (Symmetric algorithm)
Wird auch als konventioneller, geheimer Schlüssel- oder Einzelschlüsselalgorithmus bezeichnet. Der Verschlüsselungsschlüssel ist entweder mit dem Entschlüsselungsschlüssel identisch, oder ein Schlüssel kann aus dem anderen abgeleitet werden. Es gibt zwei Unterkategorien – Block und Strom.

TLS (Transport Layer Security)

Ein IETF-Entwurf. Die Version 1 basiert auf der Version 3.0 des SSL-Protokolls und dient zur Wahrung der Privatsphäre bei der Kommunikation über das Internet.

Triple-DES, 3DES

Eine Verschlüsselungskonfiguration, in der der DES-Algorithmus dreimal mit zwei oder drei unterschiedlichen Schlüsseln verwendet wird.

Unverschlüsselter Text

siehe „Klartext (Plain text oder clear text)“.

Zufallszahl (Random number)

Ein wichtiger Aspekt für viele Verschlüsselungssysteme sowie ein notwendiges Element beim Erzeugen von eindeutigen Schlüsseln, die für Gegner nicht berechenbar sind. Echte Zufallszahlen werden normalerweise aus analogen Quellen abgeleitet und erfordern in der Regel den Einsatz von besonderer Hardware.

Literatur

Die Handbücher sind online unter *http://manuals.ts.fujitsu.com* zu finden oder in gedruckter Form gegen gesondertes Entgelt unter *http://manualshop.ts.fujitsu.com* zu bestellen.

openNet Server V3.4 (BS2000/OSD)
BCAM V21.0A
Benutzerhandbuch

openCRYPT V1.2 (BS2000/OSD)
Sicherheit mit Kryptographie
Benutzerhandbuch

openNet Server (BS2000/OSD)
IPv6 Einführung und Umstellhandbuch Stufe 1
Benutzerhandbuch

SOCKETS(BS2000) V2.4
SOCKETS für BS2000/OSD
Benutzerhandbuch

openNet Server V2.0, interNet Services V2.0
SNMP-Management für openNet Server und interNet Services
Benutzerhandbuch

interNet Services (BS2000/OSD)
Administratorhandbuch

interNet Services (BS2000/OSD)
Benutzerhandbuch

interNet Value Edition V1.0B (BS2000/OSD)
Benutzerhandbuch

IMON (BS2000/OSD)
Installationsmonitor
Benutzerhandbuch

BS2000/OSD-BC
Einführung in die Systembetreuung
Benutzerhandbuch

SECOS (BS2000/OSD)
Security Control System
Benutzerhandbuch

SECOS (BS2000/OSD)
Security Control System
Tabellenheft

RFCs

Umfassende Informationen zu den Request for Comments (RFCs) finden Sie auf der Home Page der Internet Engineering Task Force (IETF):

www.ietf.org

Stichwörter