INSTITUT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Systementwicklungsprojekt

Konzeption und Implementierung eines VPN für WLAN Zugänge auf IPSEC Basis mit Unterstützung differenzierter Benutzergruppen

Bearbeiter: Tobias Krause
Aufgabensteller: Prof. Dr. Heinz-Gerd Hegering
Betreuer: Dr. Helmut Reiser
Dipl.-Inform. Harald Rölle

INSTITUT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Systementwicklungsprojekt

Konzeption und Implementierung eines VPN für WLAN Zugänge auf IPSEC Basis mit Unterstützung differenzierter Benutzergruppen

Bearbeiter:	Tobias Krause
Aufgabensteller:	Prof. Dr. Heinz-Gerd Hegering
Betreuer:	Dr. Helmut Reiser Dipl.-Inform. Harald Rölle
Abgabetermin:	01. Februar 2003

Hiermit versichere ich, dass ich das vorliegende Systementwicklungsprojekt selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 01. Februar 2003

. . . .
(Unterschrift des Kandidaten)

Zusammenfassung

In diesem Systementwicklungsprojekt soll der WLAN Funkverkehr am LMU Lehrstuhl Kommunikationssysteme und Systemprogrammierung durch die Verwendung des IPSec Protokolls abgesichert werden. Hierfür wird den mobilen Endgeräten durch den IPSec Router ein VPN bereitgestellt. Zusätzlich zur Absicherung des Funkverkehrs durch die Verschlüsselung und Authentisierung der übertragenen Daten, wird einer bestimmten Benutzergruppe auch der Remote Zugriff auf lehrstuhlinterne Dienste ermöglicht. Diese Dieste können normalerweise nur von einem Rechner im privaten Lehrstuhlnetz verwendet werden. Damit der Zugriff auch von einem beliebigen Rechner im Internet funktioniert, muss dieser Rechner alle Vorraussetzungen zur Teilnahme am VPN erfüllen. Dazu gehört neben der entsprechenden Software auch ein X.509 Zertifikat, mit dem die Authentifizierung der unterschiedlichen Benutzergruppen geregelt wird. Nach der erfolgreichen Authentifizierung wird dem Remote Rechner dann eine Adresse aus dem Lehrstuhlnetz zur Verfügung gestellt, mit der er über den IPSec Tunnel auf die Dienste zugreifen kann.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

In stark zunehmenden Maße werden heute neben den leitungsgebundenen Netzen auch Netztechnologien eingesetzt, die einen Zugang über die Luftschnittstelle ermöglichen. Als Beispiel seien hier WLAN[1], Bluetooth, IrDA[2] u.ä. genannt. Mit Hilfe dieser leitungsungebundenen Techniken wird den unterschiedlichsten Endgeräten (Notebooks,PDAs, Mobiltelefonen, etc.) der Zugang zu Unternehmensnetzen oder auch dem Internet ermöglicht. Um die bestehenden Schwachstellen der Verschlüsselung auf Hardware-Ebene zu kompensieren, ist eine Lösung notwendig, die sowohl genug Sicherheit als auch genug Verbreitung besitzt.

Ziel des SEP ist es, unter Verwendung von IPSEC[3] eine sichere Kommunikation in der bestehenden WLAN Infrastruktur des Lehrstuhlnetzes aufzubauen. Den mobilen Endgeräten am Lehrstuhl soll eine Teilname an einem sicheren VPN[4] bereitgestellt werden. Mit Hilfe eines IPSEC Routers wird die Kommunikation zwischen den mobilen Endgeräten und dem Lehrstuhlnetz gesteuert. Dabei ist darauf zu achten, dass eine möglichst nahtlose Integration in bestehenden Sicherheitskonzepte (z.B. hinsichtlich der Zertifizierung und Verschlüsselung) erfolgt.

Im Einzelnen sollen vom IPSEC Router zwei Kommunikationswege angeboten und geschützt werden. Zum Einen die Verbindung vom mobilen Endgerät über den IPSEC Router ins Internet und andererseits soll für bestimmte Benutzergruppen ein Zugriff auf lehrstuhlinterne Ressourcen ermöglicht werden. Die dafür notwendige Authentifizierung der Teilnehmer muss mit X.509 Zertifikaten durchgeführt werden. Zu den umschriebenen Aufgaben des SEPs gehört die Konfiguration des IPSEC Routers unter LINUX, sowie die Bereitstellung einer genauen Konfigurationsanleitung für mobile Endgeräte mit den Betriebssystemen LINUX, Windows 2000 und Windows XP. Diese Konfigurationsarbeit am Router und den Endgeräten umfassen unter anderm die Sicherstellung der Verschlüsselung des Datenverkehrs, die Authentifizierung von Nutzern und Daten, sowie einer dynamischen Schlüsselverwaltung.

1.1 Kapitelübersicht

Kapitel 2 erörtert die Schwächen der gegenwärtigen WLAN Lösung

Kapitel 3 gibt einen Projektüberblick und stellt die einzelnen Schritte beim Aufbau eines VPNs vor

Kapitel 4 behandelt die Funktion und Sicherheit von IPSec.

Kapitel 5 beschreibt die Installation des Projekts und gibt genau Anleitung für Teilnahme am Versuchsnetz

Kapitel 6 befaßt sich mit der Performance des IPSec Systems und liefert Testergebnisse

[1]Wireless Local Area Network
[2]Infrared Data Association
[3]IP Security Protocol
[4]Virtual Private Network

Kapitel 2

Schwachstellen der vorhandenen WLAN Lösung

Zum Zeitpunkt dieser Ausarbeitung wird der WLAN Verkehr am LRZ und seinen angeschlossenen Wissenschaftseinrichtungen durch eine Kombination aus den Sicherheitsmechanismen WEP-40Bit und PPTP geschützt.

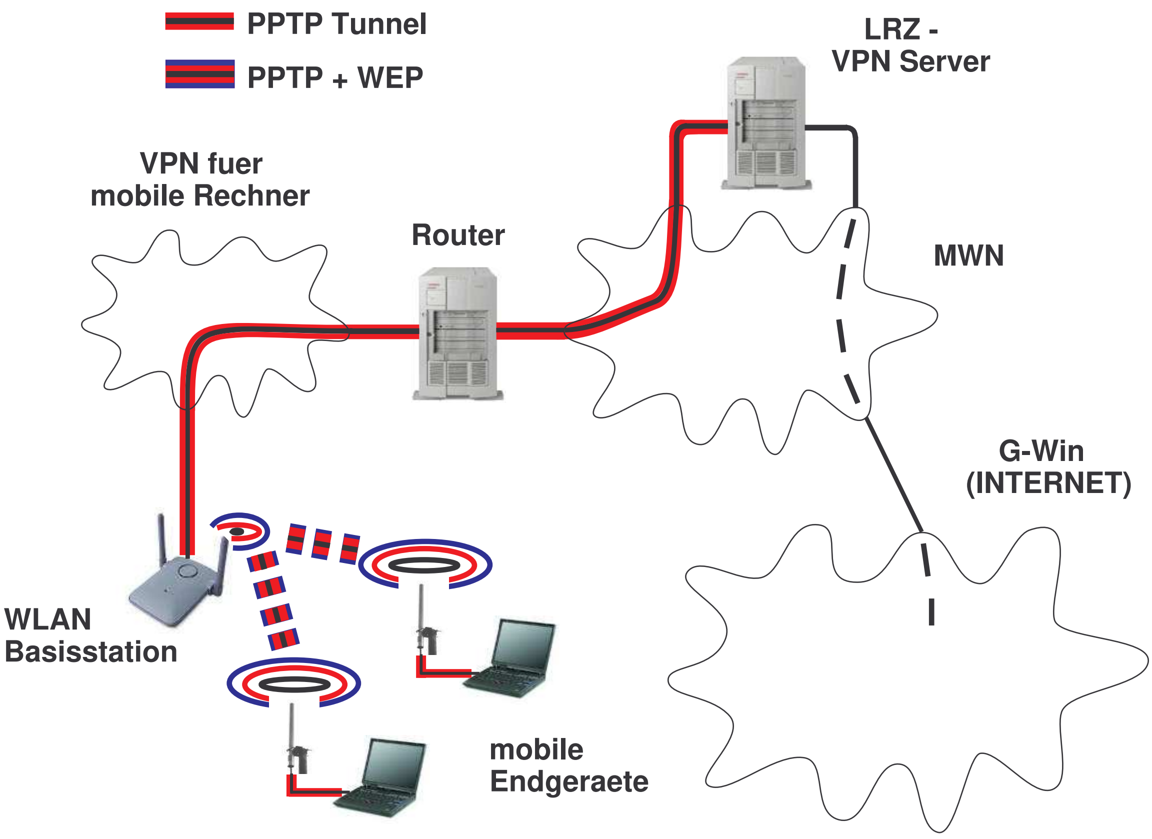

Abbildung 2.1: LRZ WLAN Lösung

Die in der Abbildung vorkommende WLAN Basisstation ist direkt am Münchner Wissenschaftsnetz (MWN) angeschlossen und wird mittels des Point-to-Point-Tunneling-Protocols (PPTP) in ein VPN [LRZ 02] eingebunden. Ohne die notwendige Authentifizierung an einem der VPN-Server des LRZ, kann kein Nutzer direkt über die WLAN Basisstation ins MWN bzw. Internet zugreifen. So kann durch die Verwendung von PPTP ein Schutz gegen den Missbrauch des Netzes zwischen der Basisstation und dem VPN-Server aufgebaut werden. Damit wird neben der Erhöhung der Sicherheit, gleichzeitig der administrative Aufwand gering gehalten.
WEP steht für Wired Equivalent Privacy und bezeichnet die im WLAN-Standard IEEE 802.11 verankerte Sicherheitskomponente. Die Anwendung des WEP Protokolls in den WLAN Komponenten führt zu einer Verschlüsselung der Luftschnittstelle. PPTP erweitert das Point-to-Point-Protocol (PPP) um VPN Funktionen und kann so für den Aufbau eines Virtuell Privaten Netzes eingesetzt werden. Bei diesem Layer-2-Tunnel-Konzept wird dem Original PPP-Paket ein zusätzlicher IP-Header vorangestellt, der dann mit seinen Quell- und Ziel-Angaben für die Weiterleitung durch das Transitnetz verwendet wird. Dadurch können unterschiedliche Protokolle wie IP, IPX und NetBEUI gleichermaßen getunnelt werden.

2.1 WEP Protokoll

Zur Sicherstellung der Vertraulichkeit bei der Übertragung von Daten verwendet das WEP-Protokoll den Verschlüsselungs-Algorithmus RC4. Die Datenintegrität wird durch eine 32-Bit-CRC-Checksumme (IC) gebildet, die gemeinsam mit dem Klartext verschlüsselt wird. Es wurden jedoch bei der Kombination aus WEP-Schlüssel und dem Stromchiffrier-Algorithmus RC4 in der Vergangenheit einige Schwachstellen entdeckt, die eine Vielzahl von Angriffsmöglichkeiten bieten.

Für die paketweise Verschlüsselung der Daten wird aus einem Schlüssel und einem Initialisierungsvektor (IV) ein pseudo-zufälliger Bitstrom erzeugt. Dieser zur Verwendung kommende Schlüssel wird auch WEP-Key oder shared secret key genannt. Er besteht aus einer Zeichenkette mit einer Länge von wahlweise 40 Bit bzw. 104 Bit beim WEP2 -Protokoll und muss den am WLAN beteiligten Clients, sowie dem Access Point, vor der Verschlüsselung zur Verfügung gestellt werden. Der aus den Schlüsseln generierte, pseudo-zufällige Bitstrom wird danach bitweise mit den Datenpaketen XOR verknüpft [Abbildung WEP Verschlüsselung]. Hieraus entsteht der Chiffretext, der bei dem Empfänger durch denselben Bitstrom und durch die XOR Verknüpfung wieder bitweise zu dem Klartext wird [Abbildung WEP Entschlüsselung].

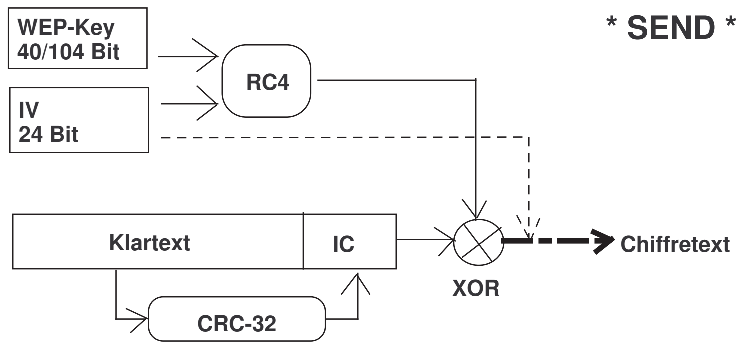

Abbildung 2.2: WEP Verschlüsselung

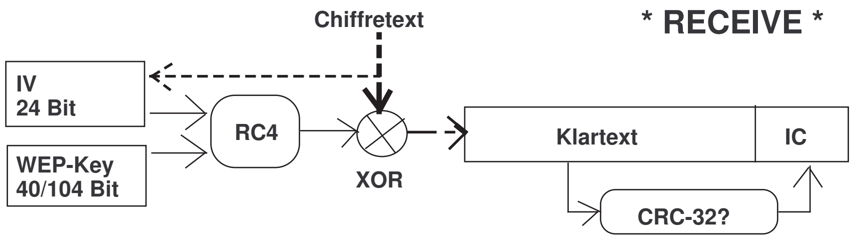

Abbildung 2.3: WEP Entschlüsselung

Beim Initialisierungsvektor (IV) handelt es sich um einen 24 Bit Wert, der für jedes Datenpaket unterschiedlich sein sollte und unverschlüsselt vor dem verschlüsselten Datenpaket über das drahtlose Funknetz übertragen wird. Manche Hersteller geben für den WEP-Key auch die Schlüssellänge (WEP-Key + Initialisierungsvektor) 40+24=64 bzw. 104+24=128 Bit an.

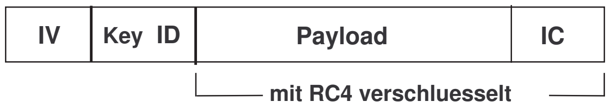

Abbildung 2.4: WEP Paket

Die Authentifizierung zwischen Mobilstation und Basisstation findet durch ein Challenge-Response-Verfahren statt. Hierfür erzeugt der Access-Point 128 zufällige Bytes und sendet sie in einem Datenpaket unverschlüsselt an einen Client. Dieser Client verschlüsselt das Datenpaket und sendet es an die Basisstation zurück (Response). Der Client hat sich erfolgreich bei der Basisstation authentifiziert, falls es der Basisstation möglich ist, die Response zur Challenge wieder zu entschlüsseln. Der Schlüssel, der zur Authentisierung und zur Verschlüsselung der Nutzdaten verwendet wird, ist dabei derselbe. Weitere Einzelheiten dazu in [SIKO 01].

2.2 Schwachstellen des WEP Protokolls

2.2.1 Länge des WEP-Keys

Das Bundesamt für Sicherheit in der Informationstechnik listet auf seiner Internetseite [BSI 02] eine Vielzahl von Schwachstellen des WEP Protokolls auf. Beispielsweise ist der WEP-Key mit einer Schlüssellänge von 40 Bit zu kurz. Ein Angreifer kann den Chiffretext aufzeichnen und mit einem handelsüblichen PC alle in Frage kommenden Schlüssel durchprobieren. Anhand eines vernünftigen Klartextes könnte so der gesuchte Schlüssel in Tagen herausgefunden werden und einem Mitlesen des Netzverkehrs stünde dann bis zum nächsten Schlüsseltausch nichts mehr im Wege. Der 104 Bit WEP-Key ist dagegen im Moment ausreichend, um sich gegen das Durchprobieren der Schlüssel zu schützen.

2.2.2 Länge des Initialisierungsvektors

Der Initialisierungsvektor (IV) ist mit einer Länge von 24 Bit ebenfalls zu kurz [BSI 02]. Die Verwendung des oben beschriebenen Stromchiffrier-Algorithmus kann nur als sicher angesehen werden, wenn der generierte Bitstrom für je zwei Datenpakete unterschiedlich ist. Werden zwei Datenpakete mit demselben Bitstrom verschlüsselt, so lassen sich sowohl die beiden Datenpakete als auch der Bitstrom in vielen Fällen rekonstruieren. Sofern der IV zufällig generiert wird, ist spätestens nach $2^{24} \approx 16,8$ Millionen verschiedenen Schlüsseln die erste Wiederholung eines IVs zu erwarten.

Bei einem Schlüsselraum von $2^{24} \approx 16,8$ Millionen und einer angenommenen Paketlänge von beispielsweise 1000 Byte, Transferrate von 11 MBit pro Sekunde, wiederholt sich spätestens nach 3,5 Stunden ein Initialisierungsvektor (IV).

In dieser Zeit werden dann höchstens 16 GByte übertragen. Der Verkehr könnte also in der Zeit mit einem Notebook mitgeschnitten werden, um so die Pakete mit gleichem IV und damit identischem RC4-Schlüssel zu erhalten. Die Sicherheitslücke des zu kurzen IVs ist unabhängig von der WEP-Schlüssellänge. Eine Erhöhung auf 104 Bit hat deshalb auch keinen Einfluss auf die beschriebene Problematik.

Dies waren nur zwei Beispiele einer ganzen Reihe von massiven Sicherheits-Schwachpunkten, die eindeutig gegen die alleinige Verwendung der Sicherheitskomponente WEP beim WLAN sprechen.

2.3 PPTP

Wie oben bereits beschrieben, ist PPTP die Abkürzung für Point-to Point-Tunnel-Protocol und erweitert das PPP Protokoll mit Hilfe der modifizierten Form des Generic Routing Encapsulation (GRE) Protokolls um VPN Eigenschaften. Für die Authentifizierung von Nutzern werden das Password Authentification Protocol (PAP) und das Challenge Protocol (CHAP) verwendet. Zur Datenverschlüsselung kann wie beim WEP-Protokoll der RC4 Algorithmus eingesetzt werden. Das Passwort trägt bei der von Microsoft Produkten eingesetzten PPTP Variante gleichzeitig zur Berechnung des 40-bit-session-keys bei, der zur Verschlüsselung der Datenpakete eingesetzt wird. Neben dem Passwort wird zur Erzeugung des keys eine Challenge verwendet [vgl. Challenge Response Verfahren 2.1], die zwischen dem Server und dem Client unverschlüsselt übertragen wird. Aus der Challenge und dem Passwort wird dann bei den Kommunikationspartnern der gleiche session key erzeugt. Mit ihm werden die Pakete verschlüsselt, bevor sie dann durch den Tunnel übertragen werden. Bei der Verwendung einer anderen Challenge, wird ein neuer session key berechnet. Eine Paket-Integritätsüberprüfung ist bei PPTP nicht implementiert.

2.4 Schwächen von PPTP

In den von Microsoft verwendeten Authentifizierungsmechanismen CHAPv1 und dem verbesserten CHAPv2 wurden in den vergangenen Jahren erhebliche Sicherheitslücken entdeckt. Ein Paper von Bruce Schneider und Peter Mudge[SchM 99] beschreibt, wie das Authentifizierungsprotokoll CHAP und der RC4 Algorithmus gebrochen werden können. Sie bemängeln außerdem, dass die Sicherheit des Authentifizierungs- und Verschlüsselungsprotokolls von der Qualität des gewählten Passworts abhängt. Das Sicherheitsrisiko bei einer unzureichenden Schlüssellänge wurde bereits beim WEP Protokoll genauer beschrieben.

Der Grund, warum das LRZ die 40 Bit Verschlüsselung des WEP-40 Protokolls verwendet und nicht die Verschlüsselung des PPTP Protokolls, liegt auf der Hand. Eine Entschlüsselung der durch PPTP übertragenen Datenpakete, würde den LRZ VPN Server sehr stark belasten. Dagegen wird die WEP-40 Bit Verschlüsselung dezentral in den WLAN Komponenten vor Ort durchgeführt.

Die Layer-2-Tunnel Protokolle L2F und L2TP kamen bei diesem Systementwicklungsprojekt nicht zum Einsatz, deren Schwächen werden deshalb nicht näher beschrieben. Der folgende Vergleich [LIPP 01] deutet jedoch die Leistungsfähigkeit von IPSec im Gegensatz zu den anderen Protokollen an:

	IPSec	PPTP	L2TP	L2F
Protokollschicht ISO/OSI	Schicht 3	Schicht 2	Schicht 2	Schicht 2
Standardisiert (RFC)	Ja	Nein	Ja	Nein
Paket-Authentifizierung	Ja	Nein	Nein	Nein
Benutzer-Authentifizierung	Ja	Ja	Ja	Ja
Datenverschlüsselung	Ja	Ja	Nein	Nein
Schlüsselmanagement	Ja	Nein	Nein	Nein
QoS-Signalisierung	Ja	Nein	Nein	Nein
IP-Tunneling	Ja	Ja	Ja	Ja
IPX/X.25-Tunneling	Nein	Ja	Ja	Ja
Verbindungs-Modell	Ende-zu-Ende	Ende-zu-Ende	Povider-Enterpr.	Povider-Enterpr.

Tabelle 2.1: Vergleich von Tunneling Protokolle

Kapitel 3

Projektübersicht

3.1 Motivation

Eines der Ziele des Systementwicklungsprojekts ist der Versuchsaufbau eines sicheren Virtuellen Privaten Netzes zur Absicherung des WLAN Funkverkehrs am Lehrstuhl für Kommunikationssysteme und Systemprogrammierung. Durch die Verwendung des IPSec-Tunneling-Protokolls soll so die sichere Kommunikation über das WLAN-Funknetz gewährleistet werden. Auf die Schwächen der gegenwärtig vom LRZ eingesetzten WEP-40 und PPTP Sicherheitsmechanismen wurde im zweiten Kapitel der Ausarbeitung genauer eingegangen. Die sicherheitstechnische Analyse des IPSec Protokolls folgt im nächsten Kapitel.

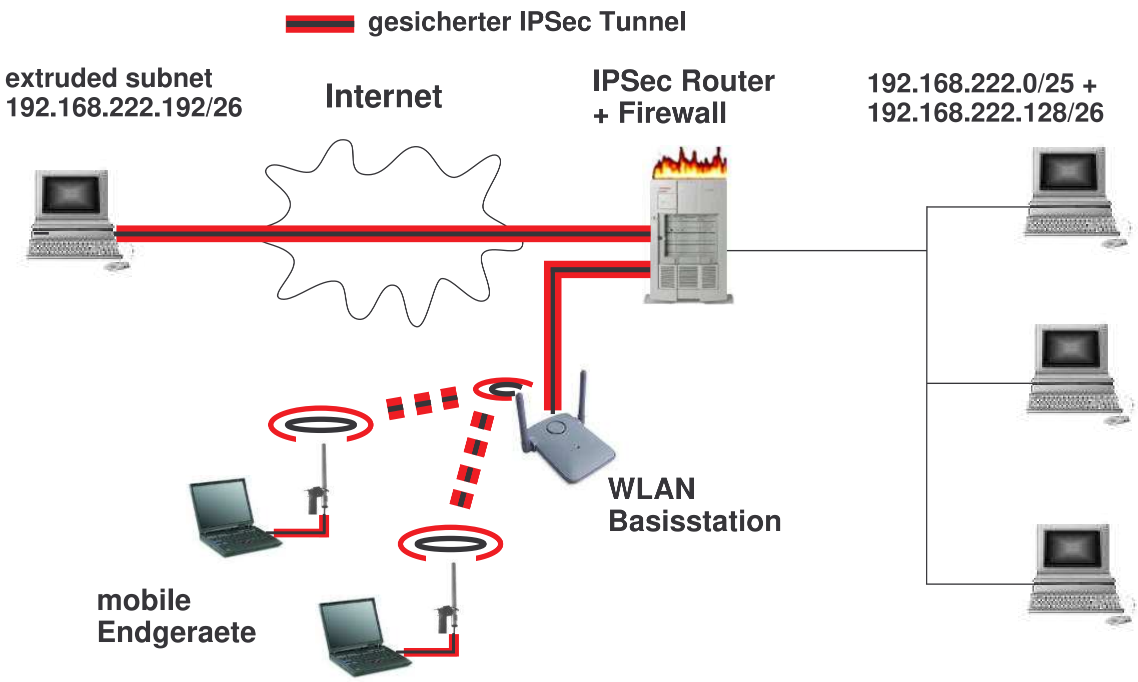

Abbildung 3.1: Projektübersicht

Bei der Abbildung oben wurden Teile der LRZ Infrastruktur absichtlich weggelassen, um die Darstellung übersichtlicher zu machen und die für den IPSec Betrieb entscheidenen Komponenten hervorzuheben. Wird das Projekt direkt am Lehrstuhl eingesetz, so müssen sich die mobilen Endgeräte natürlich zuvor am LRZ VPN Server authentifizieren, bevor sie mit anderen Rechnern kommunizieren können [vgl. Abbildung 2]. Bei diesem Beispiel würde sich der LRZ VPN Server dann zwischen der WLAN Basisstation und dem IPSec Router befinden.

In der Abbildung werden die von dem Projekt geforderten Kommunikationswege beschrieben. Zum einen die gesicherte IPSec Verbindung von den mobilen Endgeräten über den IPSec Router in das MWN bzw. INTERNET und zum anderen ein Remote Tunnelzugriff von einem authentifizierten Rechner (extruded subnet) über den IPSec Router auf die Dienste des internen Lehrstuhlnetz (Netzbereich auf der rechten Seite der Abbildung).

Nachfolgend werden zuerst die Funktionen der einzelnen Komponenten dieser IPSec Lösung beschrieben, bevor im Abschnitt [6.2] eine genauere Beschreibung der Kommunikation zwischen den Komponenten erfolgt.

3.2 IPSec Router

Das Versuchsnetz (vgl. Abbildung 3.1) besteht aus einem zentralen Router, auf dem die IPSec Software FreeS/WAN installiert wird. Der Router sorgt für den Aufbau eines VPNs und regelt die Authentifizierung der am VPN teilnehmenden Benutzer. Außerdem repräsentiert er im Versuchsnetz gleichzeitig das Bindeglied zwischen dem internen Lehrstuhlnetz und dem externen Münchener Wissenschaftsnetz (bzw. INTERNET). So hat er neben der Funktion als VPN Server zur Absicherung des WLAN Verkehrs auch die Aufgabe, den Verkehr zwischen den beiden angesprochenen Netzen zu regeln und die Kommunikation durch eine Firewall zu überwachen. Die Aufgabe als Router für das interne Netz eröffnet diesem Projekt noch weitere Möglichkeiten. Den Mitarbeitern am Lehrstuhl könnte über einen sicheren VPN Remote-Zugriff spezielle Dienste zugänglich gemacht werden, die normalerweise nur von einer Adresse aus dem internen Netz abrufbar sind. Dafür wird ein Adressbereich des internen Netzes für die Remote-Rechner reserviert und einzelne Adressen aus diesem Netzbereich durch den IPSec Tunnel zur Verfügung gestellt. Die Authentifizierung der Nutzer erfolgt wie auch beim VPN-Zugriff über WLAN mit Hilfe von X.509 Zertifikaten. Mit deren Kontrolle ist es möglich, verschiedenen Benutzergruppen unterschiedliche Zugriffsmöglichkeiten zu erteilen.

3.2.1 IPRoute2 Einsatz

Damit einem Remote-Rechner die interne Netzadresse zugewiesen werden kann, wird ein virtuelles Interface ethX auf dem externen physikalischen Interface des Remote-Rechners erzeugt und mit der für sie bestimmten internen Adresse belegt. Mit Hilfe von IPRoute2 werden die Pakete von diesem virtuellen Interface ethX aus an das IPSec Interface ipsec0 geschickt und von dort zum IPSec Router getunnelt. Diese Möglichkeit ein Paket nicht nur nach dem Ziel, sondern auch anhand des Ursprungs eines Pakets zu routen, ist Bestandteil der Routing Policy von IPRoute2. Dazu gehört ebenfalls die Möglichkeit, Pakete anhand des Protokollheaders zu routen. So könnte der IPSec Router bei einer Überlastung durch einen zu hohen Verschlüsslungsaufwand einige Protokolltypen (z.B.: HTTP,FTP, SSH..) am IPSec Tunnel vorbeisenden und dadurch eine Entlastung erreichen [weitere Informationen hierzu unter [ipr 02][Hube 02]]. Eine sicherheitstechnische Aufwertung des IPSec Systems ist durch den Einsatz von IPRoute2 in Kombination mit der iptables Firewall ebenfalls möglich.

Die genau Beschreibung vom IPRoute2 Einsatz auf den Client Rechner sowie dem IPSec Router folgt im Kapitel [5].

3.3 Aufbau des VPN

3.3.1 Verbindungsaufbau zwischen mobilem Endgerät und IPSec Router

1. **Authentifizierung mit Username und Passwort am LRZ-VPN-Server mittels PPTP**
2. **PPTP Tunnelaufbau zwischen Endgerät und LRZ-VPN Server**
3. **Authentifizierung mit X.509 Zertifikat beim IPSec Router**
4. **IPSec Tunnelaufbau vom mobilen Endgerät bis zum IPSec Router**

Ein Paket vom Endgerät zum IPSec Server durchläuft folgende Schritte:

Zuerst wird das Paket durch IPSec verschlüsselt und authentisiert, außerdem bekommt es einen neuen Header mit der Zieladresse des IPSec Routers. Dann wird dem Paket durch PPTP ein neuer Header vorangestellt und daraufhin durch das WEP-40 Protokoll in der WLAN Funk Karte verschlüsselt. Von der Basisstation wieder entschlüsselt, wird es anhand der PPTP Zieladresse zum LRZ-VPN Server geleitet und dort der PPTP Header entfernt. Es tritt wieder der IPSec Header nach vorn und deshalb wird das Paket dann zum IPSec Router geleitet. Am IPSec Server angekommen wird der AH Header entfernt, das Paket entschlüsselt und weitergeleitet.

Verbindung für Remote-Zugriff auf interne Dienste

Der Verbindungsaufbau unterscheidet sich von der oben beschriebenen Variante. Es muss keine Authentifizierung an einem VPN-Server des LRZ stattfinden, außer der Rechner ist ein mobiles Endgerät im MWN.

1. **Ein Rechner mit gültigem X.509 Zertifikat kann sich direkt am IPSec Router authentifizieren**
2. **Daraufhin wird ein IPSec Tunnel zwischen dem Rechner und dem IPSec Router etabliert.**
3. **Nun muss auf dem Rechner ein virtuelles Interface mit der vom IPSec Router zur Verfügung gestellten Adresse erzeugt werden.**

Der Remote-Zugriff mit der internen Adresse funktioniert nach folgendem Schema:

Das virtuelle Interface ethX muss auf dem physikalischen Interface installiert werden, auf dem der Verkehr zum virtuellen ipsec Interface geleitet wird. Mit Hilfe von IPRoute2 werden die Pakete von diesem neuen virtuellen Interface ethX aus zum ipsec Interface geleitet. Dadurch bekommen die Pakete die private Absenderadresse des virtuellen Interface ethX und werden vom ipsec Interface verschlüsselt und authentisiert. Durch den neuen IPSec Header werden sie nun zum IPSec Router geleitet und dort wieder entschlüsselt. Nun ist der innere Header wieder aktiv und sorgt für die Weiterleitung des Pakets zum gewünschten Ziel-Rechner, der dann den Ursprungs-Rechner des Pakets im internen Lehrstuhlnetz vermutet.

Warum nur die Inhaber eines bestimmten Zertifikats den Remote-Zugriff auf die internen Dienste durchführen können, wird in dem folgenden Abschnitt erklärt.

3.3.2 Zugriffskontrolle mit Hilfe von Zertifikaten

Neben der Gültigkeit eines Zertifikats gibt es noch ein weiteres Merkmal, mit dem der Zugriff auf die verschiedenen Dienste gesteuert wird. Dabei handelt es sich um den sogenannten Distinguished Name (DN), der bei der Erzeugung eines Zertifikats festgelegt wird und dadurch erst für die Unterscheidbarkeit sorgt. Diese ID eines Zertifikats wird bei der Authentifizierung mit übertragen und kann vom IPSec System verarbeitet werden. In der zentralen Konfigurationsdatei der IPSec Systeme können Einzelverbindungen über diese ID gesteuert werden. So kann einem Kommunikationspartner, der über die richtige signierte ID verfügt, die Benutzung des Tunnels mit Adressen aus einem bestimmten Bereich freigegeben werden.

Diese Eigenschaft ist besonders für den Remote-Zugriff auf die internen Dienste notwendig, da der Remote-Rechner schließlich über eine private Adresse aus dem internen Lehrstuhlnetz verfügen muss. Diese private Adresse wird in der FreeS/WAN Konfigurationsdatei durch die Angabe eines privaten Subnetzes für die Tunnelbenutzung freigeschaltet. So entsteht eine Verbindung zwischen Zertifikat-ID des Kommunikationspartners und der Freischaltung von privaten Adressen, ohne die eine Benutzung des IPSec Tunnels für den Remote-Zugriff nicht möglich ist.

Wie die IPSec Tunnel-Policy definiert ist [4.5.1] und wie ein Distinguished Name [5.2] festgelegt wird, kann in den folgenden Kapiteln nachgelesen werden.

Kapitel 4

Funktion und Sicherheit von IPSec

Das IPSec Protokoll wurde von der IPv6 Arbeitsgruppe der IETF (Internet Engineering Task Force)[IETF 02] entwickelt. Ziel war eine Sicherheitsarchitektur für den Nachfolger des Internetprotokolls IPv4 zu erschaffen. Während der Entwicklung erkannte man jedoch sehr schnell, dass dieses Sicherheitsprotokoll nicht nur Bestandteil des IPv6 Protokolls sein dürfte. Daraufhin wurde eine eigene IPSec Arbeitsgruppe mit der Weiterentwicklung beauftragt, die ihre Aktivitäten unabhängig vom IP-Protokoll vorantreiben sollte. So wurde IPSec nicht nur in das IPv6 Protokoll integriert, sondern kann auch mit IPv4 verwendet werden.

4.1 Die IPSec Architektur

IPSec unterstützt die folgen drei Verschlüsselungsszenarien:

- **Gateway-zu-Gateway**
- **Host-zu-Gateway**
- **Host-zu-Host**

Die IPSec Architektur beinhaltet dafür die beiden Protokolle AH und ESP, sowie eine Schlüsselverwaltung. Das soll den vertraulichen und authentisierten Transport von IP-Paketen garantieren. Dabei hat das Authentification Header (AH) Protokoll die Aufgabe, dass Herkunftsangabe und Inhalte der transportierten Daten nicht unbemerkt durch Dritte verändert werden können. Das Enscapulating Security Protokoll (ESP) umfasst zusätzlich die Möglichkeit, Daten verschlüsselt zu transportieren. Beide Protokolle können sowohl einzeln, als auch in Kombination eingesetzt und jeweils im Transport- oder Tunnelmodus betrieben werden[vgl.[Kow 02]].

4.2 Authentification Header (AH)

Um ein Datenpaket zu authentisieren, die Integrität des Pakets sicherzustellen und mehrfach gesendete Pakete zu erkennen und abzuwehren, verwendet das AH-Protokoll sogenannte Message Authentification Codes (HMAC). Dazu wird über eine Hashfunktion ein kryptographisch sicherer Wert über alle statischen IP-Headerdaten, sowie den gesamten Nutzdaten errechnet und zusammen mit weiteren Kontrollfeldern hinter dem IP-Header eingefügt. Der im nachfolgenden AH Paket Format aufgeführte Security Parameter Index (SPI) wird im Abschnitt [4.4.2] erklärt.

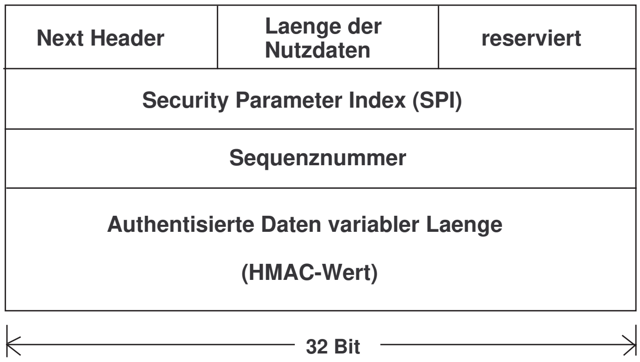

Abbildung 4.1: AH Paket Format

4.2.1 AH Transport Mode

In der Transportvariante befindet sich der AH zwischen dem IP Header des Pakets und vor den Nutzdaten. Das Feld 'Prot. Header d. n. Schicht' beinhaltet den Protokoll Header der nächsthöheren Schicht (z.B.: UDP, TCP, ICMP,...).

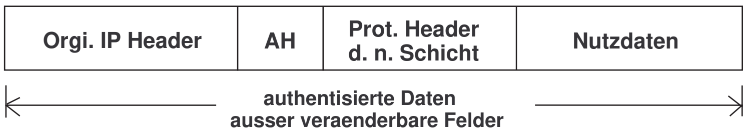

Abbildung 4.2: AH Protokoll / Transport Modus

4.2.2 AH Tunnel Mode

Bei der Tunnelvariante verpackt ein Sicherheits-Gateway das Paket mit Header und den Nutzdaten in ein neues IP Paket und fügt den Authentification Header zwischen dem neuen Header und dem ursprünglichen IP-Header ein. Auch hier wird der Hashwert über den gesamten statischen Bereich des neuen Pakets errechnet. Jedoch werden veränderbare Felder, wie zum Beispiel das "Time To Live"Feld nicht in die Berechnung mit einbezogen.

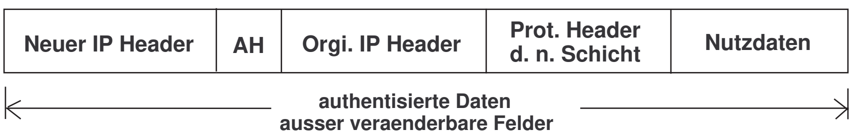

Abbildung 4.3: AH Protokoll / Tunnel Modus

4.3 Encapsulating Security Payload (ESP)

Neben der Erkennung und Abwehr von mehrfach gesendeten Paketen, der Authentisierung und der Integritätssicherung der übertragenen Daten, kann das ESP-Protokoll die Daten zusätzlich verschlüsseln und so den Schutz der Vertraulichkeit erhöhen. Für die Authentisierung wird wie beim AH-Protokoll ein Message Authentification Code (MAC) verwendet. Zur Verschlüsselung muss eine symmetrischer Blockchiffre im CBC- Modus verwendet werden.

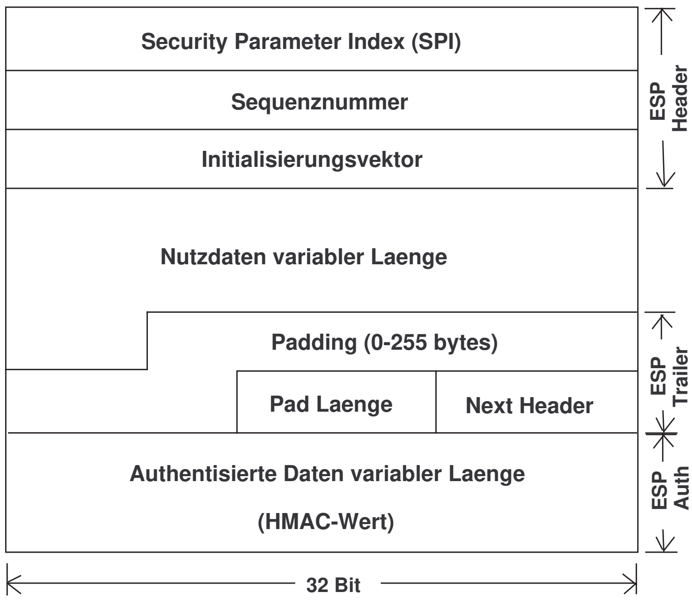

Abbildung 4.4: ESP Paket Format

4.3.1 ESP Transport Mode

Im Transportmodus wird der ESP Header nach dem IP-Header und vor den entsprechenden Nutzdaten angeordnet.

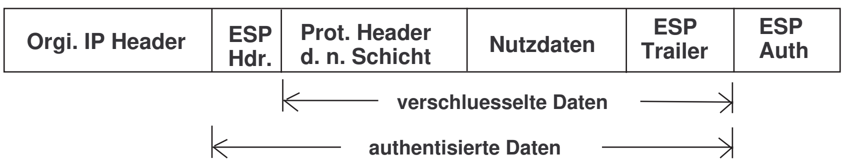

Abbildung 4.5: ESP Protokoll / Transport Modus

4.3.2 ESP Tunnel Mode

Beim Tunnelmodus verpackt ein Sicherheits-Gateway das Paket mit Header und den Nutzdaten in ein neues IP Paket und fügt den ESP Header zwischen dem neuen Header mit der Adresse des Sicherheits-Gateways und dem ursprünglichen IP-Header ein.

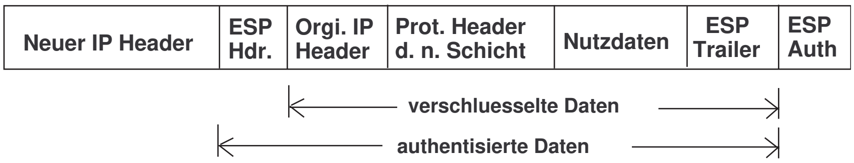

Abbildung 4.6: ESP Protokoll / Tunnel Modus

4.4 SA - Security Association

Das in den beiden Protokollen vorkommende 32-Bit Feld 'Security Parameter Index' (SPI) wird zwischen den Kommunikationspartnern zum Festlegen der Security Association (SA) genutzt. Unter dem Konzept der Security Association versteht man eine Sicherheitsstrategie, die bei den kommunizierenden IPSec-Endsystemen über den Einsatz der anzuwendenden Verschlüsslungsverfahren und Schlüssel informiert. Eine IPSec SA ist dabei unidirektional, das bedeutet bei bidirektionalen Verbindungen eine mögliche Wahl von unterschiedlichen Verfahren und Schlüsseln für die jeweilige Verbindungsrichtung. So gibt es für einen bidirektionalen Datenverkehr also immer mindestens eine eingehende (Inbound SA) und eine ausgehende (Outbound-SA). Für den Transport von Paketen mit unterschiedlichen Sicherheitsstufen, können auch mehrere SAs über eine Verbindung existieren.

4.4.1 SA Bestandteile

Folgende Angaben sind für eine SA möglich:

- **Authentifikationsverfahren, Modus (Transport/Tunnel) und Schlüssel für das AH-Protokoll**
- **Verschlüsselungsverfahren, Modus und Schlüssel für das ESP-Protokoll**
- **Authentifikationsverfahren, Modus und Schlüssel für das ESP-Protokoll**
- **Angaben über die kryptographische Synchronisation oder den Initialisierungsvektor (IV) für das ESP-Protokoll (falls erforderlich)**
- **die Lebensdauer der Schlüssel bzw. der ganzen SA**
- **die IP-Adresse des Endsystems bzw. Subsystems, auf das sich die Vereinbarung der SA bezieht. Falls die SA-Festlegungen für mehrere Empfänger gelten, so kann es auch die Adresse eines Netzes oder Teilnetzes sein**
- **falls mehrere Sicherheitsebenen zwischen den Kommunikationspartnern implementiert sind, enthält die SA auch Sicherheitsklassifikationen der zu schützenden Daten (streng geheim, geheim oder unbestimmt)**

4.4.2 SPI Security Parameter Index

Über den 32-Bit Security Parameter Index in Verbindung mit der IP-Adresse des Kommunikationsendpunktes kann der Empfänger eines IP-Pakets die zugehörige SA ermitteln. Diese SAs werden in den Endgeräten, bzw. Sicherheits-Gateways in der Security Association Database (SAD) zusammengefasst. Werden für eine Verbindung AH und ESP kombiniert, so muss für das jeweilige Protokoll eine eigene SA ausgehandelt werden.

4.5 SPD Security Policy Database

Die SAs sind jedoch nur ein Realisierungskonzept, die eine zugrunde liegende Sicherheitsstrategie umsetzen sollen. Die eigentlichen Sicherheitsspezifikationen sind in der Security Policy Database (SPD) hinterlegt. Ein SPD-Eintrag ist ein Filterkriterium, das auf den gesamten IP-Verkehr anzuwenden ist. Solche Einträge geben für eingehende und ausgehende Pakete an, ob ein IPSec Verfahren angwandt (apply), das Paket vernichtet (discard) oder einfach weitergeleitet (bypass) wird. Die einzelnen Filterwerte des nachfolgenden Beispiels, können durch die Verbindungskonfiguration in der Datei "ipsec.conf"[vgl. 5.3.1] konfiguriert werden.

4.5.1 Beispiel SPD Eintrag

```
src: 192.168.222.200
dst: 192.168.222.10
ipsec-action:   esp req   cipher des3
                          integrity hmacmd5 keylen 128
                          expiry (seconds) 60
                          tunnel
                 ah req   integrity hmacshal keylen 160
                          expiry (seconds) 60
                          transport
```

Abbildung 4.7: SPD Aufbau

Die Strategiefestlegung (Filterregel) bezieht sich auf die Absenderadresse 192.168.222.200 und die Empfangsadresse 192.168.222.10. Es ist das ESP-Protokoll im Tunnelmodus mit dem Tripel DES und CBC Modus zur Verschlüsselung und dem HMAC-MD5 mit dem vollen 128-Bit Hashwert zu verwenden. Jede davon abgeleitete SA hat eine Lebensdauer von 60 Sekunden. Das AH-Protokoll wird zur Authentifizierung im Transportmodus benutzt. HMAC-SHA-1 Verfahren mit einem 160 Bit Hashwert. Es werden AH und ESP gleichzeitig verwendet.

4.5.2 SPD Erzeugung

Falls ein SPD Eintrag mit den Spezifikationen des IP-Pakets übereinstimmt, werden die IPSec Sicherheitsdienste angewendet, und der Filtereintrag verweist auf die zugehörige SA in der SAD. Ein ankommendes Paket, dessen Spezifikationen mit keinem SA Eintrag in der SAD übereinstimmt, wird automatisch vernichtet. Bei einem ausgehenden Paket, für das kein SA Eintrag existiert und gleichzeitig im SPD Eintrag die Anwendung von IPSec gefordert wird, muss eine entsprechende SA dynamisch unter Verwendung des Internet Key Exchange Protokolls (IKE) erzeugt werden.

4.6 Schlüsselaustausch

4.6.1 IKE Protokoll

Das Internet Key Exchange Protokoll (IKE) verwendet das Internet Security Association and Key Management Protocol (ISAKMP), das OAKLEY Protokoll und die ISAKMP Domain of Interpretation, um SAs einzurichten, auszuhandeln, zu modifizieren und wieder löschen zu können. Die erforderlichen Sicherheitsinformationen zum Aufbau der SAs auf der Initiator- sowie auf der Empfänger-Seite werden durch den Strategie-Eintrag in der SPD bestimmt. Welche Chiffre- und Hashverfahren überhaupt verwendet werden können und wie sie miteinander kommunizieren, ist dagegen in der ISAKMP DOI definiert. Das OAKLEY Protokoll ist ein Schlüsselaustauschprotokoll, das als Grundlage des IKE-Protokolls für den authentifizierten Schlüsselaustausch mit Hilfe des Diffie-Hellman Verfahren und des ISAKMP Protokolls sorgt.

4.6.2 IKE-SA

Der Austausch der Sicherheitsinformationen zur Erzeugung von Schlüsseln für den Aufbau der IPSec-SA muss auch abgesichert werden und vertraulich durch den Initiator mittels des IKE Protokolls dem Kommunikationspartner mitgeteilt werden. Die zur Verwendung kommenden kryptographischen und Authentifikations-Verfahren für das IKE Protokoll werden ebenfalls durch Security Associations definiert. Diese IKE-SAs benötigen deshalb auch Sicherheitseigenschaften, die sie aus einer sogenannten IKE Strategie-Datenbank beziehen. Dort werden in einer nach Prioritäten geordneten Liste für den Austausch zu verwendende Verschlüsselungsalgorithmen, Authentifikationsverfahren, Hashingverfahren und benötigte Informationen für das Diffie-Hellman Verfahren gespeichert. Das Diffie-Hellman Verfahren muss nicht explizit in der Strategie Datenbank genannt werden, da es stets für den Austausch der geheimen Schlüssel vom IKE Protokoll verwendet wird. Der Schlüsselaustausch mittels IKE Protokoll wird somit in zwei verschiedenen Phasen vollzogen. In der ersten Phase werden die IKE-SAs ausgehandelt, bevor in der zweiten Phase die sichere und authentisierte Übertragung von Informationen zum Aufbau der IPSec-SAs gestartet werden kann.

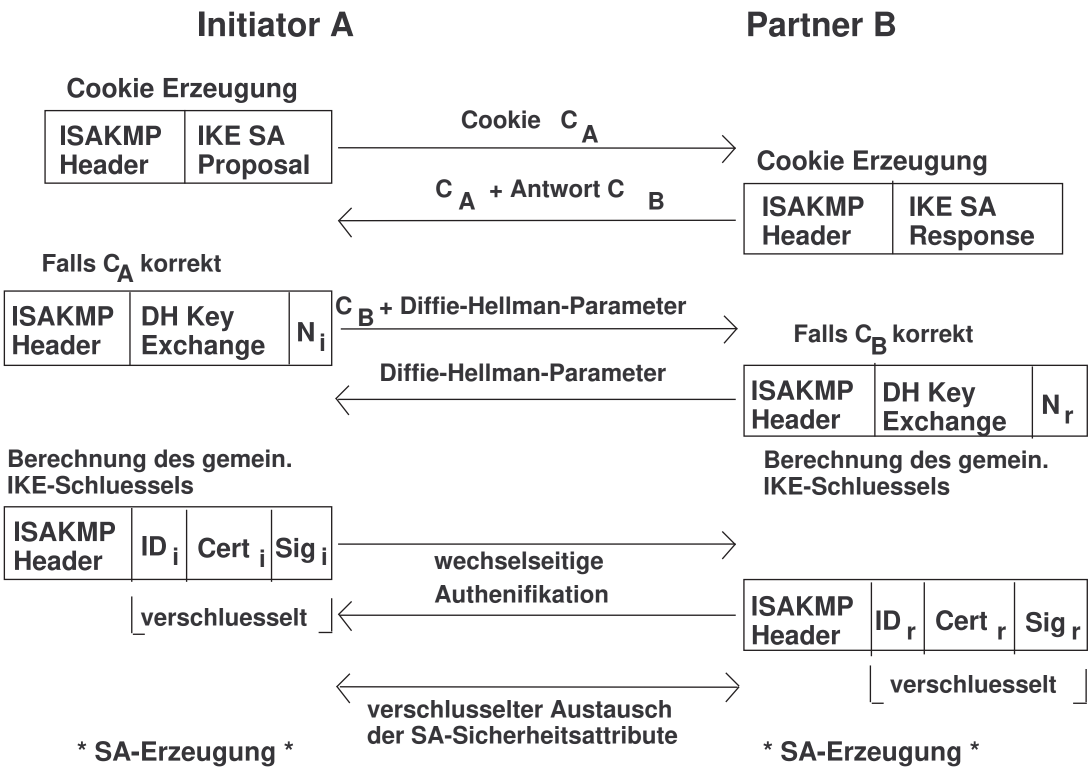

Abbildung 4.8: IKE Verfahren

Erste Phase: Main Mode

In der ersten Phase, genannt Main Mode, werden alle Informationen, aus denen beide Partner den IKE Schlüssel berechnen können, übertragen. Dieser Vorgang beginnt mit der Durchführung eines Handshake

Verfahrens durch den Austausch von Cookies, die dazu dienen, Denial-of-Service Attacken abzuwehren. Ein Cookie besteht dabei aus einer geheimen Information, die der jeweilige Cookie Erzeuger für jedes neu erzeugte Cookie generiert, einem Hashwert über der IP-Adresse des Senders und einem Zeitstempel. Die an dem Cookie Austausch authentisch beteiligten Partner, können den erhaltenen Cookie gezielt rekonstruieren. Zuerst wird ein Cookie vom Initiator an den Partner geschickt, der seinerseits mit einem eigenen Cookie antwortet und zusätzlich den zugeschickten Cookie an den Initiator zurücksendet. Der Initiator überprüft daraufhin den eigenen, vom Kommunikationspartner zurückerhaltenen Cookie auf Korrektheit und schickt, bei einem positiven Ergebnis, den vom Partner erzeugten Cookie an ihn wieder zurück. Falls beide Partner den eigenen Cookie als authentisch bestätigt haben, folgt der nächste Schritt des DH-Verfahrens.

Nun schickt der Initiator ein Paket mit allen zur IKE-Schlüssel Erzeugung benötigten Informationen an den Partner. Dazu gehören die Diffie-Hellman Parameter und eine vom Initiator erzeugte Zufallszahl N_i. Der Partner schickt ebenfalls seine DH Parameter + Zufallszahl N_r an den Initiator zurück. Da jedoch durch das Diffie Hellman Verfahren keine Authentifikation der Kommunikationspartner durchgeführt werden kann, ist im nächsten Schritt eine Authentifizierung der IKE Partner notwendig. Dafür sieht das IKE Protokoll verschiedene Möglichkeiten vor. Zum Beispiel vorab ausgetauschte Master Keys oder digitale Signaturen unter Einsatz von RSA. Für dieses Systementwicklungsprojekt wurde das Verfahren Zertifikate mit digitalen Signaturen verwendet, um eine Einbindung und Authentifizierung von verschiedenen Benutzergruppen zu erreichen. Zur Authentifizierung sendet der Initiator hierfür ein Paket, das bereits entsprechend der SA mit dem gerade ausgehandelten Schlüssel verschlüsselt ist. Es enthält seine Indentifikationsnummer, sein Zertifikat und eine Signatur, also den MD5- oder SHA-1-Hashwert über die Diffie-Hellman Parameter, die Zufallszahl N_i, die öffentlichen Schlüssel der Partner, den ausgetauschten Cookies, der SA und der Identifikationsnummer des Initiators. Dieser Wert wird anschließend mit dem Private Key des Initiators verschlüsselt und an den Partner gesendet. Wichtig dabei ist, das die Identifikationsnummer der IKE-Nachricht und damit die ID des jeweiligen Zertifikatsinhabers mit dem im Zertifikat aufgeführten Distinguished Name übereinstimmt. Der Partner kann nun mit Hilfe des Public Keys und dem erhaltenen Zertifikat die Echtheit der Signatur und damit die Authentizität des Initiators überprüfen. Er sendet dem Initiator daraufhin ebenfalls ein gleichwertiges Paket zu. Die Echtheit des Public Key beweist die digitale Unterschrift des vertrauenswürdigen Root Certification Authority (Root CA), deren Public Key beiden Kommunikationspartnern bekannt ist.

Zweite Phase: Quick Mode

Damit wurde die Authentifizierung durchgeführt und es folgt die zweite, sogenannte Quick Mode Phase. Der Name Quick Mode rührt daher, dass in der ersten Phase bereits Sicherheitsdienste erstellt wurden, die durch die zweite Phase verwendet werden können. Der Nachrichtenaustausch wird somit reduziert und kann schneller durchgeführt werden. Ziel ist der sichere Austausch der zur SA Erzeugung gehörenden Sicherheitsattribute. Diese Attribute werden bei den Kommunikationspartnern zur Festlegung der Algorithmen und Schlüssel verwendet, mit denen später der Verkehr zwischen den Partnern verschlüsselt wird. Zur Erzeugung der zum Einsatz kommenden unterschiedlichen 3 DES Schlüssel, wird aus einem Grundschlüssel über das HMAC-MD5 bzw. HMAC-SHA-1 Verfahren, ein ausreichend langer Schlüssel erzeugt, aus dem dann die 3 einzelnen Schlüssel für den 3 DES Algorithmus extrahiert werden können. Der Grundschlüssel wird durch den Hashwert über Parameter aus dem Diffie-Hellman Verfahren berechnet. Dazu gehören unter anderem die erzeugten Zufallszahlen N_i und N_r. In einer IPSec Session kann die Quick Mode Phase in Verbindung mit der Durchführung eines neuen Diffie-Hellman Verfahrens bei Bedarf neu angestossen werden. So können jedesmal neue Schlüssel für den 3 DES Algorithmus, bzw. für das HMAC Verfahren erzeugt werden. Eine ausführliche Beschreibung der Quick Mode Phase kann unter [LIPP 01] nachgelesen werden.

4.6.3 Diffie-Hellman-Verfahren

Die Besonderheit des Diffie-Hellman Verfahrens zur sicheren Schlüsselvereinbarung ist, dass jeder Kommunikationspartner einen geheimen Session Key selbst berechnet, der nicht über ein unsicheres Medium transportiert werden muss. Für diese dezentrale Berechnung wird ein gegenseitiger Austausch von Informationen zur Schlüsselerzeugung benötigt, der jedoch keine gegenseitige Authentifikation der Kommunikationspartner beinhaltet.

4.7 Sicherheitsmechanismen für Verschlüsselung und Authentifizierung

4.7.1 Hashfunktionen

Hashfunktion werden nicht zur Ver- und Entschlüsseln eingesetzt, sondern dienen zur Kontrolle der Integrität gespeicherter oder über ein unsicheres Medium übertragener Daten. Die berechnete kryptographische Prüfsumme symbolisiert dabei einen sogenannten digitalen Fingerabdruck von einem Datenobjekt, der zusammen mit dem Objekt gespeichert bzw. versandt wird. Dadurch können unautorisierte Modifikationen erkannt werden. Die Hashfunktionen werden oft auch als Einweg-Kompressionsfunktionen bezeichnet, weil sie aus dem langen Eingabewert einen kurzen Ausgabewert fester Größe erzeugen. Sie sollten folgende Sicherheitskriterien erfüllen:

- Aus einem Eingangswert M soll der Hashwert einfach und schnell berechnet werden können.
- Es soll unmöglich sein, aus dem Hashwert den Eingangswert M zu berechnen.
- Es sollte ebenfalls nicht möglich sein, zu einer gegebenen Nachricht M eine Nachricht M’ zu erzeugen, die den selben Hashwert hat.

MD5 - Message Digest 5

Dieses Verfahren wird nicht nur von IPSec verwendet, sondern kommt auch in anderen Protokollen zur Anwendungen. Dazu gehören z.B. CHAP, L2TP und das Routing Protokoll OSPD. MD5 erzeugt aus einem beliebig langen Eingabewert einen 128-Bit-Hashwert. Neben dem Eingabewert selbst geht auch die Länge des Wertes in die Berechnung des Hashwertes ein.

SHA-1 - Secure Hash Algorithm

Der SHA ist ein Verfahren, das einen größeren Hashwert als MD5 erzeugt. Er berechnet einen 160 Bit langen Hashwert aus einem beliebig langen Eingangswert.

HMAC - Hash-based Message Authentication Code

Das HMAC Verfahren enthält keinen eigenen Hashalgorithmus, sondern kombiniert die Verfahren MD5 oder SHA-1 mit symmetrischen Schlüsseln, um ein stärkeres Verfahren zur Authentisierung und Integritätsprüfung zu erhalten. Der Eingabewert der Hasfunktion besteht dabei neben den zu authentisierenden Daten zusätzlich aus einem geheimen Schlüssel. Dieser Schlüssel ist nur den Kommunikationspartnern bekannt, die authentische Nachrichten austauschen möchten. Dadurch kann die kryptographische Prüfsumme dann auch nur von diesen Partnern erzeugt und überprüft werden.

4.7.2 Verschlüsselungsalgorithmen

DES - Data Encryption Standard

Der DES Algorithmus gehört zu den symmetrischen Verschlüsselungsverfahren, das heißt, es wird zur Ver- und Entschlüsselung der gleiche Schlüssel benutzt. Zur Anwendung kommt dabei entweder die Datenblock- oder die Datenstrom-Verschlüsselung. Bei der Datenblock-Verschlüsselung werden jeweils komplette Blöcke einer bestimmten Größe mit einem Schlüssel verschlüsselt. Beim DES ist dieser Datenblock 64 Bit groß. Kleinere Blöcke werden auf die erforderliche Größe aufgefüllt, um verarbeitet werden zu können. Diese Auffüllung ist auch die Schwachstelle der Datenblock-Verschlüsselung und muss dem Sender, wie auch dem Empfänger bekannt sein. Da meistens nur der letzte Block aufgefüllt wird, ist dieses Verfahren für Known-Plain-Text-Angriff anfällig. Deshalb sollten damit niemals sehr kurze Nachrichten verschlüsselt werden. Als Verschlüsselungstechniken werden Bitpermutationen (Transpositionen), Substitutionen und die bitweise Addition modulo 2 vermischt.

3DES - Triple Data Encryption Standard

Der Triple-DES Algorithmus wurde vom DES abgeleitet und ist deshalb zu ihm kompatibel. Der Unterschied liegt hauptsächlich in der Verwendung von drei verschiedenen DES Schlüsseln, mit denen sich der Verschlüsselungsaufwand zwar verdreifacht, die effektive Schlüssellänge sich aber auch vom Standard DES mit 54 Bit auf 108 Bit beim Triple DES erhöht. Damit ergibt sich ein Schlüsselraum von 2 hoch 108 verschiedenen Schlüsseln, der zwar im Moment noch ausreichend ist, auf lange Sicht hin jedoch ebenfalls nicht genügt.

AES - Advanced Encryption Standard

Bei der Suche nach einem Nachfolger des 3DES ist man mit der Einführung des AES (Advanced Encryption Standard) zum Ziel gekommen. Der AES verwendet den Rijndael Verschlüsselungsalgorithmus, der sich gegen eine Vielzahl von anderen vorgeschlagenen Algorithmen wegen seiner Ausgeglichenheit in den vorgeschriebenen Anwendungsgebieten empfohlen hat. Dazu gehört gleichermaßen die Möglichkeit der Soft- und Hardware Implementierung, sowie die Verwendung in so unterschiedlich sicherheitstechnisch anspruchsvollen Anwendungen wie Smartcards oder Hochleistungsrechnern. Die Linux IPSec Software FreeS/WAN unterstützt ab der Version 2 diesen neuen Verschlüsselungs-Standard.

CBC - Cipher Block Chaining

DES und 3DES sind monoalphabetische Algorithmen, das heißt, bei gleichen Schlüssel werden aus gleichen Klartextblöcken auch gleiche Chiffretextblöcke erzeugt. Dies eröffnet die Möglichkeit eines Angriffs mit statischen Methoden, da sich bestimmte Muster im Chiffretext wiederholen. Aus diesem Grund wird der 3DES im Cipher Block Chaining Betriebsmodus verwendet. Dieses Verfahren verhindert bei gleichen Schlüsseln, dass gleiche Klartextblöcke zu gleichen Chiffretextblöcken verarbeitet werden. Der CBC erreicht dies durch eine XOR-Verknüpfung des nächsten zur Verwendung kommenden Klartextblocks mit dem unmittelbar zuvor erzeugten Chiffretextblock. Dadurch ist garantiert, dass sich die Klartextblöcke die dem 3DES Algorithmus als Eingangswert zugeführt werden, von den ursprünglichen Klartextblöcken unterscheiden.

4.8 Angriffsmöglichkeiten bei IPSec

4.8.1 Man-in-the-Middle Attacke

Eine Man-in-the-Middle Attacke kann durch die Verwendung von Zertifikaten beim Verbindungsaufbau verhindert werden, da es einem Angreifer nicht möglich ist, die dabei zur Anwendung kommenden Schlüssel digital zu signieren. Diese und andere in der Literatur aufgeführten Angriffsszenarien können durch den kombinierten Einsatz vom ESP Protokoll zur Verschlüsselung und AH Protokoll zur Authentisierung der Nachricht verhindert werden. Die folgende Tabelle [vgl.[Fack 00]] bezeichnet einige der Angriffe und das Protokoll, mit dem der jeweilige Angriff abgewehrt werden kann:

4.8.2 Andere Angriffsszenarien

Name	Beschreibung des Angriffs	Schutz
Wiretrapping	Netzverkehr abhören, um Informationen zu erhalten	ESP
Spoofing	vortäuschen einer falschen Identität, z.B. verändern der IP-Adresse	AH
ICMP/ARP Angriff	gefälschte Statusmeldungen versenden, um Pakete umzuleiten	AH+ESP
Denial of Service	Überlastung eines Rechners durch SYN-Flooding	AH
TCP Sequenznummer	Manipulation der Sequenznummern bei bestehenden Verbindungen	AH
Replay	Pakete aufzeichnen, um sie später wieder einspielen zu können	AH+ESP

Tabelle 4.1: Verschiedene Angriffsszenarien gegen IPSec

Die Kombination von AH und ESP kann natürlich nur so sicher sein, wie die einzelnen Authentisierungs- und Verschlüsselungs-Verfahren, die von AH und ESP verwendet werden. Einen Überblick geben [LIPP 01][ECK 01][FHW 01][KYAS 98]

4.9 IPSec RFC Standardisierung

RFC 2401: Security Architecture for the Internet Protocol (SA)

RFC 2402: IP Authentification Header (AH)

RFC 2403: The Use of HMAC-MD5-96 within ESP and AH

RFC 2404: The Use of HMAC-SHA-1-96 within ESP and AH

RFC 2405: The ESP DES-CBC Cipher Algorithm With Explicit IV

RFC 2406: IP Enscapulating Security Protocol (ESP)

RFC 2407: The Internet IP Security Domain of Interpretation for ISAKMP

RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP)

RFC 2409: The Internet Key Exchange (IKE)

RFC 2412: The OAKLEY Key Determination Protocol

Vollständige Übersicht unter [IETF 02]

Kapitel 5

Implementierung

Auf dem Linux Router, der mit dem Kernel 2.4 betrieben wird, kommt die IPSec Software FreeS/WAN in der neuesten Version zum Einsatz. Dafür wurde der für IPSec notwendige KLIPS (KerneL IPSec Support) im Kernel aktiviert. Eine Firewall kontrolliert den Verkehr ins interne und externe Netz. Für die am VPN teilnehmenden Clients wurden verschiedene Betriebssysteme erfolgreich getestet. Zu ihnen gehören Linux, Windows 2000 und Windows XP. Auf den Linux-Client Systemen kann ebenfalls die FreeS/WAN Software eingesetzt werden, den Windows 2000 und XP Systemen wird zur Konfiguration der IPSec Umgebung das freeware IPSec Tool von Marcus Müller vorgeschlagen. Mit ihm können die notwendigen Einstellungen der IPSec-Filter und -Regeln des Systems leicht vorgenommen werden. Kommerzielle IPSec Client Systeme wie beispielsweise SSH-Sentinel und PGP-Net sind auch möglich, wurden jedoch in diesem Projekt nicht explizit getestet. Zur IPSec basierten Authentifizierung verschiedener Benutzergruppen werden Zertifikate verwendet, die mit OpenSSL unter Linux erzeugt wurden.

5.1 Installation von FreeS/WAN

Zuerst müssen die Kernel Quellen an die übliche Stelle installiert werden:
`/usr/src/linux`

Dann über den ”FreeS/WAN Download”link auf http://www.freeswan.org das source-file (z.B.: `freeswan-1.99.tar.gz`) des Programms laden und in das nachfolgende Verzeichnis kopieren und das tar-file entpacken:
`/usr/src/`

Zum Zeitpunkt dieser Ausarbeitung ist die X.509 Zertifikat-Unterstützung noch nicht im FreeS/WAN Programmpaket enthalten. Es gibt jedoch ein Schweizer Unternehmen, das in Zusammenarbeit mit der ETH Zürich einen patch für die Zertifikat-Unterstützung in FreeS/WAN anbietet. Dieser patch kann ebenfalls in das FreeS/WAN Quellverzeichnis kopiert und entpackt werden. http://www.strongsec.com/freeswan

Um den patch einspielen zu können, muss das file freeswan.diff vom X509 Verzeichnis in das FreeS/WAN Programmverzeichnis kopiert werden. Danach den patch mittels
`# patch -p1 < freeswan.diff`
anwenden. Nach dieser Modifikation muss der Linux Kernel IPSec tauglich gemacht werden. Dies geschieht beispielsweise mit dem Befehl:
`# make menugo`

Mit ihm werden die Linux Kernel Source files gepatched und zusätzlich das Kernel Konfigurationsprogramm aufgerufen. Unter dem Menüpunkt `<Networking Options>` befindet sich die Auswahl `<IP Security Protocol (Free S/WAN IPSec)>`. Da hier bereits alle nötigen Optionen aktiviert sind, sollte man sich nun noch um die Frage einer festen Integration oder einer modularen Einbindung von IPSec in den Kernel kümmern, bevor man mit der Speicherung der Konfiguration die automatische Kompilierung des neuen Kernels anstößt. Nachdem nun der KerneL IPSec Support (KLIPS) aktiviert ist, können der Kernel und die Kernel Module mit dem nachfolgenden Befehl installiert werden:

```
# make kinstall
```

Dieser Vorgang hat auf einem 800 MHz Pentium in Verbindung mit dem 2.4.18 Kernel etwa 1 Stunde gedauert. Danach muss der alte Kernel im /boot Verzeichnis entweder durch den neuen Kernel ersetzt werden oder man ändert den Pfad im LILO Bootloader entsprechend ab. Nach einem Neustart des Systems sollte bei der Eingabe des Befehls:

```
# ipsec --version
```

die Versionsnummer der FreeS/WAN Software gefolgt von der Version des X.509 patches zu sehen sein:

```
Linux FreeS/WAN 2.00pre0
See `ipsec --copyright' for copyright information.
X.509-1.0.1 distributed by Andreas Steffen <andreas.steffen@strongsec.com>
```

5.2 X.509 Zertifikate erzeugen

Als nächstes folgt die Erzeugung der Zertifikate. Hierzu muss die Open Source Software OpenSSL (http://www.openssl.org) installiert sein. Eine Hilfestellung bei der Erzeugung von Zertifikaten bietet das OpenSSL Handbuch des DFN [DFN 02] und ein Artikel vom Programmierer des X.509 Patches für FreeS/WAN[Stef 02]. Zuerst wird ein root Zertifikat erzeugt, mit dem alle anderen Zertifikate unterschrieben werden und das somit als Basis für die Certification Authority (CA) dient. Es ist allerdings ratsam, davor die Konfigurationsdatei openssl.cnf etwas genauer zu betrachten. In ihr können Einstellungen, wie das Verzeichnis in dem OpenSSL die erzeugten Schlüssel und Zertifikate ablegt, bestimmt werden. Auch Voreinstellungen und Vorbelegungen für die einzelnen Elemente des Distinguished Name (DN), sowie die Gültigkeitsdauer der Zertifikate sind möglich. Dies kann bei der Erzeugung der Client Zertifikate enorm viel Zeit sparen, da statische Werte nicht jedes Mal neu eingegeben werden müssen.

```
[ CA_default ]

dir            = ./                  # Where everything is kept
certs          = $dir/certs          # Where the issued certs are kept
crl_dir        = $dir/crl            # Where the issued crl are kept
database       = $dir/index.txt      # database index file.
new_certs_dir  = $dir/newcerts       # default place for new certs.

certificate    = $dir/cacert.pem     # The CA certificate
serial         = $dir/serial         # The current serial number
crl            = $dir/crl.pem        # The current CRL
private_key    = $dir/private/cakey.pem # The private key
RANDFILE       = $dir/private/.rand  # priv. random number file
```

Abbildung 5.1: openssl.conf Einstellungsmöglichkeiten

5.2.1 CA - Certificate Authority

Zuerst wird ein 2048 Bit langer RSA Private Key cakey.pem erzeugt und durch den 3DES Algorithmus unter Eingabe eines Passworts gegen das unautorisierte Signieren von Userzertifikaten geschützt. Danach mit Hilfe des Private Keys das selbst unterzeichnete Root CA-Zertifikat, cacert.pem, mit einer Gültigkeitsdauer von 4 Jahren erzeugen. Während dieses Vorgangs werden die Elemente des Distinguished Name abgefragt.

```
# openssl req -x509 -days 1460 -newkey rsa:2048 -keyout
private/cakey.pem -out cacert.pem
```

5.2.2 Server Zertifikat

Nun kommt das Zertifikat für den IPSec Router an die Reihe. Zuerst wird der Private Key und dann ein Antrag auf ein Zertifikat erzeugt, bevor es im nächsten Schritt von der CA unterschrieben wird. Die Gültigkeit des Zertifikats beträgt 3 Jahre.

```
# openssl req -newkey rsa:2048 -keyout serverKey.pem -out
serverReq.pem
# touch index.txt; echo 01 >> serial;
# openssl ca -policy policy_anything -in serverReq.pem -days 1095 -out
serverCert.pem -notext
```

Erklärung zur Datei index.txt folgt im Abschnitt Certificate Revocation List. Aus historischen Gründen muss das Zertifikat des Linux-Gateways unter dem Dateinamen /etc/x509cert.der auch in binärer Form vorliegen. Der nachfolgende Befehl sorgt dafür:

```
# openssl x509 -in gatewayCert.pem -outform der -out /etc/x509cert.der
```

5.2.3 Client Zertifikate

Auch hier wird zuerst der Private Key und dann der Antrag auf das Zertifikat erzeugt. Danach wird es ebenfalls von der CA unterschrieben. Wichtig ist, dass die Gültigkeit des Client Zertifikats kürzer ist, als die Gültigkeit des Serverzertifikats und beide wiederum eine kürzere Lebensdauer als das Root CA Zertifikat haben.

```
# openssl req -newkey rsa:2048 -keyout clientKey.pem -out
clientReq.pem
# openssl ca -policy policy_anything -in clientReq.pem -days 1094 -out
clientCert.pem -notext
```

Um die Zertifikate auch für andere IPSec Clients zur Verfügung zu stellen, werden die erstellten Schlüssel und Zertifikate im PKCS#12 Format in einer eigenen Datei angeboten. Dazu packt die ausstellende CA das Client Zertifikat, den Client Private Key sowie das Root-Ca Zertifikat in eine Datei mit der Dateiendung p12. Unter Windows kann dieses Format in die dafür vorgesehene Schlüsselverwaltung importiert werden [siehe 5.8.1]. Der nachfolgende Befehl bewerkstelligt dies:

```
# openssl pkcs12 -export -inkey clientKey.pem -in clientCert.pem
-certfile cacert.pem -out clientCert.p12
```

Das bei der Erzeugung der Datei abgefragt Export Passwort wird später beim Import in die Windows IPSec Umgebung benötigt und darf deshalb nicht vergessen werden.

5.2.4 CRL - Certificate Revocation List

Die Certification Authority erstellt in einem periodischen Abstand eine neue unterschriebene Certification Revocation List(CRL), in der gültige und gesperrte Zertifikate aufgeführt werden. Auf diese CRL greift der IPSec Gateway beim Verbindungsaufbau zu und kann kompromittierte Zertifikate bei der Authentifizierung des Kommunikationspartners ablehnen.

Mit der Erzeugung eines Zertifikats wird seine Seriennummer und sein CommonName (CN) in der Datei index.txt gespeichert. Bei der Erneuerung der CRL werden diese Daten aus der index Datei gelesen. Möchte man nun ein Zertifikat widerrufen, so muss es in dieser Datei als gesperrt markiert werden. Die Gültigkeit der CRL und damit der Zeitpunkt der Erneuerung wird in der Datei opsenssl.cnf definiert. Als default ist ein Wert von 30 Tagen eingetragen, es kann jedoch auch eine stündliche Erzeugung gewählt werden. Ist eine sofortige Sperrung notwendig, muss dies von Hand angestoßen werden. Der Befehl

```
# openssl ca -revoke mustermannCert.pem
```

markiert das Zertifikat in der index.txt Datei als gesperrt und der Aufruf

```
# openssl ca -gencrl -out crl.pem
```

in der Kommandozeile erzeugt die aktualisierte Certification Revocation List. Zum Anzeigen aller gesperrten Zertifikate kann der nachfolgende Befehl eingegeben werden.

```
# openssl crl -in crl.pem -noout -text
```

5.2.5 Überblick Zertifikate

Einen Überblick über die Lage der angesprochenen Dateien auf dem Linux IPSec Gateway gibt die folgende Abbildung. Die X.509 Zertifikate können mit einem Simple Certificate Enrollment Protocol (SCEP) tauglichen Client von der ausstellenden CA bezogen werden, falls die CA das SCEP Protokoll zur Verteilung der Zertifikate unterstützt. Alternativ müssen die Zertifikate von Hand installiert werden.

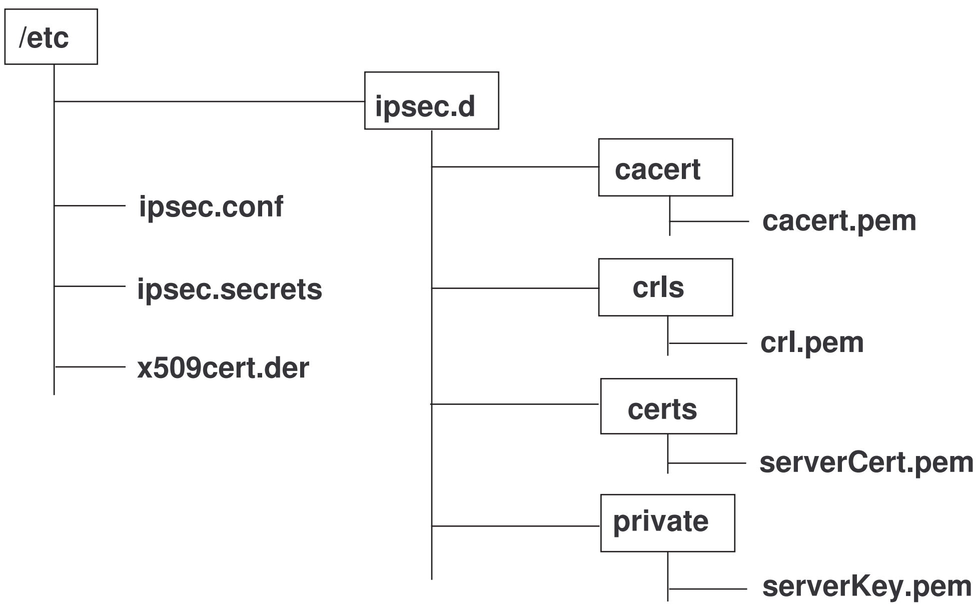

Abbildung 5.2: wichtige IPSec Verzeichnisse

Bei der Verwendung von Zertifikaten zur Authentifizierung muss die Datei ipsec.secrets den folgenden Inhalt haben: `: RSA serverKey.pem "GeheimesPasswort"` wobei serverKey.pem den Private Key im Verzeichnis /etc/ipsec.d/private bezeichnet und der String "GeheimesPasswort"das Passwort nennt, mit dem der verschlüsselte Private Key bei der Erzeugung durch den 3DES Algorithmus geschützt wurde. Die Datei ipsec.secret sollte deshalb auf jeden Fall vor nichtautorisiertem Zugriff geschützt werden.

5.3 IPSec Router Konfiguration

5.3.1 FreeS/WAN konfigurieren

Die Datei ipsec.conf ist die zentrale Verbindungs-Konfigurationsdatei der Linux FreeS/WAN Umgebung. In ihr können Debugging Optionen aktiviert und Einzelverbindungen getrennt voneinander konfiguriert werden. Die in diesem Projekt auf dem Router zum Einsatz kommende ipsec.conf hat folgendes Aussehen:

```
config setup
	# %defaultroute is okay for most simple cases.
	forwardcontrol=yes
	interfaces=%defaultroute
	#interfaces="ipsec0=eth0"
	# Debug-logging controls:  "none" for (almost) none, "all" for lots.
	klipsdebug=none
	plutodebug=control
	# Use auto= parameters in conn descriptions to control startup
	plutoload=%search
	plutostart=%search
	# Close down old connection when new one using same ID shows up.
	uniqueids=yes

conn %default
	keylife=1h
	keyingtries=5
	auth=ah
	authby=rsasig
	leftrsasigkey=%cert
	rightrsasigkey=%cert
	rightcert=serverCert.pem
	right=%defaultroute
	rightid="C=DE, ST=Bayern, L=Muenchen, O=LMU, OU=Kommunikationssysteme,
CN=projekt0.nm.informatik.uni-muenchen.de/Email=krause@in.tum.de"
	pfs=yes
	auto=add

conn roadwarrior-mitarbeiter
	rightsubnet=192.168.222.0/0
	left=%any
	leftsubnet=192.168.222.192/26
	leftid="C=DE, CN=Mitarbeiter"

conn roadwarrior-student
	rightsubnet=0.0.0.0/0
	left=%any
	leftid="C=DE, CN=Student"
```

Abbildung 5.3: Router ipsec.conf

Im Abschnitt **config setup** können Einstellungen zum PLUTO/KLIPS Debugging und zum virtuellen und physikalischen Interface getätigt werden. Der PLUTO Daemon ist ein Prozess, der den Schlüsselaustausch mit dem Internet Key Exchange(IKE) Protokoll steuert.

Der Abschnitt - **conn default** beinhaltet Einstellungsmöglichkeiten, die grundsätzlich für alle Verbindungen gelten.

Dann folgen die Abschnitte mit den unterschiedlichen Verbindungsparametern für die jeweiligen Einzelverbindungen. Diese Parameter sind in der FreeS/WAN Dokumentation[FSWAN 02] genau beschrieben. Die für das Projekt interessanten Einstellungsmöglichkeiten, besonders die zur Verwendung von Zertifikaten, werden nun im einzelnen erklärt.

Abschnitt - **conn default:**

`Keylife=1h`
Die Dauer der Gültigkeit eines session keys und SA (Angabe möglich in Sekunden,Minuten,Stunden)

`Keyingtries=5`
Wie oft der Verbindungsaufbau versucht wird (0 steht für unendlich)

`Auth=ah`
Mit welchem Protokoll die Authentizität des Datenverkehrs gewährleistet wird (das Authentification Header (AH) Protokoll)

`Authby=rsasig`
Den Typ der Authentifizierung der IPSec Gateways (entweder shared secrets oder es wird eine mit RSA erzeugte digitale Signatur verwendet)

`rightrsasigkey=cert`
Wie sich der IPSec Router authentifiziert (es wird in dem Beispiel ein Zertifikat gewählt, möglich wäre es auch, einen RSA key direkt anzugeben)

`rightid="C=DE, ST=Bayern, L=Muenchen, O=LMU, OU=Kommunikationssysteme, CN=projekt0.nm.informatik.uni-muenchen.de/Email=krause@in.tum.de"`
Der komplette Distinguished Name des vom IPSec Router eingesetzten Zertifikats.

`pfs=yes`
Das Diffie-Hellman Verfahren zum Schlüsselaustausch soll im Quick Mode erneut durchgeführt werden. [vgl. 4.6.2]

`auto=add`
Die Verbindungsaufbau wird vom Kommunikationspartner gestartet.

Parameter, die sich bei den verschiedenen Kommunikationspartnern ändern, werden in den jeweiligen Abschnitten der Einzelverbindungen aufgeführt:

Abschnitt - **conn roadwarrior-mitarbeiter**

`Left=any`
Diese Verbindungsbeschreibung ist für angehörige des Lehrstuhls gedacht, die auf interne Ressourcen zugreifen möchten. Es sind alle Adressen zum Verbindungsaufbau zugelassen:

`rightsubnet=192.168.222.0/0`
Hiermit wird die Tunnelbeschränkung [vgl. 3.3.2] für die Pakete aus dem Lehrstuhlnetz gesetzt. Es werden alle Pakete vom Router zum Remote-Rechner durch den Tunnel geleitet. Erkennbar an der /0. Dies bedeutet, dass der komplette Netzteil der Adresse frei wählbar ist. Eine /32 würde dagegen eine einzelne Hostadresse bezeichnen, von der aus die Pakete dann stammen müssten, um den Tunnel durchqueren zu können. Normalerweise wird ein Adressbereich gewählt, der mit dem am IPSec Gateway angeschlossene Sunbnetz übereinstimmt, so dass alle Pakete aus diesem Netz durch den Tunnel gesendet werden können.

`leftsubnet=192.168.222.192/26`
Mit der folgenden Einstellung werden nur Pakete vom Kommunikationspartner in Richtung IPSec Router durch den Tunnel geroutet, die eine Ursprungs-Adresse aus dem angegebenen Netzbereich haben. Dieser angegebene Netzbereich ist ein Teilbereich des internen Lehrstuhlnetzes und ist für den Remote-Zugriff[3.3.2] auf interne Dienste zwingend notwendig.

`leftid="C=DE, CN=Mitarbeiter"`
Die vom Kommunikationspartner zur Authentifikation verwendeten Zertifikate müssen einen bestimmten Distinguished Name haben. Über diese ID wird auch die Vergabe der internen Adressen gesteuert. Hat der Kommunikationspartner ein Zertifikat, welches nicht mit dem Distinguished Name übereinstimmt, so kann er auch nicht erfolgreich authentifiziert werden.

Abschnitt - **conn roadwarrior-student**

`Left=any`
Die IP-Adresse des Kommunikationspartners ist nicht beschränkt. Es dürfen alle Adressen verwendet werden, vorausgesetzt wird jedoch ein gültiges Zertifikat.

`leftid="C=DE, CN=Student"`
Eindeutiger Distinguished Name für die Benutzergruppe Student

`Rightsubnet=0.0.0.0/0`
Es soll der komplette Verkehr vom IPSec Router in Richtung Kommunikationspartner durch den IPSec Tunnel geleitet werden. Alle Adressebereiche werden freigegeben.

Hier fehlt die Angabe eines **Leftsubnet** Bereichs, dass bedeutet, Pakete vom mobilen Endgerät werden nur durch den Tunnel geschleust, falls sie als Ursprungs-Adresse die gleiche Adresse haben, mit der sich das mobile Endgerät beim Verbindungsaufbau(Left=...) angemeldet hat. Eine selbst zugewiesene private Adresse könnte nicht verwendet werden.

5.4 IPRoute2 Einsatz

Wie oben in der Projektbeschreibung [vgl. 3.2.1] gezeigt, wird mit Hilfe von IPRoute2 ein Policy Routing durchgeführt, mit dem einem Remote-Rechner eine Adresse aus dem internen Subnetz zugewiesen wird. Dieser ausgegliederte Netzbereich wird auch als Extruded Subnet bezeichnet. Wenn man auf dem Client die zugewiesene Adresse verwenden möchte, muss mit IPRoute2 ein virtuelles Interface eth0:1 auf dem von IPSec verwendeten externen physikalischen Interface eth0 erzeugt werden. Dann werden durch IPRoute2 die Pakete von diesem virtuellen eth0:1 Interface aus an das IPSec Interface ipsec0 geleitet und von dort zum IPSec Router getunnelt.

5.4.1 Client IPRoute2 Setup

1. `ip rule add iif lo table projekt2.host priority 500`
2. `ip addr add 192.168.222.200/32 brd 141.84.218.255 dev eth0 label eth0:1`
3. `ip route add default table projekt2.host dev ipsec0 src 192.168.222.200`
4. `ip route flush cache`

1. Zeile: Erzeugt einen IPRoute2 Table mit dem Namen projekt2.host, der durch den Priority Wert vor dem Standard Routing Table zum Einsatz kommt. Der Parameter ”iif lo”bewirkt, dass diejenigen Pakete von der Regel behandelt werden, die den Ursprung auf diesem Host haben.
2. Zeile: Es wird auf dem eth0 Interface ein virtuelles Interface mit dem Namen eth0:1 und der Adresse 192.168.222.200 erzeugt. Die broadcast Adresse wird vom physikalischen Interface übernommen.
3. Zeile: Als nächstes wird eine default route erzeugt, die den Verkehr vom virtuellen Interface eth0:1 zum ipsec0 Interface leitet. Damit bekommen alle Pakete die Absenderadresse des eth0:1 Interface.
4. Zeile: Mit diesem Befehl werden die IPRoute2 Einträge sofort aktiviert

5.4.2 Server IPRoute2 Setup

Der Einsatz von IPRoute2 auf dem IPSec Router ist nicht zwingend. Die oben genannten Möglichkeiten durch das Policy Routing sprechen für den Einsatz. So könnten bei einer zu starken Beanspruchung des Routers bestimmte Protokolle an der IPSec Verschlüsselung vorbeigeführt werden und damit der Router entlastet werden. Zu Testzwecken kann der erzeugte IPRoute2 Table leicht wieder deaktiviert werden. Dadurch werden die Eigenschaften des Standard Routing Table wieder aktiv. Eine weitere Punkt, der für den Einsatz spricht und damit die Sicherheit des IPSec Systems erhöht, ist die durch das Policy Routing ermöglichte Kombination aus Firewall und IPRoute2. Die folgenden Befehle auf dem IPSec Router sorgen für das korrekt Routing der Pakete:

IPRoute2 für den IPSec Router:

1. `ip rule add iif lo table projekt0.host priority 500`
2. `ip route add default via 141.84.218.254 table projekt0.host dev eth0`
3. `ip route add 192.168.222.0/25 via 192.168.222.126 table projekt0.host dev eth1`
4. `ip route add 192.168.222.128/26 via 192.168.222.126 table projekt0.host dev eth1`
5. `ip route add 192.168.222.192/26 table projekt0.host dev ipsec0`
6. `ip route flush cache`

1. Zeile: Diese Zeile erzeugt einen IPRoute2 Table mit dem Namen projekt0.host, der durch den Priority Wert vor dem Standard Routing Table zum Einsatz kommt. Der Parameter ”iif lo”bewirkt, dass diejenigen Pakete von der Regel behandelt werden, die den Ursprung auf diesem Host haben.
2. Zeile: Es wird eine default Route zum Standard Gateway eingerichtet
3. Zeile: Pakete die an einen Rechner im internen Netz adressiert sind werden über das interne Interface dahin geleitet.
4. Zeile: Pakete die dem anderen Adressbereich des internen Netzes angehören werden ebenfalls über das interne eth1 interafce geleitet.
5. Zeile: Pakete die an einen Rechner im externen Subnetz adressiert sind werden über das virtuelle ipsec0 Interface in den Tunnel geleitet.
6. Zeile: Mit diesem Befehl werden die IPRoute2 Einträge sofort aktiviert

5.5 Einstellungen zur Firewall

Es kommt eine Linux iptable Firewall vom Typ SuSEfirewall2 zum Einsatz. Sie sollte alle Regeln enthalten, die das interne Netz vor der Installation des IPSec Routers geschützt haben. Dazu muss nun jedoch auf der Seite zum externen Netz der Port UDP 500 geöffnet werden und die IP Protokolle 50 und 51 durchgelassen werden. Der Port UDP 500 wird vom IKE Protokoll zum Schlüsselaustausch verwendet. Das zur Authentifizierung der Daten notwendige AH-Protkoll hat die Protokollnummer 51 und das zur Verschlüsselung der Daten notwendige ESP die Nummer 50. Diese und andere Einstellungen, zum Beispiel die Belegung des internen und externen, sowie des virtuellen ipsec Interface, können sehr komfortabel im Konfigurationsfile der SuSEfirewall2 eingestellt werden. Besonderer Wert muss auch auf das Interface zum internen Netz gelegt werden, da die verschlüsselten Daten über das Protokoll 50 oder 51 ungefiltert durch den IPSec Tunnel über das externe Interface wandern und dort nicht kontrolliert werden können. Der Verkehr kann also frühestens nach dem Entpacken der Pakete am virtuellen ipsec0 Interface durch die Firewall beschränkt werden.

Manuelle iptables Einstellung:

Schlüsselaustausch über IKE Protokoll / UDP Port 500:

```
# iptables -A INPUT -p udp --sport 500 --dport 500 -j ACCEPT
# iptables -A OUTPUT -p udp --sport 500 --dport 500 -j ACCEPT
```

ESP Protokoll erlauben / Protokollnummer 50

```
# iptables -A INPUT -p 50 -j ACCEPT
# iptables -A OUTPUT -p 50 -j ACCEPT
```

AH Protokoll erlauben / Protokollnummer 51

```
# iptables -A INPUT -p 51 -j ACCEPT
# iptables -A OUTPUT -p 51 -j ACCEPT
```

5.6 Linux IPSec Client Software Installation für Mitarbeiter

FreeS/WAN Installation unterscheidet sich von der FreeS/WAN IPSec Router Installation nur in der ipsec.conf Datei und den Zertifikaten. Siehe: [5.1]

Die ipsec.conf Datei für den Linux Client hat den folgenden Aufbau:

```
config setup
        # THIS SETTING MUST BE CORRECT or almost nothing will work;
        # %defaultroute is okay for most simple cases.
        forwardcontrol=yes
        interfaces=%defaultroute
        # Debug-logging controls:  "none" for (almost) none, "all" for lots.
        klipsdebug=none
        plutodebug=none
        # Use auto= parameters in conn descriptions to control startup
        plutoload=%search
        plutostart=%search
        # Close down old connection when new one using same ID shows up.
        uniqueids=yes

conn   roadwarrior-mitarbeiter
        keylife=1h
        keyingtries=5
        auth=ah
        authby=rsasig
        leftrsasigkey=%cert
        left=%defaultroute
        leftsubnet=192.168.222.192/26
        leftcert=mitarbeiterCert.pem
        leftid="C=DE, CN=Mitarbeiter"
        rightrsasigkey=%cert
        right=141.84.218.150
        rightsubnet=192.168.222.0/0
        rightid="C=DE, ST=Bayern, L=Muenchen, O=LMU,
OU=Kommunikationssysteme,
CN=projekt0.nm.informatik.uni-muenchen.de/Email=krause@in.tum.de"
        pfs=yes
        auto=start
```

Abbildung 5.4: Linux ipsec.conf für Mitarbeiter

Der Unterschied zu der nachfolgenden `ipsec.conf` für Studenten sind die drei Zeilen:

```
leftsubnet=192.168.222.192/26
leftcert=mitarbeiterCert.pem
leftid="C=DE, CN=Mitarbeiter"
```

Die Funktion der Zeilen wird im Abschnitt [5.3.1] erklärt.

5.7 Linux IPSec Client Software Installation für Studenten

Es wird ebenfalls FreeS/WAN verwendet. Der Vorgang läuft genau gleich ab, wie bei der FreeS/WAN IPSec Router Installation. Siehe: [5.1]

Die ipsec.conf Datei für den Linux Client hat den folgenden Aufbau:

```
config setup
        # THIS SETTING MUST BE CORRECT or almost nothing will work;
        # %defaultroute is okay for most simple cases.
        forwardcontrol=yes
        interfaces=%defaultroute
        # Debug-logging controls:  "none" for (almost) none, "all" for lots.
        klipsdebug=none
        plutodebug=none
        # Use auto= parameters in conn descriptions to control startup
        plutoload=%search
        plutostart=%search
        # Close down old connection when new one using same ID shows up.
        uniqueids=yes

conn   roadwarrior-student
        keylife=1h
        keyingtries=5
        auth=ah
        authby=rsasig
        leftrsasigkey=%cert
        left=%defaultroute
        leftcert=studentenCert.pem
        leftid="C=DE, CN=Student"
        rightrsasigkey=%cert
        right=141.84.218.150
        rightsubnet=192.168.222.0/0
        rightid="C=DE, ST=Bayern, L=Muenchen, O=LMU,
OU=Kommunikationssysteme,
CN=projekt0.nm.informatik.uni-muenchen.de/Email=krause@in.tum.de"
        pfs=yes
        auto=start
```

Abbildung 5.5: Linux ipsec.conf für Studenten

5.8 Windows 2000 IPSec Client Software Installation für Studenten

Für eine kostenlose und möglichst einfache IPSec Verbindungs-Konfiguration unter Windows 2000 kann das Freeware VPN-Tool von Marcus Müller verwendet werden. Zwar können Regeln und IP Filter zur Steuerung des IPSec Verkehrs auch in der von Microsoft zur Verfügung gestellten Management-Konsole (MMC) selbst erzeugt werden, dies ist aber mit einem so hohen Aufwand verbunden, dass sich eine Vielzahl von Falscheinstellungen einschleichen könnten. Deshalb wird für dieses Projekt die Verwendung des VPN Tools empfohlen. Es benutzt die von Microsoft zur Verfügung gestellten Programme ipsecpol.exe (Windows2000) oder ipseccmd.exe (Windows XP) um die notwendigen Einstellungen mit Hilfe einer an FreeS/WAN orientierten ipsec.conf Konfigurationsdatei durchzuführen.

Für die Installation werden die folgenden Pakete benötigt:

- `Windows 2000 Service Pack 2`
  - `http://www.microsoft.com/windows2000/downloads/servicepacks /sp2/sp2lang.asp`

- `Windows 2000 ipsecpol.exe Tool Version 1.22`
  - `http://agent.microsoft.com/windows2000/techinfo/reskit/tools /existing/ipsecpol-o.asp`
- `Windows 2000 VPN Tool von Marcus Müller`
  - `http://vpn.ebootis.de/package.zip`

Das Windows 2000 Service Pack 2 enthält das High Encryption Package für die von FreeS/WAN benötigte 3DES Verschlüsselung. Das IpsecPol Tool von Microsoft ist im Windows 2000 Ressource Kit enthalten. Im von Marcus Müller bereit gestellten zip file befindet sich neben einem exemplarischen ipsec.conf file, dem eigentlichen IPSec Setup Werkzeug, auch noch ein Plugin für Microsofts-Management-Console (MMC).

Zuerst die Service Pack 2 Installation durchführen, dann das Windows 2000 ipsecpol Programm in ein beliebiges Verzeichnis installieren. Dieses Verzeichnis wird im weiteren Verlauf der Ausarbeitung IPSec-Verzeichnis genannt. Daraufhin das VPN Tool ebenfalls in das IPSec-Verzeichnis entpacken. Im nächsten Schritt sollte das vorher erzeugt Client Zertifikat für die Windows Umgebung mit Hilfe des MMC Plugins in die IPSec Umgebung integriert werden. Für den Transport vom IPSec Router bzw. dem Aussteller des Zertifikats zum Client muss dabei ein sicherer Weg gewählt werden. Eine unverschlüsselte Übertragung per email oder FTP widerspricht selbstverständlich den Sicherheitsanforderungen. Eine sichere, wie auch einfache Übertragung kann mit Hilfe einer Diskette durchgeführt werden. Bei VPN-IPSec Systemen mit einer hohen Benutzerzahl wäre der daraus resultierende Administrations-Aufwand natürlich riesig. Hier müsste über ein anderes Übertragungsverfahren zum Austausch der Zertifikate nachgedacht werden.

5.8.1 Zertifikate importieren

Nun also das MMC Plugin starten, um das Zertifikat zu importieren. Die folgenden Schritte beschreiben den dafür notwendigen Vorgang[Carl 02]:

Menüpunkt-Konsole:

1. `'Konsole' - 'Snap-In hinzufügen/entfernen'`
2. `Auf 'Hinzufügen' klicken`
3. `'Zertifikate' auswählen und 'Hinzufügen' drücken`
4. `'Computerkonto' auswählen und 'Weiter' klicken`
5. `'Lokalen Computer' auswählen und 'Fertig stellen' klicken`
6. `'IP-Sicherheitsrichtlinien auf lokalem Computer' auswählen und 'Hinzufügen' drücken`
7. `'Computerverwaltung' auswählen und 'Hinzufügen' klicken`
8. `'Lokalen Computer' auswählen und 'Fertig stellen' klicken`
9. `'Schließen' und 'OK' drücken`

Im Konsolenstamm:

1. `Das Pluszeichen bei 'Zertifikate (Lokaler Computer)' drücken`
2. `Rechte Maustaste auf 'Eigene Zertifikate', 'ALL Tasks' wählen und dann 'Importieren' klicken`
3. `auf 'Weiter' klicken`

4. Pfad zur .p12 Datei eingeben oder 'Durchsuchen' klicken und die Datei auswählen. 'Weiter' drücken
5. Das Passwort eingeben, mit dem das Zertifikat bei der Herstellung geschützt wurde (Export Passwort). Dann 'Weiter' klicken
6. 'Zertifikatspeicher automatisch auswählen' markieren und 'Weiter' klicken
7. 'Fertig stellen' klicken und 'OK' drücken

Danach die Veränderungen abspeichern und das Programm beenden. Nun noch die Verbindungskonfiguration in ipsec.conf vornehmen:

```
conn    roadwarrior-student
        auth=ah
        left=%any
        right=141.84.218.150
        rightsubnet=*
        rightrsasigkey=%cert
        rightca="C=DE, S=Bayern, L=Muenchen, O=LMU,
OU=Kommunikationssysteme, CN=Tobias Krause, E=krause@in.tum.de"
        network=auto
        auto=start
        pfs=yes
```

Abbildung 5.6: Windows ipsec.conf für Studenten

5.8.2 IPSec Verbindung starten

Durch das Programm ipsec.exe im IPSec Verzeichnis kann nun der IPSec Verbindungsaufbau gestartet werden. Der Output in der DOS Konsole informiert über den Verbindungsstatus. Zum Testen der Verbindung kann ein Rechner im Internet gepingt werden. Wenn alles geklappt hat, gehen alle Pakete von nun an geschützt zum IPSec Router und werden dort weitergeleitet.

5.9 Windows XP Client Installation für Studenten

Eine Installation unter Windows XP unterscheidet sich grundsätzlich nur in den zwei folgenden Punkten: Die 3DES Unterstützung ist in XP bereits enthalten. Die Installation des High Encryption Package entfällt.

Es wird anstatt des Programms IPSecPol.exe das Programm IPSecCmd.exe von der Windows XP CD benötigt. Dafür müssen die XP Support Tools durch die setup.exe im Verzeichnis `\SUPPORT\TOOLS` installiert werden. Um das Programm IPSecCmd.exe zu erhalten, muss die Option `<Complete Installation>` gewählt werden.

Ansonst folgt die Installation dem bei Windows 2000 beschriebenen Weg.

Kapitel 6

Test & Performance

6.1 Verbindungsaufbau

Der IPSec Verbindungsaufbau wurde mit dem Befehl:
projekt2: `# /etc/init.d/ipsec start`
durchgeführt. Die nachfolgenden KLIPS und PLUTO Debugging Meldungen deuten auf den erfolgreichen Verbindungsaufbau hin und werden in der Datei /var/log/messages angezeigt:

Jan 4 16:55:54 projekt2 ipsec_setup: Starting FreeS/WAN IPsec 2.00pre0...
Jan 4 16:55:54 projekt2 ipsec_setup: KLIPS debug ‘none’
Jan 4 16:55:55 projekt2 kernel: klips_debug:ipsec_sadb_cleanup: removing all SArefFreeList entries from circulation.
Jan 4 16:55:55 projekt2 ipsec_setup: KLIPS ipsec0 on eth0 141.84.218.151/255.255.255.128 broadcast 141.84.218.255
Jan 4 16:55:55 projekt2 ipsec__plutorun: Starting Pluto subsystem...
Jan 4 16:55:55 projekt2 ipsec_setup: ...FreeS/WAN IPsec started
Jan 4 16:55:55 projekt2 Pluto[11123]: Starting Pluto (FreeS/WAN Version 2.00pre0 X.509-1.0.1)
Jan 4 16:55:56 projekt2 Pluto[11123]: Changing to directory ’/etc/ipsec.d/cacerts’
Jan 4 16:55:56 projekt2 Pluto[11123]: loaded cacert file ’cacert.pem’ (1716 bytes)
Jan 4 16:55:56 projekt2 Pluto[11123]: Changing to directory ’/etc/ipsec.d/crls’
Jan 4 16:55:56 projekt2 Pluto[11123]: loaded crl file ’crl.pem’ (703 bytes)
Jan 4 16:55:58 projekt2 Pluto[11123]: loaded host cert file ’/etc/ipsec.d/certs/clientCert.pem’ (1570 bytes)
Jan 4 16:55:58 projekt2 Pluto[11123]: added connection description ”projekt2-projekt0”
Jan 4 16:55:58 projekt2 Pluto[11123]: listening for IKE messages
Jan 4 16:55:58 projekt2 Pluto[11123]: adding interface ipsec0/eth0 141.84.218.151
Jan 4 16:55:58 projekt2 Pluto[11123]: loading secrets from ”/etc/ipsec.secrets”
Jan 4 16:55:58 projekt2 Pluto[11123]: loaded private key file ’/etc/ipsec.d/private/clientKey.pem’ (1751 bytes)
Jan 4 16:55:59 projekt2 Pluto[11123]: ”projekt2-projekt0”#1: **initiating Main Mode**
Jan 4 16:56:00 projekt2 Pluto[11123]: ”projekt2-projekt0”#1: **Peer ID is ID_DER_ASN1_DN: ’C=DE, ST=Bayern, L=Muenchen, O=LMU, OU=Kommunikationssysteme, CN=projekt0.nm.informatik.uni-muenchen.de, E=krause@in.tum.de’**
Jan 4 16:56:00 projekt2 Pluto[11123]: ”projekt2-projekt0”#1: **ISAKMP SA established**
Jan 4 16:56:00 projekt2 Pluto[11123]: ”projekt2-projekt0”#2: **initiating Quick Mode RSASIG+ENCRYPT+AUTHENTICATE+TUNNEL+PFS+DISABLEARRIVALCHECK+UP**
Jan 4 16:56:01 projekt2 Pluto[11123]: ”projekt2-projekt0”#2: **sent QI2, IPsec SA established**
Jan 4 16:56:01 projekt2 ipsec__plutorun: 104 ”projekt2-projekt0”#1: **STATE_MAIN_I1: initiate**
Jan 4 16:56:01 projekt2 ipsec__plutorun: 106 ”projekt2-projekt0”#1: **STATE_MAIN_I2: sent MI2, expecting MR2**
Jan 4 16:56:01 projekt2 ipsec__plutorun: 108 ”projekt2-projekt0”#1: **STATE_MAIN_I3: sent MI3, expecting MR3**
Jan 4 16:56:01 projekt2 ipsec__plutorun: 004 ”projekt2-projekt0”#1: **STATE_MAIN_I4: ISAKMP SA established**
Jan 4 16:56:01 projekt2 ipsec__plutorun: 112 ”projekt2-projekt0”#2: **STATE_QUICK_I1: initiate**
Jan 4 16:56:01 projekt2 ipsec__plutorun: 004 ”projekt2-projekt0”#2: **STATE_QUICK_I2: sent QI2, IPsec SA established**

die Bedeutung der Statusmeldungen können in den Abschnitten 4.6.2 und 5.2 nachgelesen werden.

6.2 Verbindungstest

Es wird von einem beliebigen Rechner im Internet aus ein Zugriff auf die internen Dienste am Lehrstuhl simuliert. Dafür baut der Rechner (projekt2 / 141.84.218.151) einen sicherern Tunnel zum IPSec Router (projekt0 / 141.84.218.150) auf und schickt einen ping request an einen Rechner im Lehrstuhlnetz (192.168.222.10). Dieser beantwortet den request durch einen ping reply. Mit tcpdump wird nun der ping Vorgang dokumentiert:

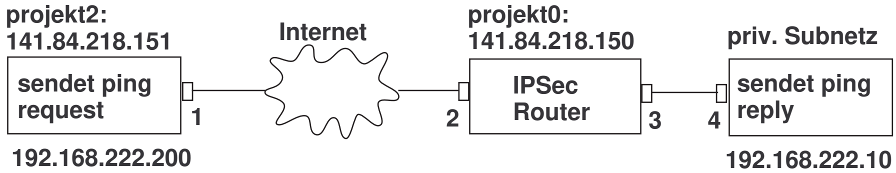

Abbildung 6.1: Verbindungstest durch ICMP

mit dem Befehl
`# ping -p deadbeef 192.168.222.10`
wird der ping request gestartet. Durch das pattern **"deadbeef"** kann der verschlüsselte oder nicht verschlüsselte Zustand des ping Pakets jederzeit erkannt werden. Bei jeder Schnittstelle wird jeweils 1 request und 1 reply Paket gezeigt.

Der ping request wird mit IPRoute2 vom virtuellen Interface eth0:1 (192.168.222.200) aus zum virtuellen ipsec Interface ipsec0 geleitet. Am ipsec0 Interface ist noch die private Absenderadresse vom virtuellen eth0:1 Interface erkennbar.

```
projekt2: #tcpdump -n -x proto 1 -i ipsec0
tcpdump: listening on ipsec0
16:57:36.694416 192.168.222.200 > 192.168.222.10: icmp: echo request
(DF)
                     4500 0054 0000 4000 4001 fc84 c0a8 dec8
                     c0a8 de0a 0800 ae96 bd2b 0300 7004 173e
                     9497 0a00 dead beef dead beef dead beef
                     dead beef dead beef dead beef dead beef
                     dead beef dead beef dead beef dead beef
                     dead
16:57:36.696425 192.168.222.10 > 192.168.222.200: icmp: echo reply
                     4500 0054 deb6 0000 fe01 9fcd c0a8 de0a
                     c0a8 dec8 0000 b696 bd2b 0300 7004 173e
                     9497 0a00 dead beef dead beef dead beef
                     dead beef dead beef dead beef dead beef
                     dead beef dead beef dead beef dead beef
                     dead
```

Nun wird das Paket mit dem ESP Protokoll verschlüsselt und zusätzlich durch das AH Protokoll authentisiert. Dies ist am externen eth0 Interface des Rechners (projekt2) sichtbar: **AH(spi=0x133e76ee,seq=0x15): ESP(spi=0x133e76ef,seq=0x15)** Das pattern "deadbeef"ist nicht mehr zu erkennen.

projekt2: #tcpdump -n -x proto 51 -i eth0

```
tcpdump: listening on eth0
16:57:49.525705 141.84.218.151 > 141.84.218.150:
AH(spi=0x133e76ee,seq=0x15): ESP(spi=0x133e76ef,seq=0x15)
                         4500 0094 9ca4 0000 4033 0dbc 8d54 da97
                         8d54 da96 3204 0000 133e 76ee 0000 0015
                         b4de dab6 25dc 8297 bcc7 7c35 133e 76ef
                         0000 0015 aae2 a956 4ef3 de49 db70 2e70
                         8093 6ab4 eb1a 233b 21f2 f113 b990 4a99
                         0723
16:57:49.527304 141.84.218.150 > 141.84.218.151:
AH(spi=0xecbf3547,seq=0x15): ESP(spi=0xecbf3548,seq=0x15)
                         4500 0094 6645 0000 4033 441b 8d54 da96
                         8d54 da97 3204 0000 ecbf 3547 0000 0015
                         65ff 4bcc 9b88 0372 763a b27e ecbf 3548
                         0000 0015 c0ed 94c6 5a7e f734 c48d d161
                         5fcf 4419 4bb7 0f95 7fb2 490e 9927 4014
                         74bd
```

Das externe Interface eth0 auf der Seite des IPSec Routers (projekt0) zeigt das gleiche Bild:

projekt0: #tcpdump -n -x proto 51 -i eth0

```
tcpdump: listening on eth0
16:52:46.177820 141.84.218.151 > 141.84.218.150:
AH(spi=0x133e76ee,seq=0x12): ESP(spi=0x133e76ef,seq=0x12)
                         4500 0094 9ca1 0000 4033 0dbf 8d54 da97
                         8d54 da96 3204 0000 133e 76ee 0000 0012
                         4bed b75b 2342 6d8d fdae b7ca 133e 76ef
                         0000 0012 ee0e 01e4 06c9 cd4c f3ae bc71
                         196d 510a 7cd2 c262 4791 a8f0 9bb7 8e38
                         aa3d
16:52:46.178440 141.84.218.150 > 141.84.218.151:
AH(spi=0xecbf3547,seq=0x12): ESP(spi=0xecbf3548,seq=0x12)
                         4500 0094 6642 0000 4033 441e 8d54 da96
                         8d54 da97 3204 0000 ecbf 3547 0000 0012
                         5dc7 10d5 95f8 bc1c f669 be4c ecbf 3548
                         0000 0012 1e68 e194 834e 06af e3d5 17bb
                         e194 35ae 372b 9d0c e32b fa96 7461 2f18
                         4ffc
```

Am ipsec0 Interface des IPSec Routers (projekt0) liegen die Pakete wieder entschlüsselt vor. AH und ESP Header wurden entfernt. Hier tritt wieder die private Adresse des Absenders zum Vorschein. Das pattern "deadbeef"ist auch wieder erkennbar:

```
projekt0: #tcpdump -n -x proto 1 -i ipsec0
tcpdump: listening on ipsec0
16:52:52.948216 192.168.222.200 > 192.168.222.10: icmp: echo request
(DF)
                    4500 0054 0000 4000 4001 fc84 c0a8 dec8
                    c0a8 de0a 0800 a735 bd2b 0c00 7c04 173e
                    85f8 0b00 dead beef dead beef dead beef
                    dead beef dead beef dead beef dead beef
                    dead beef dead beef dead beef dead beef
                    dead
16:52:52.949133 192.168.222.10 > 192.168.222.200: icmp: echo reply
                    4500 0054 debf 0000 fe01 9fc4 c0a8 de0a
                    c0a8 dec8 0000 af35 bd2b 0c00 7c04 173e
                    85f8 0b00 dead beef dead beef dead beef
                    dead beef dead beef dead beef dead beef
                    dead beef dead beef dead beef dead beef
                    dead
```

Nun werden die Pakete über das interne Interface in das Subnetz des IPSec Routers geleitet. Durch die private Absenderadresse (192.168.222.200) aus dem Lehrstuhlnetz, wird der externe Standort (141.84.218.151) des Rechners verborgen und so ein Standort im privaten Lehrstuhlnetz vorgetäuscht:

```
projekt0: #tcpdump -n -x proto 1 -i eth1
tcpdump: listening on eth1
16:52:56.247325 192.168.222.200 > 192.168.222.10: icmp: echo request
(DF)
                    4500 0054 0000 4000 3f01 fd84 c0a8 dec8
                    c0a8 de0a 0800 ca9a bd2b 1000 8004 173e
                    5993 0c00 dead beef dead beef dead beef
                    dead beef dead beef dead beef dead beef
                    dead beef dead beef dead beef dead beef
                    dead
16:52:56.247628 192.168.222.10 > 192.168.222.200: icmp: echo reply
                    4500 0054 dec3 0000 ff01 9ec0 c0a8 de0a
                    c0a8 dec8 0000 d29a bd2b 1000 8004 173e
                    5993 0c00 dead beef dead beef dead beef
                    dead beef dead beef dead beef dead beef
                    dead beef dead beef dead beef dead beef
                    dead
```

6.3 Performance

Die Ressourcen Verwendung von IPSec hängt von einer grossen Zahl von Faktoren ab, die eine einfache Aussage über die Performance nicht zulassen.

Authentifizierung mit AH HMAC-MD5-96 im Transportmodus hat selbstverständlich eine geringere Ressourcenverwendung als ESP mit Triple-DES und HMAC-SHA-1-96 im Tunnelmodus. Daneben ist die verwendete Paketgröße ebenfalls ausschlaggebend. Der Overhead fällt hier umso mehr ins Gewicht, je kleiner die Nutzdatenpakete sind.

Im Folgenden werden zwei Performance Messungen genauer beschrieben, bei denen jeweils eine einzige 20 MB Datei für den FTP Download zum Einsatz kam:

1. Verbindung eines mobilen Endgeräts zu einem Rechner im Internet mit den folgenden Anwendungen: WEP Verschlüsselung / PPTP Protokoll / mit und ohne IPSEC Router Einsatz.
2. Kommunikation von einem Roadwarrior mit einem Rechnern im Internet bzw. im internen Netz. Direkt oder über den IPSec Router mit AH und ESP Einsatz.

1.

- FTP Download

  -ohne ESP

  Bei der Anwendung der Standard WEP Verschlüsselung und des PPTP Tunneling-Protokolls konnte bei einem FTP Download von leo.org eine Geschwindigkeit von 309 KB/s erreicht werden.

  -mit ESP

  Wurde zusätzlich noch die IPSec ESP Verschlüsselung verwendet, das heißt, der Verkehr wurde über den IPSec Router geleitet, sank die durchschnittliche Downloadgeschwindigkeit auf 250 KB/s. Dies entspricht einem Durchsatz Verlust von ca. 19%.

  Andere Messungen ergaben eine Reduzierung von 15% bis 22%.

2.

- FTP Download

  -ohne AH+ESP

  Wurde vom leo.org FTP Server eine Datei direkt geladen, so konnte mit einem der Versuchsrechner eine Geschwindigkeit von 62 KB/s erreicht werden.

  -mit AH+ESP

  Der gleiche Downloadvorgang über den IPSec Router mit AH und ESP Anwendung verringerte die Geschwindigkeit auf 50 KB/s

  Verlust an Durchsatz von 19%. Folgemessungen unterschieden sich extrem stark, so dass der Messwert falsch sein kann.

  Bei einem weiteren Versuch wurde der Download von einem Rechner im privaten Netz durchgeführt. Dieser hat mittels FTP eine Datei von einem Versuchsrechner im MWN geladen, zu dem eine Tunnelverbindung bestand.

  Für die Performance Messung wurde zuerst eine Datei ohne die Anwendung von IPSec übertragen. Dabei erreichte der Downloadvorgang eine durchschnittliche Geschwindigkeit von 865 KB/s.

  Bei der Verwendung von IPSec mit ESP+AH, konnte nur noch eine Downloadgeschwindigkeit von 825 KB/s erreicht werden. Das entspricht einem Verlust bei der Datenübertragung von 5% in der Bandbreite.

FTP Download	ohne ESP	mit ESP	ohne AH und ESP	mit AH und ESP
mobiles Endgerät - Internet	309 KB/s	250 KB/s	-	-
Remote Rechner - internes Netz	-	-	62 KB/s	50 KB/s

Tabelle 6.1: Performance Messung

Trotz einer großen Anzahl von unterschiedlichen Messungen, konnte ich keine allgemeine Aussage über den tatsächlichen Ressourcen Aufwand des IPSec Systems treffen. Das mir zur Verfügung stehende Versuchsnetz bietet hierfür auch nicht die optimalen Voraussetzungen, da es von dem Einfluß des angeschlossenen MWN nicht vollständig abgekoppelt werden konnte.

Die erzielten Messungen bestätigen jedoch, dass die Nutzbandbreite beim IPSec Einsatz reduziert wird. Je mehr Nutzdatenpakete für die Übertragung eines Files benötigt werden, desto mehr Overhead wird für die Verschlüsselung und Authentisierung der einzelnen Pakete erzeugt. Dadurch wird dann die Nutzdatenbandbreite verringert.

Trotz aufwendiger Verschlüsselung und ausgelasteter VPN-Verbindung lag die Systemauslastung des Pentium 800 Systems bei unter 4%. Es ist deshalb davon auszugehen, dass der IPSec Router mit einer begrenzten Anzahl von Clients, zu denen er eine Tunnelverbindung aufbaut, keine Probleme haben wird.

Literaturverzeichnis

[BSI 02] *BSI - Bundesamt fuer Sicherheit in der Informationstechnik / Sicherheit im Funk-LAN (WLAN, IEEE 802.11)*. 2002, http://www.bsi.bund.de/ .

[Carl 02] CARLSON, NATE: *configuring an ipsec tunnel between frees/wan and windows*, 2002, http://www.natecarlson.com/linux/ipsec-x509.php .

[DFN 02] *DFN - Deutsches Forschungsnetz / Das OpenSSL Handbuch*, 2002, http://www.pca.dfn.de/certify/ssl/handbuch/ossl095/ossl095.html .

[ECK 01] ECKERT, CLAUDIA: *IT - Sicherheit*. Oldenbourg Verlag, ISBN 3-486-25298-4, 2001. 551 Seiten.

[Fack 00] FACKELMANN, DIETMAR: *IPSec Integration in einer Linux Umgebung*. 2000, http://www.nm.informatik.uni-muenchen.de/Literatur/MNMPub/Fopras/fack00/fack00.shtml .

[FHW 01] FUHRBERG, KAI, DIRK HAEGER und STEFAN WOLF: *Internet Sicherheit*. Carl Hanser Verlag, ISBN 3-446-21725-8, 2001. 494 Seiten.

[FSWAN 02] *FreeS/WAN Documentation*, 2002, http://www.freeswan.org/doc.html .

[Hube 02] HUBERT, BERT: *Linux Advanced Routing and Traffic Control HOWTO*, 2002, http://lartc.org/howto/index.html .

[IETF 02] *IETF - Working Group / IP Security Protocol (Internet-Drafts und RFCs)*, 2002, http://www.ietf.org/html.charters/ipsec-charter.html .

[ipr 02] *Fun with iproute2 and Linux FreeS/WAN*. 2002, http://www.quintillion.com/moat/ipsec+routing/iproute2.html .

[Kow 02] KOWALK, W.: *Rechnernetze Vorlesung*. 2001, http://einstein.informatik.uni-oldenburg.de/rechnernetze/authentikation.htm .

[KYAS 98] KYAS, OTHMAR: *Sicherheit im Internet*. International Thomson Publishing, ISBN 3-8266-4024-1, 1998. 460 Seiten.

[LIPP 01] LIPP, MANFRED: *VPN - Virtuelle Private Netzwerke*. Addison-Wesley, ISBN 3-8273-1749-5, 2001. 420 Seiten.

[LRZ 02] *LRZ - Leibniz-Rechenzentrum / Anschluss von mobilen Rechnern im MWN*. 2002, http://www.lrz-muenchen.de/services/netz/mobil/anschluss-mobil/ .

[SchM 99] SCHNEIDER, BRUCE und PETER MUDGE: *Cryptanalysis of Microsoft's PPTP Authentication Extensions MS-CHAPv2*. 1999.

[SIKO 01] SIKORA, AXEL: *Wireless LAN / Protokolle und Anwendungen*. Addison-Wesley, ISBN 3-8273-1917-X, 2001. 224 Seiten.

[Stef 02] STEFFEN, ANDREAS: *Den Ausweis bitte / Eigener Schluesseldienst*. 2002, http://www.heise.de/ct/02/05/220/ .