

# Teil I

## Grundlagen

Dieser Teil des Buches behandelt die allgemeinen und theoretischen Grundlagen bei dem Aufbau von virtuellen privaten Netzwerken mit Linux. Die Informationen in diesem Teil sind für das Verständnis und die Wartung von derartigen VPNs unbedingt erforderlich. Wenn Sie dieses Wissen bereits besitzen oder Sie ungeduldig mit dem Aufbau eines VPNs beginnen wollen, so können Sie direkt zum Teil 2 vor blättern. Ich möchte Ihnen jedoch auch bei entsprechender Erfahrung empfehlen später den Teil Eins nachzulesen. Vielleicht findet sich doch noch die eine oder andere Information die für Sie interessant ist.



# 1 Einleitung

Virtuelle Private Netzwerke (VPN) erlauben eine sichere, stabile und preisgünstige Kommunikation über das Internet. Mit ihrer Hilfe können verteilte Unternehmensnetze verbunden werden oder Außendienstmitarbeiter auf Ressourcen und Daten in dem Unternehmensnetz zugreifen. Sie bieten eine kostengünstige und sichere Anbindung von Filialen an eine Zentrale und erlauben den Einsatz von Telearbeitsplätzen, bei denen die Angestellten von Zuhause ihre Arbeit erledigen.

Die Implementierung eines VPNs erforderte bis vor wenigen Jahren immer die Beteiligung eines Providers und den Einsatz komplizierter und kostspieliger Hardware. Seit einiger Zeit besteht jedoch die Möglichkeit derartige VPNs komplett softwarebasiert auf der Basis des Internet Protokolls IP zu implementieren. Es existieren eine ganze Reihe von kommerziellen Lösungen, die unterschiedlich gut und häufig sehr kostspielig sind.

Das Open Source Betriebssystem Linux wird in den letzten Jahren und Monaten immer häufiger als Alternative zur Senkung der Lizenzkosten bei proprietärer Software eingesetzt. Auch seine Firewallfähigkeiten sind allgemein anerkannt. Leider wissen bisher nur wenige, dass Linux auch in der Lage ist anspruchsvolle VPN Lösungen zu bieten. Dieses Buch versucht, die Möglichkeiten und Grenzen der unter Linux existierenden Technologien aufzuzeigen und eine Anleitung für die Praxis zu geben.

Um speziell den praktischen Gesichtspunkt nicht außer Acht zu lassen, werden im Rahmen dieses Buches verschiedene klassische Szenarien für den Einsatz eines VPNs besprochen und die Linux Lösungen beschrieben.

## 1.1 Was ist ein Virtuelles Privates Netzwerk?

Ein Virtuelles Privates Netzwerk (VPN) ermöglicht die private vertrauliche Kommunikation über ein öffentliches und eigentlich unsicheres Netz. Hierzu werden virtuelle Verbindungen genutzt, die ein Abhören und Modifizieren der Informationen unmöglich machen. Als öffentliches Transfernetz wird hier meist das Internet gewählt.

Derartige VPNs erlauben eine kostengünstige Kommunikation zwischen Unternehmensnetzen weltweit. Wenn in der Vergangenheit eine dedizierte Leitung durch einen Kommunikationsanbieter bereitgestellt werden musste,

genügt nun die Anbindung über einen lokalen Internet Service Provider (ISP). So können auch lokale Netze (Local Area Network, LAN) an vollkommen unterschiedlichen Standorten, etwa Berlin und New York, ohne dedizierte Leitung, bei der der Telefonanbieter die Vertraulichkeit durch sein eigenes Netzwerk bereitstellt, sicher und vertraulich kommunizieren.

Ein VPN ermöglicht auch den sicheren Zugang zu internen Ressourcen für Außendienstmitarbeiter die von unterschiedlichen Standorten zugreifen wollen. Bisher wurden hierzu meist Modem Pools durch die Unternehmen zur Verfügung gestellt, bei denen sich der Außendienstmitarbeiter – meist über ein Ferngespräch – einwählen konnte. Die Verbindungskosten sind jedoch hoch und garantieren dennoch nicht die Vertraulichkeit, wenn die Informationen nicht verschlüsselt übertragen wurden.

Ein VPN kann hier eine Lösung bieten, da die Außendienstmitarbeiter sich nun über nationale oder internationale ISPs in das Internet einwählen können. Anschließend kann ein VPN auf der Basis dieser Internetverbindung aufgebaut werden. So kann die Vertraulichkeit und Integrität der ausgetauschten Informationen sichergestellt werden.

## 1.2 Aufgaben eines VPN

Ein Virtuelles Privates Netzwerk stellt ein Sicherheitsinstrument dar. Um die Aufgaben eines VPNs richtig verstehen zu können, sollen zunächst einige Risiken im Internet aufgezählt und erläutert werden.

### 1.2.1 Gefahren im Internet

Ein lokales Netzwerk (LAN) ist üblicherweise relativ gut geschützt. Ein Zugriff von außen ist nicht möglich. Wenn die Benutzer des Netzwerks vertrauenswürdig sind, ist mit böswilligen Angriffen nicht zu rechnen. Häufig genügen dann einfache Maßnahmen um die Sicherheit und Funktionalität des Netzwerkes zu gewährleisten. Hierbei handelt es sich um Maßnahmen zur Sicherung der Daten (Backup), zum Schutz vor Viren und vor Hardwareausfall. Es sollte jedoch nicht vergessen werden, dass allgemein davon ausgegangen wird, dass 40 bis 60 Prozent aller erfolgreichen Angriffe von Innen ausgeführt werden. Hier kann die Intrusion Detection eine Hilfe sein (siehe auch *Intrusion Detection für Linux Server*, Ralf Spenneberg, Markt+Technik Verlag 2002, ISBN 3-8272-6415-4).

Sobald dieses LAN jedoch mit dem Internet verbunden wird, kommen einige wesentliche Gefahren hinzu. Das Internet ist ein anonymes Netzwerk, das keine Schutzmechanismen bietet. Angriffe sind meist nicht verfolgbar und sehr einfach auszuführen. Sämtliche Informationen, die im LAN transportiert und gespeichert werden, sind nun diesen Angriffen ausgesetzt. Dies gilt insbesondere für die Daten, die über das Internet transportiert werden.

Im wesentlichen existieren vier verschiedene Gefahren im Internet:

- **Einbruch:** Der Einbruch in ein Netzwerk ist eine sehr große Gefahr bei der Anbindung dieses Netzwerkes an das Internet. Einbrüche sind möglich, da die eingesetzten Produkte Sicherheitslücken aufweisen oder bei der Administration und Konfiguration dieser Produkte Fehler gemacht wurden. Der Einbrecher entdeckt diese Lücken und Fehler und nutzt diese aus, um sich Zugang zum LAN zu verschaffen. Üblicherweise wird eine Firewall eingesetzt, um derartige Einbrüche zu vereiteln.

Sobald der Einbrecher aber erfolgreich eine Sicherheitslücke ausgenutzt hat (Exploit), ist er in der Lage sämtliche Funktionen des LANs zu nutzen, zu stören und möglicherweise auch umzukonfigurieren.

- **Falsche Identität:** Eine große Gefahr im Internet besteht in seiner scheinbaren Anonymität. Es existiert nicht die Möglichkeit einwandfrei die Identität eines Kommunikationspartners zu garantieren. Dieser kann eine falsche IP Adresse (IP Spoofing), eine falsche MAC Adresse (ARP Spoofing) oder sogar einen falschen DNS Namen (DNS Spoofing) vortäuschen. So besteht grundsätzlich die Möglichkeit, dass ein Angreifer vortäuscht, den DNS Namen `www.sparkasse.de` zu besitzen. Mögliche Besucher der Website werden dann auf die Website des Angreifers geleitet. Existieren keine weiteren über die TCP/IP Protokolle hinausgehende Methoden zur Feststellung der Authentizität der Website, kann dieser Angriff durch den Besucher nicht erkannt werden.
- **Lauschangriff (Sniffen):** Die TCP/IP Protokolle bieten keinen Schutz vor dem Abhören der übertragenen Informationen. Alle klassischen Protokolle der TCP/IP Familie in Version 4 übertragen die Informationen im Klartext. Daher ist es die Aufgabe der Applikationsprotokolle oder des Benutzers die Daten zu verschlüsseln, bevor sie an die TCP/IP Protokolle übergeben werden. Dies ist recht umständlich und erfordert einen zusätzlichen Aufwand durch den Benutzer. Erfolgt keine Verschlüsselung, so können die Daten entlang der gesamten Verbindungsstrecke mit gelesen werden (siehe 1.1). Eine Vertraulichkeit der Daten ist nicht gewährleistet.
- **Modifikation der Daten:** Sobald ein Mitlesen der Daten möglich ist, können diese Daten auch bei deren Transport verfälscht oder zusätzliche Daten eingeschleust werden. Die klassischen Protokolle der TCP/IP Familie

bieten keinerlei Integritätsschutz der transportierten Informationen. Ein Angreifer kann übertragene E-Mails oder Word-Dokumente bei deren Transport modifizieren oder erweitern. Diese Modifikation ist durch den Empfänger nicht zu erkennen (siehe Abbildung 1.1).

Ein VPN hat die Aufgabe in Zusammenarbeit mit anderen Maßnahmen einen Schutz vor diesen Gefahren zu bieten.

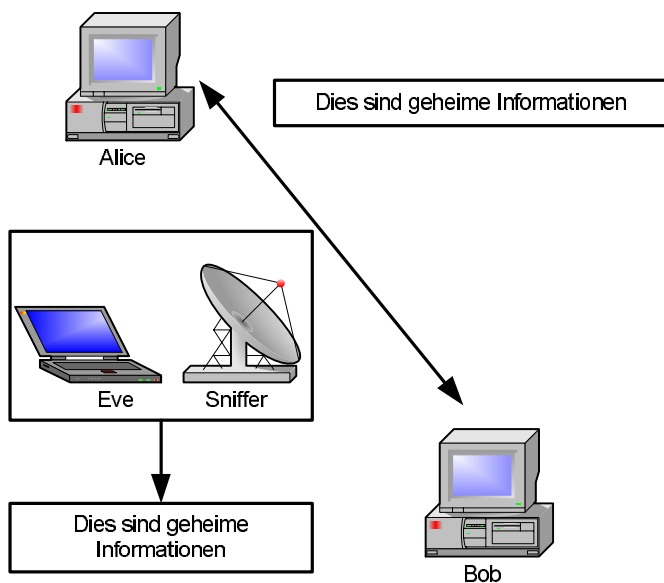


Abbildung 1.1 Tauschen Alice und Bob geheime Informationen unverschlüsselt aus, kann Eve mit einem Sniffer diesen Austausch mitlesen

## 1.2.2 Schutz durch eine Firewall

Der Schutz vor einem Einbruch wird üblicherweise durch eine Firewall gewährleistet. Die Aufgabe dieser Firewall ist es den Netzwerkverkehr über die Firewall zu untersuchen und zu kontrollieren.

Der Begriff Firewall ist vermutlich grundsätzlich bekannt. Hier soll aber dennoch kurz der Begriff und die verwendeten Technologien erläutert werden. In der Autoindustrie kennzeichnet das Wort Firewall die Wand, die den Motorraum von den Insassen trennt. Sie stellt einen Schutz vor einem möglichen Motorbrand dar, und muss in der Lage sein, diesem zu widerstehen.

Übertragen auf Computernetze stellt eine Firewall ein trennendes Gerät zwischen mindestens zwei Netzen dar. Sie unterbindet den ungehinderten Austausch von Informationen. Lediglich ausgewählte Daten dürfen transportiert

werden. Damit die Firewall diese Funktion erfüllen kann, darf keine zusätzliche Verbindung zwischen den beiden Netzen existieren, die eine Umgehung der Firewall erlauben würde. Die Firewall muss die einzige Verbindung sein.

Es gibt nun verschiedene Techniken eine Firewall zu implementieren. Die beiden am weitesten verbreiteten Techniken sind der Paketfilter und der Filter auf der Schicht des Applikationsprotokolls, häufig auch als Proxy bezeichnet. In vielen Fällen setzen Firewallssysteme beide Techniken ein. Beispiele für Open Source Produkte, die diese Techniken implementieren, sind *ipchains* und *iptables (netfilter)* als Paketfilter und *squid* und *httpf* als Proxy. Beide Ansätze unterscheiden sich stark in ihrer Performanz und in ihren Filtermöglichkeiten.

## Paketfilter

Wie es der Name schon sagt: Der reine Paketfilter ist in der Lage Pakete zu filtern. Dazu betrachtet er die Header der IP Pakete. Die meisten Paketfilter können den IP Header und, wenn vorhanden, auch den TCP, UDP und ICMP Header lesen und verarbeiten. Bei diesen Informationen handelt es sich um die IP Adressen, das IP Protokoll, zum Beispiel TCP, UDP, ICMP, IGMP, ESP, AH, wenn vorhanden die TCP und UDP Ports und den ICMP Code. Weitere Informationen im IP Header sind beispielsweise der Fragmentierungszustand, Länge des Paketes, TTL und TOS Werte. Virtuelle private Netzwerke auf der Basis von IPsec verwenden als IP Protokolle das Encapsulated Security Payload Protokoll (ESP) und das Authentication Header Protokoll (AH). Dies sind die IP Protokolle 50 und 51 respektive. Diese Protokolle verwenden jedoch keine Portnummern.

Mit Hilfe dieser Kriterien können Regeln definiert werden, die zum Beispiel nur Pakete zu einem Webserver durchlassen, wenn sie an seinen Port 80 gerichtet sind. Da ein Paketfilter normalerweise nicht in der Lage ist, den Inhalt der Pakete zu betrachten, kann er jedoch nicht feststellen, ob diese Pakete tatsächlich eine HTTP-Anfrage enthalten und ob das HTTP-Protokoll fehlerfrei verwendet wird. Der Paketfilter arbeitet meist im Kernel des Betriebssystems auf den Schichten 3 und 4 des OSI Modells. Er hat normalerweise keinerlei Zugriff auf die Applikationsdaten. Die zu filternden Pakete müssen nicht an eine Applikation im Userspace weitergegeben werden. Dadurch kann der Paketfilter sehr schnell arbeiten.

Es existieren zwei verschiedene Varianten eines Paketfilters: einfache zustandslose Paketfilter und zustandsorientierte Paketfilter, sogenannte »Stateful Packetfilter«.

Ein zustandsloser Paketfilter (zum Beispiel `ipchains`<sup>1</sup>) ist in der Lage einzelne Pakete zu filtern. Er ist jedoch nicht in der Lage einen Zusammenhang zwischen verschiedenen Paketen herzustellen. Bei einem Paket, welches den Paketfilter von außen erreicht, ist er nicht in der Lage festzustellen, ob dieses Paket Teil einer bereits aufgebauten Verbindung ist oder eine neue Verbindung öffnet. Ein zustandsloser Paketfilter muss daher alle theoretisch möglichen Antwortpakete von außen erlauben, um eine reibungslose Kommunikation zu unterstützen.

Ein zustandsorientierter Paketfilter (zum Beispiel `iptables`<sup>2</sup>) prüft bei jeder neuen Verbindung, ob sie entsprechend den Regeln erlaubt ist. Er erzeugt dann einen Eintrag in seiner Zustandstabelle. Anschließend können weitere Pakete dieser Verbindung automatisch zugelassen werden. Es müssen nicht mehr alle denkbar möglichen Antwortpakete erlaubt werden. Der Paketfilter erlaubt nur noch diejenigen Pakete, die zu vorher aufgebauten und entsprechend den Regeln authentifizierten Verbindungen gehören. Dies erhöht die Sicherheit des Paketfilters. Damit ist ein zustandsorientierter Paketfilter gewissermaßen ein Verbindungsfilter.

Viele dieser Paketfilter unterstützen die »Stateful Inspection«. Hierbei betrachtet der Paketfilter auch den Inhalt einiger Pakete für die Verwaltung seiner Regeln. Dieses Verhalten wird benötigt, da einige Protokolle vom üblichen Standard einer IP Verbindung zwischen einem Client und Server abweichen. Normalerweise kontaktiert der Client von einem hohen Port (Port  $\geq 1024$ ) den Server auf einem privilegierten Port (Port  $< 1024$ ). Über diese Verbindung werden *alle* Informationen ausgetauscht.

Der bekannteste Vertreter der Protokolle, die sich nicht an diesen Standard halten ist FTP. Der Client verbindet sich von einem hohen Port auf dem Port 21 (`ftp control port`) auf dem Server. Diese Verbindung wird verwendet um die Informationen zur Anmeldung und die weiteren Befehle zu übertragen.

1. Der Paketfilter `ipchains` ist auch in der Lage Pakete zu maskieren. Hierbei wird die Absender IP Adresse in den Paketen ausgetauscht. Damit die Antwortpakete später den korrekten Absendern und Ports zugeordnet werden können, muss `ipchains` eine Zustandstabelle pflegen und stellt in dem Moment eine Art zustandsorientierten Paketfilter dar. Dies trifft jedoch nur für die maskierten Verbindungen zu!
2. Der Paketfilter `iptables` ist nur dann ein zustandsorientierter Paketfilter, wenn das `ip_conntrack.o`-Modul geladen wurde. Dies erfolgt automatisch, wenn der Paketfilter ein Network Address Translation (NAT) durchführt. Zusätzlich müssen jedoch diese Funktionalitäten auch von den Regeln genutzt werden. Für die »Stateful Inspection« müssen ebenfalls weitere Module geladen werden.

Sobald der Server Daten auf den Client übertragen muss (Verzeichnisinhalt oder Datei), öffnet der Server eine Verbindung von Port 20 (ftp data port) auf einen anderen hohen Port des Clients. Dies bezeichnet man als aktives FTP, da der Server eine aktive Rolle einnimmt (siehe Abbildung 1.2). Der zu verwendende hohe Port wird zuvor von dem Client an den Server in einem sogenannten PORT Kommando übertragen. Stateful Inspection bedeutet, dass die Firewall in der Lage ist, das PORT Kommando zu erkennen und anschließend spezifisch die aktive FTP Verbindung zu erlauben. Eine zustandslose Firewall kann diesen Zusammenhang nicht herstellen und muss daher grundsätzlich Pakete von jedem beliebigen Rechner und Port 20 auf jeden hohen Port eines Clients zulassen, um aktives FTP zu unterstützen.

Die Stateful Inspection stellt die einzige Ausnahme dar, bei der ein Paketfilter intelligent auf den Inhalt des Paketes zugreift. Dies kann auch für die Applikationsprotokolle Internet Relay Chat (IRC), Point to Point Tunneling Protocol (PPTP), H.323, ICMP und andere erfolgen. Ansonsten betrachtet jedoch ein Paketfilter nur die Header der Pakete. Er ist mehr oder weniger ein intelligenter Router! Das bedeutet, dass die Verbindung bei einem Paketfilter (im Gegensatz zum Proxy) zwischen dem echten Client und dem echten Server aufgebaut wird.

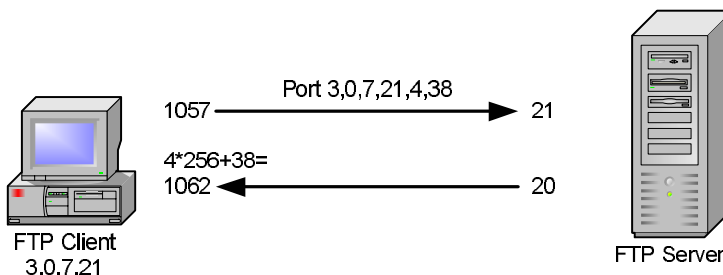


Abbildung 1.2 Aktive FTP Verbindung

## Proxy-Firewall

Ein Filter auf der Schichten des 5 bis 7 des OSI Modells (Proxy) betrachtet die Pakete nicht. Ein Proxy arbeitet im Userspace und das Betriebssystem bereitet ihm die Pakete zu einem Datenstrom auf. Diesen Datenstrom kann nun der Proxy verarbeiten. Dabei ist er theoretisch in der Lage auf sämtliche Informationen des Datenstroms zuzugreifen, diese zu untersuchen und zu verändern.

Der Proxy fungiert hierbei als ein Mann in der Mitte (Man-in-the-Middle). Der Proxy nimmt an Stelle des Servers die Anfragen des Clients als Datenstrom entgegen. Er verarbeitet und filtert diese Anfrage und leitet sie anschließend als Client an den echten Server weiter. Dieser sendet seine Antwort an den Proxy, der erneut in der Lage ist, die Daten zu analysieren und zu filtern. Schließlich wird der Proxy die Daten an den echten Client zustellen.

Ein Proxy erlaubt nicht den Aufbau von Netzwerkverbindungen zwischen dem Client und dem Server. In Wirklichkeit werden zwei Netzwerkverbindungen aufgebaut: Client-Proxy und Proxy-Server. Es existiert kein Paket-austausch zwischen dem Client und dem Server!

Das größte Problem bei der Implementierung einer Firewall rein auf der Basis von Proxies stellen die Applikationsprotokolle selbst dar. Diese weisen keine gemeinsame Grundlage auf. Sie unterscheiden sich in ihren Befehlen, ihrer Syntax, Sprache und Funktionalität sehr stark. Daher ist es erforderlich für jedes Applikationsprotokoll einen eigenen Proxy zu entwickeln, der in der Lage ist dieses Protokoll zu verstehen, zu filtern und weiterzuleiten. So stellt das HTTP Applikationsprotokoll andere Anforderungen an einen Proxy als das POP3 E-Mail Protokoll.

Kommerzielle Firewalllösungen auf der Basis eines Proxy als auch Open Source Lösungen sind daher nicht in der Lage sämtliche Protokolle nativ zu unterstützen. In solchen Fällen kommen häufig weitere generische Proxies zum Einsatz, die lediglich die Verbindung auf einem Port entgegennehmen und eine neue Verbindung aufbauen. Hierbei ist aber keine Analyse oder Filterung des Datenstroms möglich. Diese Proxies werden auch als Circuit Relay oder Plug Proxy bezeichnet.

Ein Proxy hat durch seine Sicht auf den Datenstrom wesentlich mehr Möglichkeiten als ein einfacher Paketfilter. Dies soll am Beispiel eines HTTP-Proxies für den WWW Zugriff beschrieben werden.

- Der Proxy kann in Abhängigkeit der URL filtern. Ein Paketfilter sieht lediglich die IP Adressen der Kommunikationspartner. Heute werden häufig viele verschiedene Websites auf einem Rechner gehostet. Ein Paketfilter ist nicht in der Lage, zwischen diesen Websites oder zwischen verschiedenen Bereichen eine Site zu unterscheiden.
- Der Proxy kann in Abhängigkeit des Inhaltes der Datei filtern. Ein Proxy erkennt den Beginn und das Ende der Dateien. Dadurch kann er den Dateityp erkennen und überprüfen und den Inhalt auf bestimmte Eigen-

schaften oder Viren testen. Bei einem Bild kann zum Beispiel geprüft werden, ob es sich tatsächlich um ein Bild handelt oder ob es doch eine ausführbare Datei ist.

- Ein Proxy kann den Dateiinhalt verändern. Dies ist zum Beispiel sinnvoll bei aktiven Inhalten von Webseiten. Ein Proxy kann JavaScript Inhalte filtern und so modifizieren, dass sie vom Client nicht ausgeführt werden.
- Ein Proxy kann eine Authentifizierung eines Benutzers vor dem Zugriff auf die Webseite verlangen.

Dies sind Fähigkeiten, die ein normaler Paketfilter nicht zur Verfügung stellen kann. Der Proxy benötigt jedoch auf Grund der fortgeschrittenen Möglichkeiten wesentlich mehr Ressourcen als ein Paketfilter. Speziell ein Virenskan ist sehr zeitaufwendig und ermöglicht teilweise auch Denial-of-Service Angriffe.<sup>3</sup>

Diese Fähigkeiten stehen deshalb auch nicht bei allen Proxies zur Verfügung. Besonders der generische Proxy ist nicht in der Lage derartige Filterfunktionen zur Verfügung zu stellen. In vielen Umgebungen ist die Implementierung fortgeschrittener Filterfunktionen durch einen Proxy nicht möglich, da die Anforderungen an die Bandbreite der Netzwerkverbindung nur von einem Paketfilter erfüllt werden können.

## Zusammenfassung

Eine Firewall ist also in der Lage die Kommunikation einzuschränken und nur in einer bestimmten Richtung bestimmte Inhalte zu erlauben. Dennoch kann eine Firewall nur im Rahmen der Richtlinien ihre Filterfunktionen wahrnehmen. Erlaubt eine Firewall den Zugriff auf JavaScript Inhalte einzuschränken, so besteht meist nicht die Möglichkeit zwischen gutartigem und bösartigem JavaScript zu unterscheiden. Ähnliche Einschränkungen gelten für Java und andere aktive Inhalte.

Eine Firewall ist nicht in der Lage die folgenden Punkte zu garantieren:

- **Integrität der übertragenen Daten** Eine Firewall kann nicht erkennen, ob die Daten bei ihrer Übertragung verändert oder ausgetauscht wurden.
- **Vertraulichkeit der übertragenen Daten** Eine Firewall ist nicht in der Lage einen verschlüsselten Kanal aufzubauen, um die Vertraulichkeit der übertragenen Daten sicherzustellen.

Dies sind Funktionen, die von einem VPN bereitgestellt werden.

---

3. 42.zip führt einen DoS Angriff gegen Mailserver mit Virusscanner durch. Diese etwa 42 kByte große Datei erzeugt beim Auspacken 1.048.576 Dateien mit einer Gesamtgröße von etwa 4 PentaByte (4.503.599.626.321.920 Byte).

### 1.2.3 Schutz durch ein VPN

Ein VPN bietet Authentifizierung, Vertraulichkeit und Schutz der Integrität der übertragenen Informationen. Diese Punkte sollen im weiteren genauer betrachtet werden.

#### Authentifizierung

Die Authentifizierung ist ein sehr wichtiger Bestandteil beim Aufbau eines virtuellen privaten Netzwerkes. Eine erfolgreiche Authentifizierung ist die Voraussetzung für den Aufbau einer anschließenden verschlüsselten Verbindung. Wird die Authentifizierung übersprungen so besteht die Gefahr eines sogenannten Man-in-the-Middle-Angriffes (s.u.).

Für eine Authentifizierung können drei unterschiedliche Faktoren einzeln oder in Kombination genutzt werden:

- **Wissen:** zum Beispiel ein Kennwort. Die Authentifizierung kann erfolgen, da der Benutzer etwas weiß.
- **Besitz:** zum Beispiel eine Smartcard. Die Authentifizierung kann erfolgen, da der Benutzer eine Smartcard besitzt. Dieser Besitz zeichnet ihn als korrekten Benutzer aus.
- **Person:** zum Beispiel ein Fingerabdruck. Die Authentifizierung erfolgt biometrisch und testet die Person direkt. Die Identität der Person wird so eindeutig erkannt.

Häufig werden diese Verfahren in Kombination eingesetzt. So sind Smartcards meist zusätzlich mit einem Kennwort geschützt. Die biometrischen Verfahren haben leider noch nicht eine Reife erlangt, die ihren Einsatz im Consumerbereich rechtfertigen würde.<sup>4</sup>

Im Folgenden soll nun die Wichtigkeit der Authentifizierung eines Kommunikationspartners an zwei Beispielen verdeutlicht werden.

Stellen Sie sich vor Sie möchten ein gebrauchtes Auto privat für 5000 Euro erwerben. Dann werden Sie sicherlich nicht mit dem Verkäufer lediglich die Schlüssel gegen den Geldbetrag tauschen. Sie werden zusätzlich eine Authentifizierung verlangen, dass das Fahrzeug auch tatsächlich dem Verkäufer ge-

---

4. Siehe c't 11/2002, S. 114: Biometrie.

hört. Diese Authentifizierung erfolgt zum Beispiel, indem der Verkäufer Ihnen sowohl den Fahrzeugbrief als auch seinen Personalausweis vorzeigt. Erst dann können Sie sicher sein, dass er das Recht hat das Fahrzeug zu verkaufen, denn er besitzt den Brief. Und Sie wissen, dass er tatsächlich derjenige ist, der er vorgibt zu sein.

Stellen Sie sich nun vor, dass der Verkäufer Ihnen einen italienischen Fahrzeugbrief zeigt und selbst über einen spanischen Personalausweis verfügt. Sie werden sicherlich nicht leicht bereit sein, ihm das Fahrzeug abzukaufen, da Sie zunächst nicht in der Lage sind, die Validität seiner Dokumente zu überprüfen. Im Falle der deutschen Dokumente stellt das jedoch kein Problem dar, da das Layout eines Personalausweises und eines Fahrzeugbriefes grundsätzlich bekannt sind.

Ein ähnliches Problem tritt auf, wenn Sie fünf Tage vor Weihnachten feststellen, dass Ihnen noch ein Geschenk fehlt. Sie haben leider keine Zeit mehr, um lange durch Geschäfte zu streifen und nach einem Geschenk zu suchen. Sie erinnern sich, dass Online Shops wie zum Beispiel Amazon.de eine Versendung bis Weihnachten noch garantieren und suchen dort entsprechende Geschenke aus. Nachdem Sie sämtliche Geschenke in Ihrem virtuellen Einkaufskorb gesammelt haben, gehen Sie zur virtuellen Kasse. Hier stellt Amazon.de fest, dass Sie bisher noch nicht Kunde sind und bittet Sie um die Eingabe Ihrer Konto- oder Kreditkarteninformationen.

Nun stehen Sie vor einem Problem. Zum einen möchten Sie ihre Informationen verschlüsselt übertragen. Hierzu müssen Sie einen verschlüsselten Tunnel aufbauen. Dies ist seit Diffie Hellman den nach ihnen benannten Schlüsselaustausch erfunden haben, (siehe Abschnitt 2.5.6, »Diffie Hellman«) sehr einfach mit der Secure Socket Layer (SSL) des Hypertext Transport Protokolls (HTTP) möglich. Ihnen fehlt jedoch zuvor eine Authentifizierung von Amazon.de. Nur weil die Website überzeugend aussieht, bedeutet das ja noch lange nicht, dass Sie tatsächlich auf der Website von Amazon gelandet sind. Es könnte ja sein, dass ein Angreifer unsere Anfrage an Amazon.de abgefangen hat und auf seinen Rechner umgeleitet hat (siehe Exkurs DNS-Spoofing). Bei dem Autokauf konnten Sie sich den Personalausweis des Verkäufers zeigen lassen. Ganz so einfach ist das in diesem Fall nicht möglich. Ein Man-in-the-Middle Angriff ist möglich (siehe Abbildung 1.3).

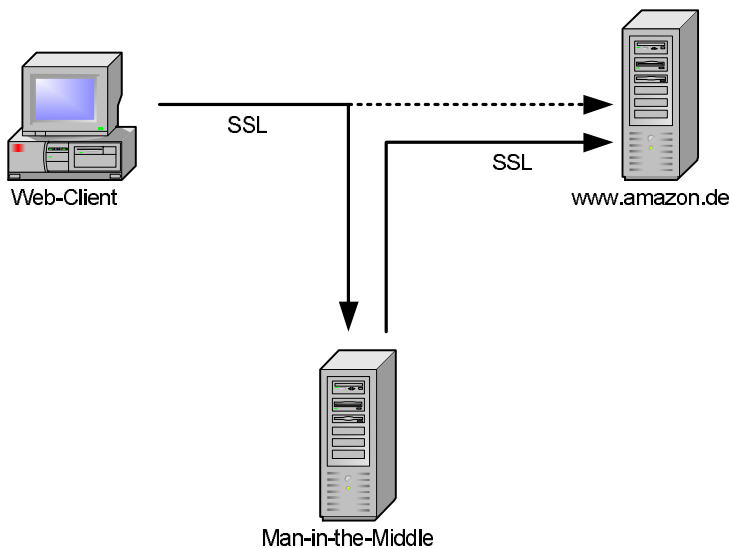


Abbildung 1.3 Ein Man-in-the-Middle Angriff

### Exkurs: DNS-Spoofing

Computer im Internet kommunizieren miteinander über ihre IP Adresse. Dies ist eine Nummer, die aus vier Bytes besteht. Zur einfachen Darstellung werden die Bytes üblicherweise durch Punkte voneinander getrennt, zum Beispiel 10.5.171.253. Da es sehr schwer ist sich diese IP Adressen zu merken, erhalten Computer zusätzlich einen Namen. Für die Auflösung des Namens in die entsprechende IP Adresse und umgekehrt haben sich im Laufe der Zeit verschiedene Systeme etabliert, von denen das Domain Name System (DNS) das heute am meisten verwendete System ist. Dieses System ist verantwortlich dafür, dass ein Rechner einen DNS Namen in die entsprechende IP Adresse auflösen und für eine IP Adresse auch den entsprechenden Namen ermitteln kann.

Wenn ein Benutzer in seinem Browser die Adresse `http://www.amazon.de` eingibt, so wird dieser Browser zunächst eine DNS Anfrage stellen um die IP Adresse des entsprechenden Rechners in Erfahrung zu bringen. Erhält er hierbei eine falsche IP Adresse, so spricht man von DNS-Spoofing. So besteht zum Beispiel die Möglichkeit, dass ein Angreifer einen Benutzer auf eine andere Website umlenkt und ihm falsche Informationen unterschiebt.

Es existieren grundsätzlich zwei Methoden, mit denen das DNS-Spoofing erfolgen kann:

1. DNS Server kennen nicht alle DNS Namen des Internets. Daher müssen Sie häufig bei anderen DNS Servern nachfragen um die Auflösung eines DNS Namens in eine IP Adresse zu gewährleisten. Um nicht für denselben DNS Namen nach kurzer Zeit eine neue Anfrage zu starten, cachen die DNS Server diese, von anderen DNS Servern gelieferten, Ergebnisse. Die Dauer der Zwischenspeicherung bestimmt der liefernde DNS Server. Grundsätzlich erlaubt es das DNS Protokoll dem antwortenden DNS Server zusätzliche Informationen, die nicht ursprünglich angefragt wurden, mitzuliefern. Diese werden von dem fragenden DNS Server dann häufig auch gecached. Das wird als DNS Cache Poisoning (siehe Abbildung 1.4) bezeichnet. Moderne DNS Server bieten üblicherweise Funktionen um dies zu unterbinden.

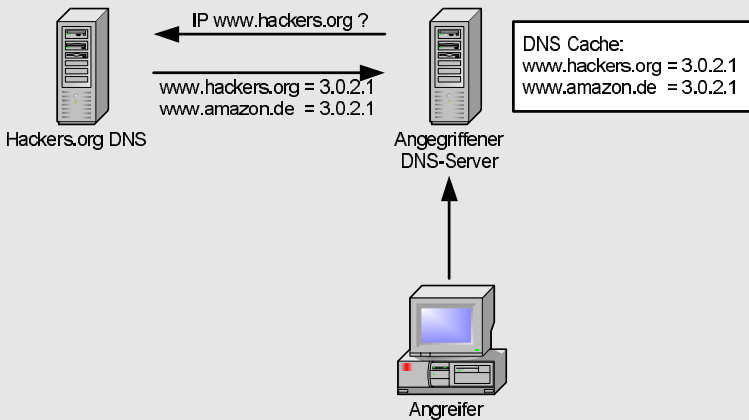


Abbildung 1.4 DNS Cache Poisoning

Solch ein Angriff wurde in dem New Yorker Wahlkampf 1999 auf die Website von Hillary Clinton angewendet, um Zugriffe auf Ihre Website <http://www.hillary2000.org> umzulenken auf die Website <http://www.hillaryno.org>. Genauso kann der Angriff genutzt werden, um Kunden von Amazon.de auf eine andere Website umzuleiten.

2. Bei der zweiten Variante werden direkt die Anfragen des Browsers an den DNS Server oder des DNS Servers an weitere DNS Server aufgefangen und durch ein Programm des Angreifers direkt beantwortet. Da dieses Programm wahrscheinlich wesentlich schneller die Anfrage beantworten kann als ein DNS Server, der zunächst in seiner Datenbank suchen muss, wird diese Antwort als korrekte Antwort akzeptiert. So kann ein Angreifer also warten, bis er eine entsprechende Anfrage im Netz erkennt und dann sein Opfer gezielt auf die falsche IP Adresse lenken.

Um dies im Zusammenhang mit einer SSL verschlüsselten Verbindung ausnutzen zu können wird noch eine Anwendung benötigt, die auf dem Rechner des Angreifers läuft, den verschlüsselten Tunnel aufbaut und dem Opfer den Eindruck vermittelt dies sei der korrekte Rechner. Dug Song hat derartige Werkzeuge bereits vor mehr als zwei Jahren öffentlich vorgestellt. Hierbei handelt es sich um die Werkzeuge `webmitm` und `dnsspoof` seines Programmpaketes `dsniff`.

Public Key Kryptografie bietet hier Hilfe. Eine genauere Betrachtung dieser Methode erfolgt in den späteren Kapiteln. Bei der Public Key Kryptografie, erzeugt ein Benutzer für sich immer zwei Schlüssel. Der eine Schlüssel wird als privater Schlüssel bezeichnet und stellt die Identität des Benutzers dar. Jeder, der über diesen Schlüssel verfügen kann, kann sich als der entsprechende Benutzer ausgeben. Häufig wird dieser Schlüssel zum Schutz noch mit einem Kennwort verschlüsselt und auf einer Smartcard gespeichert. Der zweite Schlüssel wird als öffentlicher Schlüssel bezeichnet. Dieser Schlüssel kann frei abgegeben werden.

Die Besonderheit der Public Key Kryptografie liegt nun in der Beziehung der beiden Schlüssel. Eine Nachricht, die mit dem privaten Schlüssel verschlüsselt wurde, kann *nur* mit dem entsprechenden öffentlichen Schlüssel entschlüsselt werden. Dies gilt dementsprechend auch in die andere Richtung. Eine mit dem öffentlichen Schlüssel verschlüsselte Nachricht kann nur mit dem privaten Schlüssel entschlüsselt werden (siehe Abbildung 1.5).

Dieses Verfahren kann nun zur Authentifizierung von Amazon.de genutzt werden. Für den Webserver von Amazon.de wird ein derartiges Schlüssel-paar erzeugt. Der öffentliche Schlüssel wird zum Kunden übertragen. Anschließend kann der Kunde bevor er sensitive Daten an Amazon überträgt die Authentifizierung von Amazon verlangen. Hierzu kann er eine große zufällige Zahl an Amazon übermitteln und Amazon.de auffordern, diese Zahl mit ihrem privaten Schlüssel zu verschlüsseln. Amazon.de sendet diese verschlüsselte Herausforderung (Challenge) an den Kunden zurück, der sie mit dem öffentlichen Schlüssel entschlüsseln und mit der Original-Zahl vergleichen kann.

Kommen wir zurück zum Problem: Sie wollen wenige Tage vor Weihnachten noch die Geschenke einkaufen. Wie erhalten Sie den öffentlichen Schlüssel von Amazon.de? Ganz einfach. Amazon.de sendet Ihnen diesen Schlüssel über das Internet. Dies erfolgt noch nicht verschlüsselt. Da es sich um den öffentlichen Schlüssel handelt ist das auch nicht erforderlich. Woher wissen Sie nun, dass der Schlüssel tatsächlich von Amazon.de ist und nicht von einem

Man-in-the-Middle gesendet wurde? Dies stellt nun das zentrale Problem dar.

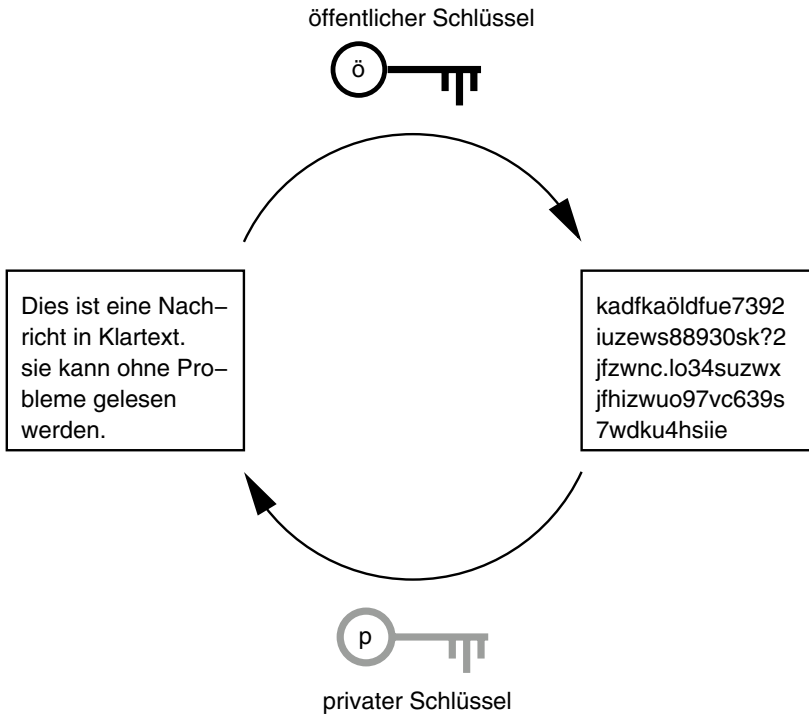


Abbildung 1.5 Ver- und Entschlüsselung mit dem Public Key Verfahren (vereinfacht)

Im Falle des Fahrzeugkaufs war es einfach, wenn der Verkäufer einen deutschen Personalausweis besaß. Dieser Personalausweis zertifizierte ihn als deutschen Staatsbürger und konnte sehr einfach überprüft werden, da das Layout und die Stempel allgemein bekannt sind. Im Grunde vertraut der Käufer der ausstellenden Stelle des Personalausweises, dass sie die Identität der Person geprüft hat. Es wird also hier eine dritte Zertifizierungsstelle (Certificate Authority, CA) genutzt.

Das Verfahren wurde auf das Internet übertragen. Hierzu hat Amazon.de den eigenen öffentlichen Schlüssel mit einer »Kopie des eigenen Personalausweises« an eine Zertifizierungsstelle gesendet. Die Zertifizierungsstelle bestätigt die Echtheit des Schlüssels, in dem sie ihn mit ihrem eigenen privaten Schlüssel signiert. Diese Signatur kann nun mit dem öffentlichen Schlüssel der Zertifizierungsstelle validiert werden. Ist die Signatur echt, dann ist auch der öffentliche Schlüssel von Amazon.de echt und der Browser des

Amazon.de Kunden kann den Challenge an Amazon.de senden. Wenn Amazon.de den Challenge richtig verschlüsselt, handelt es sich tatsächlich um den Webserver von Amazon.de.

Wie erhält nun der Amazon.de Kunde den öffentlichen Schlüssel der Zertifizierungsstelle um deren Signatur zu prüfen? Hier besteht ja dasselbe Problem wie zuvor mit dem öffentlichen Schlüssel von Amazon.de. Der Trick liegt in der Tatsache, dass die verwendeten Browser bereits sämtliche öffentlichen Schlüssel der anerkannten Zertifizierungsstellen enthalten. So können sie Zertifikate, die von diesen CAs unterzeichnet wurden, validieren.

Dieses Verfahren der Authentifizierung von Kommunikationspartnern mit Zertifikaten wird in späteren Kapiteln noch genauer erläutert. Im Grunde arbeiten die meisten guten Authentifizierungssysteme auf diese oder ähnliche Weise. Jedoch sollte der Stellenwert der Authentifizierung deutlich geworden sein. Ohne eine vorherige Authentifizierung der Kommunikationspartner kann keine Datensicherheit garantiert werden. Die Authentifizierung garantiert den Ursprung der Daten und stellt damit sicher, dass die Daten von dem gewünschten Kommunikationspartner stammen.

## **Vertraulichkeit**

Die Garantie der Vertraulichkeit der übertragenen Daten ist eine weiterer wichtiger Aspekt eines virtuellen privaten Netzwerkes. Diese Vertraulichkeit kann technisch durch einen Provider in Form eines ATM Netzwerkes gewährleistet oder durch eine sichere Verschlüsselung der Daten während des Transports garantiert werden. Die Realisierung durch einen Provider in Form eines ATM Netzwerkes ist nicht Thema dieses Buches und soll daher hier vernachlässigt werden. Wenn heute von einem Software VPN Produkt gesprochen wird, so garantiert dies die Vertraulichkeit durch eine Verschlüsselung (meist mit IPsec) der übertragenen Informationen.

Die heute eingesetzte Verschlüsselungsverfahren werden in symmetrische und asymmetrische Verfahren unterschieden. In beiden Fällen sind die mathematischen Verfahren bekannt und werden dauernd auf Herz und Nieren geprüft.

Bei den symmetrischen Verfahren wird für die Ver- und Entschlüsselung der identische Schlüssel eingesetzt. Bei den asymmetrischen Verfahren handelt es sich um Public Key Algorithmen, die mit zwei Schlüsseln arbeiten. Dabei wird die Nachricht mit einem Schlüssel verschlüsselt und kann nur mit dem entsprechenden Pendant entschlüsselt werden (vereinfacht, siehe Kapitel 2, »Kryptografie«).

Die heute im Einsatz befindlichen symmetrischen Verfahren wie DES, 3DES, AES, Blowfish, Twofish, CAST, RC4, RC5, weisen bei richtiger Anwendung keine wesentlichen Sicherheitslücken auf, die es ermöglichen würden, aus einem verschlüsselten Text auf den Klartext oder den verwendeten Schlüssel zu schließen. Ein Angriff ist lediglich durch einen sogenannten Brute Force Angriff möglich. Hierbei muss der Angreifer sämtliche möglichen Schlüssel ausprobieren. Dies dauert in Abhängigkeit des verwendeten Algorithmus, der verwendeten Schlüssellänge und der zur Verfügung stehenden Hardware unterschiedlich lange. So errechnete das Projekt *distributed.net* (<http://www.distributed.net>), dass sie bei einer dauerhaften Rechenleistung von 45,998 2GHz AMD Athlon XP Rechnern 790 Tage benötigt hätten um sämtliche möglichen RC5-64 Schlüssel auf einem verschlüsselten Text anzuwenden. Diese für das Knacken aufzuwendende Zeit lässt sich sehr einfach durch einen längeren Schlüssel exponentiell verlängern. So erfordert ein 1 Bit längerer Schlüssel den doppelten und ein 2 Bit längerer Schlüssel bereits den vierfachen Aufwand. Heutzutage übliche Längen eines symmetrischen Schlüssels sind 40, 56, 64, 128, 168 und 256 Bit. Schlüssellängen kleiner als 128 Bit werden jedoch als nicht sicher eingestuft.

Die symmetrischen Verfahren haben jedoch den Nachteil, dass der verwendete Schlüssel beiden Kommunikationspartnern bekannt sein muss. Das bedeutet, dass der symmetrische Schlüssel vor dem Aufbau der Verbindung auf geheimem Weg ausgetauscht werden muss. Erhalten dritte Personen Zugang zu diesem Schlüssel, so sind sie in der Lage die Verbindung mitzulesen.

Diesen Nachteil weisen asymmetrische Public Key Verfahren (RSA, DSA, El-Gamal, etc.) nicht auf. Hierbei erzeugt jeder Kommunikationspartner ein Schlüsselpaar aus öffentlichem und privatem Schlüssel. Anschließend werden die öffentlichen Schlüssel ausgetauscht und können zur Verschlüsselung von Nachrichten genutzt werden. Da eine Nachricht, die mit einem öffentlichen Schlüssel verschlüsselt wurde, nur mit dem privaten Schlüssel gelesen werden kann, können die so erzeugten Mitteilungen nur von der gewünschten Person gelesen werden.

Damit die asymmetrischen Verfahren jedoch als sicher gelten können sind wesentlich längere Schlüssel erforderlich. Übliche asymmetrische Schlüssellängen sind 512, 768, 1024, 2048 und 4096 Bit. Schlüssellängen kleiner als 1024 können nicht mehr als sicher eingestuft werden. Diese Schlüssellängen führen jedoch dazu, dass eine asymmetrische Verschlüsselung von rechen technischer Sicht aufwändiger ist als eine symmetrische Verschlüsselung. Daher werden üblicherweise beide Verfahren gemeinsam in einem sogenannten Hybridverfahren eingesetzt. Dabei wird die Nachricht mit einem zufälligen symmetrischen Schlüssel verschlüsselt und dieser mit einem öffentlichen

Schlüssel verschlüsselt und angehängt. Nur der Besitzer des entsprechenden privaten Schlüssels kann den symmetrischen Schlüssel und damit die ganze Nachricht entschlüsseln.

## Integrität

Schließlich ist ein VPN auch in der Lage die Integrität der übertragenen Daten zu sichern. Dies ist erforderlich, damit die übertragenen Daten nicht verfälscht oder zusätzliche Daten injiziert werden können.

Hierfür werden üblicherweise kryptografische Prüfsummen verwendet. Diese haben eine ähnliche Bedeutung wie zum Beispiel die Quersumme. Wenn zwei Personen eine Zahl austauschen und sicherstellen möchten, dass bei der Übertragung kein Fehler passiert, so ermitteln sie eine Prüfsumme zum Beispiel in Form der Quersumme und übertragen diese ebenfalls (siehe Abbildung 1.6).

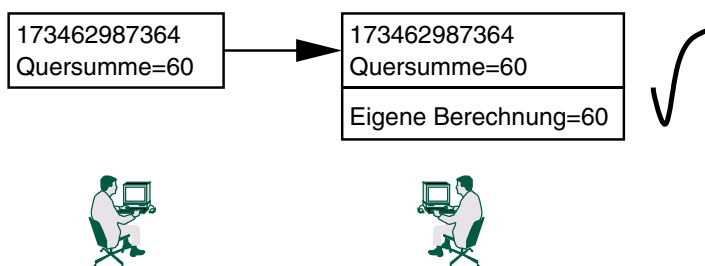


Abbildung 1.6 Schutz vor Übertragungsfehlern mit einer Quersumme

Eine einfache Prüfsumme, wie eine Quersumme, ein Paritätsbit oder die CRC32 Prüfsumme, genügt üblicherweise um zufällige Datenübertragungsfehler zu entdecken. Für diese Anwendung genügt ihre Komplexität. Wenn jedoch bewusste Veränderungen durch einen Angreifer erkannt werden sollen, so reichen diese Prüfsummen nicht mehr. Hier sind kryptografische Prüfsummen, wie MD5, SHA-1 oder RipeMD160 erforderlich. Diese Prüfsummen (Hash) verwenden derartige Algorithmen, dass es in praktikabler Zeit unmöglich ist, einen Text so zu verändern, dass er eine identische Prüfsumme ergibt.

Mit diesen Prüfsummen können nun sogenannte Authentifizierungswerte (Hash Message Authentication Codes, HMAC) erzeugt werden. Dazu erzeugt der Absender aus einem vorher ausgetauschten Geheimnis (Preshared Key, PSK) und der Nachricht eine Prüfsumme und hängt diese an. Der Emp-

fänger liest die Nachricht und erzeugt auf identische Weise die Prüfsumme. Stimmen beide Prüfsummen überein, so wurde die Nachricht nicht verfälscht und stammt aus der erwarteten Quelle. Ein Angreifer kann nicht die Nachricht so verändern, dass der Empfänger es nicht merkt, da ihm das PSK zur Erzeugung des HMAC fehlt.

## 1.3 Vor- und Nachteile eines VPN

Der Einsatz eines Virtuellen Privaten Netzwerkes weist sowohl Vor- als auch Nachteile auf. Zunächst scheint ein VPN nur Vorteile zu bieten. Seine Funktionen umfassen den Schutz der Vertraulichkeit, der Integrität und garantieren die Authentifizierung der Kommunikationspartner. Damit wird die sichere und vertrauliche Übertragung sämtlicher Daten im VPN gewährleistet. Dies gilt für alle transportierten Informationen. Bei einem VPN ist es nicht erforderlich, jedes Applikationsprotokoll einzeln abzusichern.

In der Vergangenheit wurden häufig einzelne Applikationsprotokolle gegriffen und mit zusätzlichen Methoden (zum Beispiel Secure Socket Layer, SSL) gesichert. Diese zusätzliche Ebene garantierte die Vertraulichkeit, Integrität und Authentifizierung der mit dem Applikationsprotokoll übertragenen Daten. Jedoch traf dies nur für die Daten zu, die mit dem entsprechenden Protokoll übertragen wurden. Mit SSL können HTTP, Telnet, POP, IMAP und die meisten weiteren TCP Applikationsprotokolle gesichert werden. Zusätzlich wurden aber auch komplett neue Anwendungen entwickelt, die die Verschlüsselung bereits enthielten. Die Secure Shell ist ein Beispiel für eine derartige Anwendung. Sie ersetzt die klassischen UNIX r-Dienste durch entsprechende verschlüsselnde s-Dienste.

Dennoch war für jedes Applikationsprotokoll die eigene Entwicklung einer derartigen Verschlüsselung oder eine Anpassung der Secure Socket Layer oder ihrer Weiterentwicklung, der Transport Layer Security (TLS), erforderlich. Ein VPN ist nicht auf ein Applikationsprotokoll beschränkt. Sämtliche übertragenen Daten werden unabhängig von dem verwendeten IP Protokoll verschlüsselt übertragen. Hierbei spielt es keine Rolle, ob es sich um eine TCP Verbindung oder eine UDP Verbindung handelt. Auch ICMP Pakete und selbst das Appletalk DDP Protokoll können über ein VPN übertragen werden (siehe Abbildung 1.7).

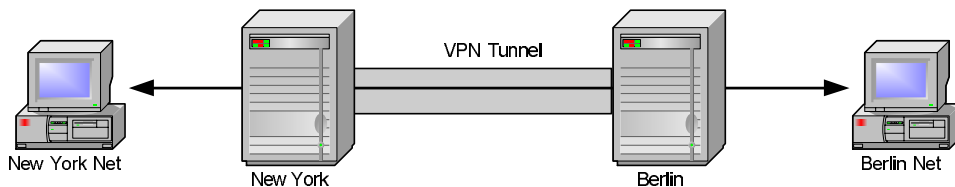


Abbildung 1.7 Ein typischer VPN Tunnel

Der Aufwand für die sichere Übertragung all dieser Protokolle hält sich in Grenzen. Es muss lediglich einmal der verschlüsselte Tunnel aufgebaut werden, anschließend können über diesen Tunnel jegliche Informationen ausgetauscht werden.

Ein derartiges VPN hat jedoch nicht nur Vorteile. Die Verschlüsselung ist nur gewährleistet zwischen den entsprechenden Maschinen, die die Verschlüsselung durchführen. Der Bereich zwischen dem Client A und dem Gateway A beziehungsweise zwischen dem Gateway B und dem Client B in der Abbildung 1.7 ist nicht verschlüsselt. Hier werden die Daten im Klartext übertragen.

Der Endbenutzer ist darüber hinaus nicht in der Lage die korrekte Verschlüsselung seiner Daten zu überprüfen. Beim Einsatz von zum Beispiel HTTPS hat der Benutzer direkt eine positive Rückmeldung der Verschlüsselung durch den Browser. Dieser kennzeichnet den erfolgreichen Aufbau einer verschlüsselten Verbindung üblicherweise mit einem geschlossenen Vorhängeschloß in der unteren Ecke. Hierbei handelt es sich also um eine Ende (Webserver) zu Ende (Webbrowser) Verschlüsselung. Bei einem VPN muss der Endbenutzer vertrauen, dass das VPN (Abbildung 1.7) seine Aufgabe korrekt erfüllt.

Möchte der Endbenutzer gar sicherstellen, dass eine E-Mail von keinem außer dem gewünschten Empfänger gelesen werden kann, so kann ein VPN dies nicht leisten. Eine derartige Verschlüsselung kann nur durch Werkzeuge wie Pretty Good Privacy (PGP) oder GNU Privacy Guard (GnuPG) erreicht werden.

### 1.3.1 VPNs und Firewalls

Die größten Probleme bei dem Einsatz eines VPN entstehen jedoch, wenn ein VPN gemeinsam mit einer Firewall eingesetzt werden soll. Dabei ist dass zunächst gar nicht zu verstehen. Beide Systeme versuchen die Sicherheit der Daten zu gewährleisten. Sie erhöhen die Sicherheit des Unternehmens. Bei

genauer Betrachtung stellt man jedoch fest, dass eine Firewall und ein VPN vollkommen unterschiedliche Methoden einsetzen um dieses Ziel zu erreichen.

VPN	Firewall
Verschlüsselung erlaubt keinen Einblick	Untersucht den IP Header und den Inhalt und protokolliert dies
Erlaubt üblicherweise über das VPN ungehinderten Zugang	Schränkt den Zugriff stark ein
Erweitert das Netz um weitere Rechner und Netze	Reduziert das zu schützende Netz auf einen Single Point of Defense

*Tabelle 1.1 Vergleich VPN – Firewall*

Die Tabelle 1.1 versucht bereits die wesentlichen Unterschiede aufzuzeigen.

Die wesentliche Tätigkeit eines VPNs liegt in der Verschlüsselung sämtlicher übertragener Informationen. Eine Firewall kann diese verschlüsselten Daten dann nicht mehr analysieren, unterscheiden oder protokollieren. Die Firewall ist sozusagen blind. Eine Firewall kann lediglich die unverschlüsselten Daten filtern.

Ein weiterer wesentlicher Bestandteil eines VPNs ist häufig der ungehinderte Zugang zum Intranet über das VPN. Der Vorstandsvorsitzende eines Unternehmens möchte von zu Hause über das VPN genauso arbeiten können, als ob er sich an seinem Arbeitsplatz in der Firma befindet. Hierzu benötigt er ungehinderten Zugang zu allen Systemen und Ressourcen, die die Firma bietet, einschließlich der Datenbanken, Mailserver oder Dokumentenrepositorien. Die Aufgabe einer Firewall ist es jedoch, derartige Zugänge zu unterbinden oder auf ein Mindestmaß zu reduzieren. Auch hier kommt es zu einem Interessenkonflikt zwischen dem Firewall- und dem VPN-Administrator.

Der letzte Punkt stellt jedoch nach meiner Ansicht das größte Problem dar. Sobald eine VPN Verbindung mit einem anderen Netzwerk oder einem Außendienstmitarbeiter aufgebaut wurde, werden die entsprechenden Rechner Teil des eigenen Netzes. Die eigene Firewall ist plötzlich auch für den Schutz dieser Rechner vor Angriffen von außen verantwortlich. Diese Rechner befinden sich nun logisch hinter der Firewall. Die Sicherheit dieser Rechner definiert plötzlich die Sicherheit des gesamten Rechnernetzes. Wenn die Firewall von Netzwerk New York in Abbildung 1.7 nicht richtig konfiguriert ist und ein Einbruch in Netzwerk New York erfolgte, so kann der Angreifer direkt auf die Rechner in Netzwerk Berlin unter Umgehung der Firewall in Netzwerk Berlin zugreifen.

Dies ist natürlich nur möglich, wenn das VPN eine Umgehung der Firewall erlaubt. Leider wird in vielen Fällen das VPN derartig konfiguriert, dass es einen Zugang parallel zu Firewall erlaubt (Abbildung 1.8).

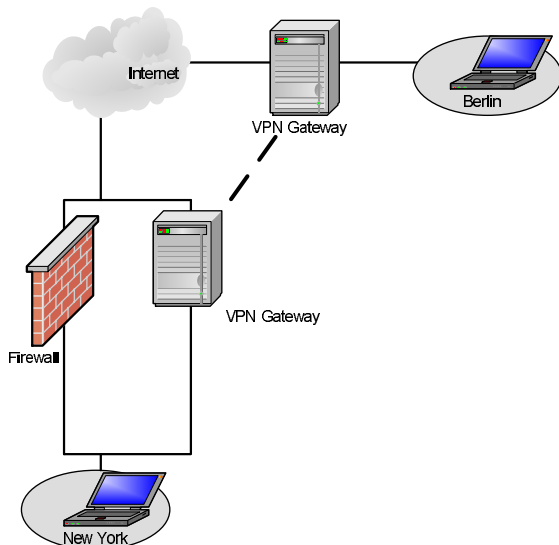


Abbildung 1.8 Firewall und VPN sind parallel zueinander aufgebaut (schlecht)

Daher sollte sich immer zwischen dem VPN Gerät und dem internen Netzwerk noch eine Firewall befinden, die den Zugriff auf das interne Netzwerk über das VPN kontrollieren und beschränken kann. Sinnvollerweise befindet sich auch vor dem VPN Gerät eine Firewall, die das VPN Gerät schützen kann (Abbildung 1.9).

So sind die über das VPN transportierten Daten und das dahinterliegende Netz optimal geschützt. Dieser logische Aufbau einer VPN/Firewall Struktur wird auch von vielen kommerziellen Anbietern geschätzt. Diese bieten häufig ein gebündeltes Produkt an, welches beide Funktionen (VPN und Firewall) bietet. Wird dieses Produkt auf einem physikalischen Gerät installiert, so kann von der logischen Funktion die in Abbildung 1.10 dargestellte Struktur realisiert werden.

Hierbei wird der normale Verkehr durch die Firewall 1 gefiltert. Parallel hierzu existiert ein VPN Gateway, welches durch zwei weitere Firewalls (2 und 3) geschützt wird. Hierbei filtert die Firewall 2 den verschlüsselten Verkehr von und nach außen und schützt die VPN Gateway Software vor Angriffen. Die Firewall 3 filtert den entschlüsselten Verkehr, der über das VPN Gateway in das interne Netz gelangt.

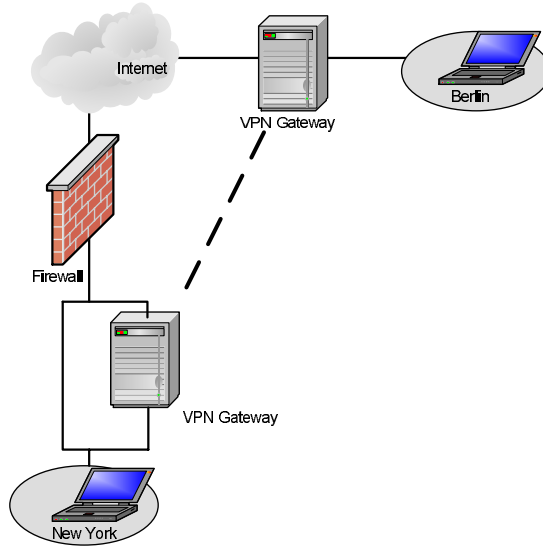


Abbildung 1.9 Firewall und VPN nacheinander geschaltet

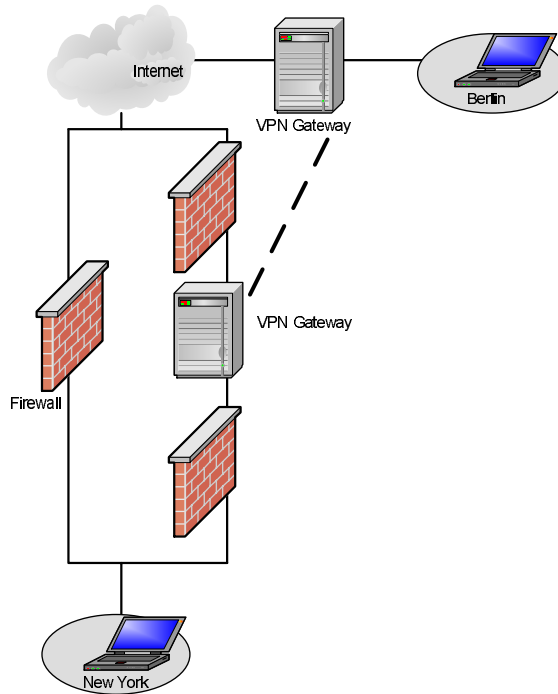


Abbildung 1.10 Ideale VPN/Firewall Struktur

Eine derartige Struktur kann mit Linux ebenfalls aufgebaut werden. Bei der Besprechung der entsprechenden Szenarien und Implementierungen werden Beispiel-Firewallregeln für Linux vorgestellt und erklärt. Dabei kann leider aus Platzgründen nicht sehr in die Tiefe gegangen werden. Wenn Sie weitere Hintergrundinformationen zum Thema Firewall und Linux benötigen, können Sie diese in dem Buch »Linux Firewalls« von Robert Ziegler, auch im Markt+Technik Verlag (ISBN 3-8272-6257-7) erschienen, nachlesen.

## 1.4 Open-Source und Sicherheit

Beim Einsatz von Sicherheitstechnologien kommt dem Stichwort Open-Source besondere Bedeutung zu – besonders im Zusammenhang mit Verschlüsselungstechnologien. Die Vergangenheit hat mehrfach gezeigt, dass in vielen Fällen Hersteller in Closed Source Produkten mangelhafte Verschlüsselung implementiert haben oder bewusst Hintertüren eingebaut haben, die die Verschlüsselung aushebeln konnten. Dabei wählten die Hersteller häufig sichere Algorithmen, jedoch wurden die Parameter falsch gewählt.

Zunächst wird wahrscheinlich jeder normale Mensch aufschrecken, wenn er hört, dass die gesamten Verschlüsselungsalgorithmen und der gesamte Quelltext für jedermann offenliegt. Wie kann eine derartige Software eine sichere Verschlüsselung durchführen?

Wie jedoch im Kapitel zur Kryptografie noch ausgeführt werden wird, liegt die Stärke der modernen kryptografischen Algorithmen genau in diesem Detail. Der Algorithmus verwendet allgemein zugängliche Methoden und Algorithmen um einen Klartext mit einem geheimen Schlüssel in den sogenannten *ciphertext* zu wandeln. Dadurch können sämtliche Kryptoanalytiker weltweit versuchen eine Sicherheitslücke im Algorithmus zu finden. Solange dies nicht der Fall ist und das Geheimnis lediglich im Schlüssel verborgen ist, gilt der entsprechende Algorithmus als sicher (entsprechende Schlüssellängen vorausgesetzt).<sup>5</sup> Implementiert jedoch eine Firma einen neuen Algorithmus und hält ihn geheim, so sollte sich der Verdacht aufdrängen, dass der Erfinder nicht das entsprechende Vertrauen in seinen eigenen Algorithmus besitzt. Dann sollte von dem entsprechenden Produkt Abstand genommen werden.

### TIPP

Der bekannte Kryptologe Bruce Schneier vergleicht den Kryptografiealgorithmus mit einem Safe. Selbst bei Kenntnis der Baupläne muss dieser Safe allen Angriffen widerstehen.

5. Schwache Schlüssel werden in dem Kapitel »Kryptografie« behandelt.

Jedoch ist es nicht nur wichtig, dass der Algorithmus allgemein begutachtet und anerkannt wurde. Dies gilt auch oder umso mehr für die Implementierung des Algorithmus in Software. Hier können Programmierfehler vorliegen, die eine Sicherheitslücke erst ermöglichen. Bei Closed Source Software besteht darüberhinaus die Möglichkeit, dass der Hersteller bewusst eine Hintertür eingebaut hat, um die Verschlüsselung oder die Authentifizierung zu umgehen. Dies kommt leider relativ häufig bei proprietärer Software vor. Der Anwender kann sich aber nicht davor schützen, da es für den Benutzer sehr schwer möglich ist die Vorgänge in der Software nachzuvollziehen.

Im Folgenden sollen nun einige Beispiele gegeben werden:

- **Borland Interbase** Die Datenbank Interbase von Borland wurde nach vielen Jahren als Closed Source Projekt am 25. Juli 2000 als Open Source freigegeben. Dieser Code wurde dann vom Firebird Projekt (<http://firebird.sourceforge.net>) weitergepflegt.

Etwa ein halbes Jahr später, am 9. Januar 2001, wird in dem Sourcecode ein hart kodiertes Login und Kennwort gefunden (<https://www.kb.cert.org/vuls/id/247371>): `politically correct`. Dies war wahrscheinlich seit 1993 bereits in der Datenbank hartkodiert vorhanden. Mit diesem Login war es möglich mit administrativen Rechten auf die Datenbank zuzugreifen. Wäre die Datenbank nicht ein Open Source Produkt geworden, wäre diese Hintertür wahrscheinlich bis heute nicht bekannt und nicht entfernt worden.

- **Crypto AG** Die Crypto AG (<http://www.crypto.ch>) ist eine Schweizer Firma, die seit über 50 Jahren kryptografische Hard- und Software herstellt und vertreibt.

Die Crypto AG vertreibt die entsprechenden Geräte weltweit und war als Schweizer Firma nicht an die Exportbeschränkungen der Vereinigten Staaten von Amerika gebunden. Der Export von starker Kryptografie aus den USA fiel dort unter das Kriegswaffenkontrollgesetz und war verboten. So verkaufte die Crypto AG auch Geräte in den Iran. Im März 1992 wurde Hans Bühler, ein Verkaufsrepräsentant der Crypto AG im Iran von iranischen Behörden mit dem Vorwurf der Spionage festgenommen und neun Monate inhaftiert. Die Crypto AG zahlte eine Million Dollar Lösegeld und holte Hans Bühler aus dem iranischen Gefängnis. In der Heimat wurde Hans Bühler entlassen und die Crypto AG forderte das gezahlte Lösegeld von ihm zurück.

Anschließend wurden Gerüchte laut, dass die iranischen Vorwürfe korrekt seien und die Geräte der Crypto AG tatsächlich vom deutschen Bundesnachrichtendienst und der amerikanischen National Security Agency (NSA) modifiziert wurden, so dass diese den Austausch von Informa-

tionen abhören konnten. Eine offizielle Bestätigung dieser Gerüchte erfolgte jedoch nie.

- **Windows NSA Key** Hierbei handelt es sich wahrscheinlich mehr um heiße Luft als um einen tatsächlichen National Security Agency Key in den verschiedenen Microsoft Windows Betriebssystemen (Windows 9x, NT und 2000). Jedoch existiert hier dennoch ein Problem mit dem Cryptosystem der Betriebssysteme. Das Cryptosystem besitzt *zwei* öffentliche Schlüssel, KEY und \_NSAKEY, die von Microsoft verwendet werden um die kryptografischen Anwendungen so zu signieren, dass die Microsoft Betriebssysteme ihnen vertrauen. Der Name des zweiten Schlüssels war Anlass zur Spekulation ob möglicherweise die NSA Zugang zu diesem Schlüssel hätte und ebenfalls derartige Anwendungen zertifizieren könnte. Ob dies der Fall ist mag bezweifelt werden. Jedoch können diese Schlüssel von Microsoft nicht zurückgerufen werden. Beide können von Microsoft eingesetzt werden. Dies ist unüblich. Normalerweise wird für derartige Operationen nur ein Schlüsselpaar eingesetzt. Dies ermöglicht einen theoretischen Angriff auf das Cryptosystem, der vom Anwender nicht erkannt werden kann (<http://www.counterpane.com/crypto-gram-9904.html#certificates>).
- **Clipper Chip** Handelte es sich bei den bisher beschriebenen Verschlüsselungslücken um heimlich implementierte Hintertüren, die öffentlich bekannt wurden, so handelt es sich beim Clipper Chip um ein spezielles Gerät zur Verschlüsselung von privater Kommunikation. Der Clipper Chip wurde am 16. April 1993 vom Weißen Haus angekündigt. Der Clipper Chip stellt sicher, dass staatliche Behörden freien Zugriff auf die verschlüsselten Informationen erhalten und so die Kommunikation überwachen können. Als kryptografischer Algorithmus kam im Clipper Chip der Skipjack Algorithmus zum Einsatz. Für den Zugriff auf die verschlüsselten Informationen erhalten zwei US Bundesbehörden (NIST und Department of Treasury) jeweils einen Schlüssel, die zusammen die Entschlüsselung der Informationen erlauben. Dies bezeichnet man als ein Key Escrow System. So soll eine Verschlüsselung möglich sein, die es dennoch den Strafverfolgungsbehörden erlaubt auf die verschlüsselten Informationen zuzugreifen. Der Nachfolger des Clipper Chip ist der Capstone Chip.
- **PGP und Key Escrow** Die Programmierer der Software PGP haben ebenfalls angefangen seit der Version 5.5 ein Key Escrow System mit einzubinden. Diese Funktionalität steht in der Business Version der Software PGP zur Verfügung. Die Software PGP wird seit 2002 von einer neuen Firma

gepflegt. Die verfügbare PGP Enterprise Version 8.0 enthält ebenfalls derartige Funktionen zur Schlüsselwiederherstellung. Diese Funktion wird von PGP als Key Reconstruction bezeichnet. Es stellt jedoch nichts anderes als ein Key Escrow System dar.

- **Lotus und Key Escrow** Lotus Notes enthielt ebenfalls lange Jahre ein Key Escrow System. Bereits vor der Lockerung der US amerikanischen Exportbeschränkungen für starke Kryptografie wollte IBM 1996 die Lotus Notes Software mit 64 Bit Schlüsseln ausstatten. Die Exportregelungen erlaubten jedoch nur Schlüssellängen von 40 Bit. Um dennoch eine Exportgenehmigung zu erhalten, wurde zusammen mit der NSA das Workgroup Differential Verfahren entwickelt. Hierbei handelt es sich um die sogenannte Differential Workgroup Cryptography. Diese verschlüsselt die Nachricht mit 64 Bit. Anschließend werden 24 Bit des Schlüssels mit einem öffentlichen Schlüssel der NSA verschlüsselt und an die Nachricht angehängt. Die NSA kann so auf 24 Bit des originalen 64 Bit langen Schlüssels in Klartext zugreifen. Die restlichen 40 Bit des Schlüssels können mit modernen Rechnern in Bruchteilen von Sekunden errechnet werden. Dennoch war die Nachricht vor jedem weiteren Angreifer mit 64 Bit geschützt. Nur die NSA konnte  $2^{24}=16.777.216$  mal einfacher den Schlüssel knacken.

Open-Source Software wird offen entwickelt. Heimlich können hier keine derartigen Hintertüren eingebracht werden, ohne dass diese relativ schnell gefunden werden – vorausgesetzt das Produkt ist so interessant, dass es mehrere Programmierer pflegen!

Es soll allerdings nicht außer Acht gelassen werden, dass in einigen Fällen derartige Hintertüren bewusst gewünscht werden. Hierbei handelt es sich um:

- Gesetze, die den Strafverfolgungsbehörden die Möglichkeit verschaffen wollen auf verschlüsselte Daten zuzugreifen.
- Unternehmen, die weiterhin Zugriff auf verschlüsselte Daten eines ausgeschiedenen Mitarbeiters benötigen. Hierbei genügt es jedoch ein Key Escrow System lediglich für die dauerhaft gespeicherten Daten einzusetzen. Es ist nicht notwendig für verschlüsselte Transaktionen.

In solchen Fällen sind Key Escrow oder Key Recovery Systeme nötig. Jedoch muss in diesen Fällen die Sicherheit des entsprechenden Systems gewährleistet werden. Darüberhinaus betreffen diese Anwendungen meist nur gespeicherte Daten. Ein Key Escrow für Kommunikationen ist meist nicht erforderlich und sollte daher auch nicht eingerichtet werden.

## 1.5 Kommerzielle Lösungen

Es existieren eine ganze Reihe von kommerziellen Lösungen die den Aufbau eines VPNs ermöglichen. Im Folgenden sollen einige Anbieter vorgestellt werden. Die Liste erhebt keinerlei Anspruch auf Vollständigkeit und ist es sicherlich auch nicht. Die Auswahl erfolgt auf Grund eigener Erfahrungen mit den Systemen. Insbesondere soll bei den einzelnen Systemen die Kompatibilität in einer gemeinsamen heterogenen VPN Lösung mit Linux betrachtet werden.

### 1.5.1 Cisco

Die Produkte von Cisco umfassen unter anderem Router, Switches, Firewalls und Intrusion Detection Systeme. Hierbei sind die meisten Router und Firewalls in der Lage auch ein VPN mit Hilfe von IPsec aufzubauen.

Die Preise der Cisco Produkte richten sich nach der Größe, dem Ausbau und der verwendeten Software. Besondere Funktionalitäten wie zum Beispiel ein VPN werden bei den Routern als zusätzliche Feature Packs verkauft.

So kostet zum Beispiel ein kleiner aktueller Cisco Router der Reihe 2600 zwischen 2000 und 3000 Euro. Hinzu kommt immer noch das entsprechende IP-Feature Pack zur Verschlüsselung, das mit weiteren etwa 800 Euro zu Buche schlägt. Der Cisco VPN Concentrator 3000 kostet rund 3000 Euro.

Die Interoperabilität der Cisco Produkte mit der IPsec Implementierungen unter Linux ist sehr hoch. Erst im November 2001 wurden diese wieder auf der jährlichen IPsec Konferenz getestet (<http://www.hsc.fr/ressources/ipsec/ipsec2001/>). Die Cisco Produkte sind in der Lage sowohl mit Zertifikaten als auch mit Kennworten (Preshared Keys, PSK) einen verschlüsselten Kanal aufzubauen.

### 1.5.2 Checkpoint FW-1/VPN-1

Checkpoint stellt mit seiner Software Firewall-1 sicherlich eine der bekanntesten Software Firewalls her. Dieses Produkt benötigt immer zusätzlich einen Rechner mit einem entsprechenden Betriebssystem (Linux/Intel, WinNT/Intel, Solaris/Intel, Solaris/SPARC, HPUX, AIX). Die einzige Ausnahme stellen die Geräte von Nokia dar. Nokia produziert Hardwaregeräte mit einem abgespeckten BSD als Betriebssystem, auf dem dann die Firewall-1 vorinstalliert ist.

Das aktuelle Produkt NG (Next Generation) bietet sowohl Firewall als auch VPN Funktionalitäten und ist sehr modular erweiterbar. Die Lizenzierung erfolgt bei Checkpoint in Abhängigkeit der zu schützenden Rechner (siehe Tabelle 1.2). Die jeweiligen VPN-Clients, die benötigt werden um die Verbindung zum Checkpoint VPN Gateway aufzubauen, sind lizenzkostenfrei.

Max. Clients	ca. EUR
25	4500
50	7000
100	11000
250	14000

*Tabelle 1.2 Checkpoint VPN-1 NG Lizenzpreise*

Die Interoperabilität ist im Falle der Checkpoint IPsec Implementierung nicht in allen Fällen gegeben. Es besteht die Möglichkeit Linux als Client mit Checkpoint kommunizieren zu lassen. Jedoch kann der freie Checkpoint VPN-Client nicht genutzt werden, um eine Verbindung zu einem Linux Gateway aufzubauen. Bei der Authentifizierung unterstützt Checkpoint sowohl x509-Zertifikate als auch Preshared Secret Keys (PSK). Die Verwendung von PSKs setzt sinnvollerweise den sogenannten Aggressive Mode voraus. Dieser wird von FreeS/WAN mit einem Patch unterstützt.

### 1.5.3 Microsoft Windows 2000 und XP

Microsoft hat mit Windows 2000 und Windows XP begonnen IPsec in seinen Betriebssystemen zu unterstützen. Dies erfolgt im Zuge der Unterstützung für IPv6, der nächsten Generation des Internet Protokolls. Die von Microsoft implementierte VPN Unterstützung setzt jedoch das IPsec Protokoll gemeinsam mit dem L2TP Protokoll ein. Dieses zusätzliche Protokoll erlaubt die Vergabe von dynamischen IP Adressen und die Authentifizierung von Benutzern. Für Linux existieren ebenfalls L2TP Dienste, jedoch ist deren Konfiguration recht aufwendig (siehe Abschnitt 9.4.2, »L2TP«).

Aber sowohl Windows 2000 als auch XP erlauben es, reine IPsec Tunnel aufzubauen, die mit Linux interoperieren können. Dies ist jedoch mit Einschränkungen verbunden (siehe Kapitel 7, »Aufbau heterogener Virtueller Privater Netze«).

### 1.5.4 Microsoft Windows 98/ME/NT

Die Microsoft Betriebssysteme Windows 98, Windows ME und Windows NT enthalten keine Unterstützung für das IPsec Protokoll. Hier sind normalerweise Werkzeuge von Drittherstellern (siehe SSH Sentinel und SafeNet SoftPK) erforderlich. Jedoch hat Microsoft im Juni 2002 einen kostenlosen IPsec/L2TP Client veröffentlicht (<http://www.microsoft.com/windows2000/server/evaluation/news/bulletins/l2tpclient.asp>). Dieser Client unterstützt jedoch leider nicht reine IPsec Verbindungen, sondern nur kombinierte IPsec/L2TP Tunnel (siehe Abschnitt 9.4.2, »L2TP«).

Dieser Client soll in Zukunft das unsicherere Point-To-Point-Tunneling Protokoll (PPTP) ablösen. Dies stellte bislang die einzige von Microsoft unterstützte Möglichkeit zum Aufbau eines VPN unter den genannten Betriebssystemen dar.

### 1.5.5 SSH Sentinel

Die Firma SSH (<http://www.ssh.com>) ist bekannt geworden durch die Vermarktung der kommerziellen Version der Secure Shell. SSH produziert jedoch inzwischen auch PKI und VPN Produkte. Der SSH Sentinel (ehemals Internet Pilot, <http://www.ssh.com/products/security/sentinel/>) ist ein IPsec VPN Client für Microsoft Windows Betriebssysteme.

Er erweitert diese Betriebssysteme um einen IPsec Stack und erlaubt die Authentifizierung mit x509 Zertifikaten und Preshared Secret Keys. Hierbei kann er in eine vorhanden Public Key Infrastructure (PKI) eingebunden werden. Die privaten Schlüssel können auf Chipkarten gespeichert werden. Zusätzlich verfügt der SSH Sentinel über eine eingebaute Personal Firewall. Der SSH Sentinel ist interoperable mit den entsprechenden Lösungen unter Linux.

Der SSH Sentinel kostet in einer Einzellizenz etwa 160 Euro und kann direkt über den SSH Online Store (<http://www.ssh.com/company/sales/store/>) bezogen werden.

### 1.5.6 SafeNet SoftRemote

Die SafeNet SoftRemote Software (früher SoftPK) wird von SafeNet, Inc. (ehemals IRE, <http://www.safenet-inc.com>) hergestellt. Dieser Client wird auch von einigen anderen Herstellern eingesetzt und stellt auch die Basis für den

oben erwähnten freien Windows L2TP-Client dar. Auch SoftRemote enthält eine persönliche Firewall. Laut Webpage ist SoftRemote der am häufigsten eingesetzte VPN Client weltweit.

Der Vertrieb der Software erfolgt über verschiedene Partner oder Online über die URL <http://www.safenet.biz/>. Die Safenet SoftRemote Software ist verfügbar für die Microsoft Betriebssysteme Windows 95, 98, 2000, NT 4.0, ME und XP. Zusätzlich existieren SoftRemotePDA Versionen für Pocket PC 2002 und PalmOS  $\geq 3.5$ . Diese Software ist verfügbar für 149 Dollar für die Desktopversionen und 39 Dollar für die PDA Versionen.

SoftRemote weist keine Probleme bei einem kombinierten Einsatz mit den Linuxversionen auf.

### 1.5.7 OpenBSD, FreeBSD, NetBSD

Bei den Betriebssystemen OpenBSD (<http://www.openbsd.org>), FreeBSD (<http://www.freebsd.org>) und NetBSD (<http://www.netbsd.org>) handelt es sich um freie UNIX Systeme. Alle diese Betriebssysteme sind in der Lage eine IPsec Verbindung aufzubauen und weisen keine Probleme in Kombination mit Linux auf. Die von diesen Systemen verwendeten IKE-Daemonen sind inzwischen auf Linux lauffähig.

### 1.5.8 Weitere Produkte

Weitere VPN Produkte werden von weiteren zahlreichen Herstellern angeboten. Hier soll kurz auf die folgenden hingewiesen werden:

- **MacOSX** Das Betriebssystem MacOSX enthält einen IPsec Stack. Dieser muss jedoch mit einem Kommandozeilenwerkzeug konfiguriert werden (ähnlich Linux). Ein grafischer Client mit dem Namen *VPN Tracker* ist verfügbar bei Equinux (<http://www.equinux.com/us/products/vpntracker/index.html>).
- **PDAs** Zwei weitere Firmen, die Clients für PDAs herstellen sind Certicom (<http://www.certicom.com/products/movian/movianvpn.html>) und Funk Software (<http://www.funk.com/>). Der Movian VPN Client scheint momentan nicht interoperabel zu sein mit FreeS/WAN. Über den Funk Client existieren keine Informationen.

## 1.6 Verschiedene VPN Szenarien

Dieses Kapitel soll bereits einige Szenarien für den Einsatz eines Virtuellen Privaten Netzwerks vorstellen. Diese sollen sowohl als Anregung dienen als auch als Einleitung zu den späteren Kapiteln, die dann Lösungen für diese Szenarien bieten.

Die im Folgenden beschriebenen Szenarien sind sicherlich nicht vollständig und beispielhaft für jede mögliche Situation. Sie sollen jedoch die klassischen Fälle für den Einsatz eines VPN beschreiben. Im weiteren Verlauf des Buches werden diese dann wieder aufgegriffen und Lösungen präsentiert.

### 1.6.1 Kommunikation zwischen zwei Netzwerken

Die Kommunikation zwischen zwei Netzen ist die häufigste Anwendung für ein Virtuelles Privates Netzwerk. Hierbei werden zwei Standorte, die jeweils über eine Internetanbindung verfügen, mit einem VPN vernetzt, so dass sie vertrauliche Informationen austauschen können. Dieses Szenario wird auch als Site-to-Site VPN bezeichnet. Abbildung 1.11 stellt exemplarisch ein derartiges VPN dar. Sämtliche zwischen den beiden Gateways ausgetauschten Informationen werden für den Transport im Internet verschlüsselt und über das Internet transportiert. Die Kommunikation innerhalb der Netze erfolgt im Klartext.

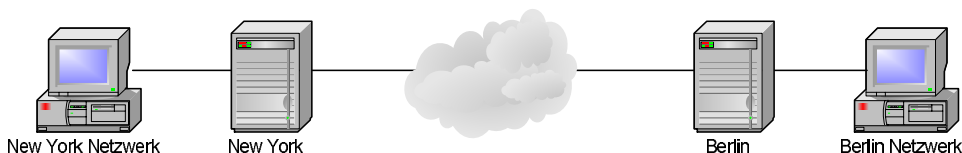


Abbildung 1.11 Site-to-Site VPN

Diese Lösung wird häufig gewählt, um zwei Filialen miteinander zu verbinden.

### 1.6.2 Kommunikation zwischen zwei Rechnern

Häufig soll nicht die Kommunikation zwischen zwei Netzwerken verschlüsselt und gesichert werden, sondern die Kommunikation zwischen zwei Rechnern. Dies wird auch als ein End-to-End VPN bezeichnet. Abbildung 1.12 skizziert ein derartiges VPN. Hierbei enthalten die kommunizierenden

Rechner auch bereits die VPN Funktionalität. Die Informationen verlassen die Rechner bereits verschlüsselt und müssen nicht mehr durch ein VPN Gateway verschlüsselt werden (Abbildung 1.12).

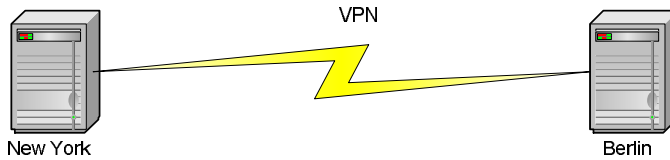


Abbildung 1.12 VPN zwischen zwei Rechnern

### 1.6.3 Kommunikation zwischen vielen festen Standorten

Dieses Szenario gleicht dem ersten Szenario. Lediglich die Anzahl der zu verbindenden Netzwerke ist größer zwei. Um eine größere Anzahl von Standorten mit einem VPN zu vernetzen existieren grundsätzlich zwei verschiedene mögliche Strukturen für den Aufbau eines VPNs: Stern und Netz.

Bei einer Sternstruktur bauen alle Standorte eine Verbindung zu einem zentralen VPN Gateway auf. Dessen Aufgabe ist es, die über das VPN transportierten Nachrichten, entsprechend zu den Empfängern zu routen (siehe Abbildung 1.13). Diese Struktur erlaubt einen sehr einfachen Aufbau. Es wird nur jeweils ein Tunnel pro Standort benötigt. Jedoch hat diese Struktur auch den Nachteil, dass bei Ausfall des zentralen Gateways die komplette VPN Kommunikation ausfällt. Aus diesem Grunde werden in derartigen Szenarien häufig die zentralen Gateways hochverfügbar ausgelegt.

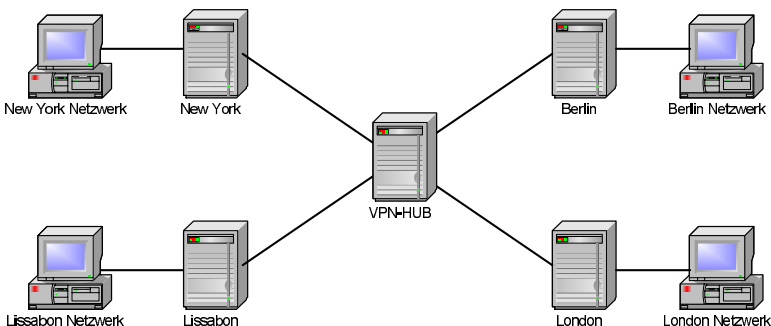


Abbildung 1.13 Sternförmiger Aufbau eines VPNs

Die netzförmige Struktur stellt wesentlich höhere Ansprüche an die Administration und Wartung des VPNs. Hierbei baut jeder Standort einen Tunnel zu jedem weiteren Standort auf. Dies gewährleistet die direkte Kommunikation und umgeht mögliche Verfügbarkeitsprobleme eines zentralen Gateways (siehe Abbildung 1.14).

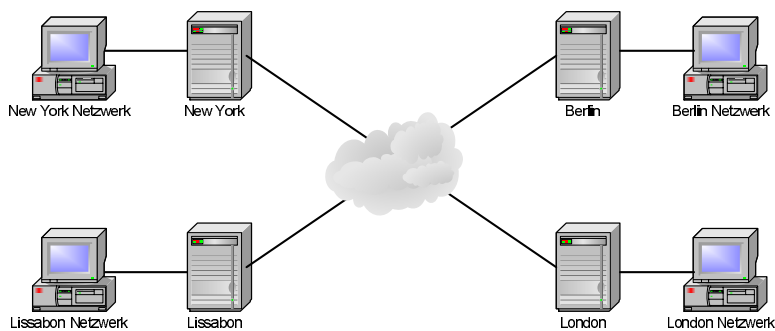


Abbildung 1.14 Netzförmiger Aufbau eines VPNs

### 1.6.4 Anbindung von Telearbeitsplätzen an einen Standort

Viele Firmen und ihre Mitarbeiter entdecken heute die Möglichkeiten der Telearbeit. Dabei greifen die Mitarbeiter mit ihrem Rechner von zuhause auf das Intranet der Firma zu. Aus Kostengründen werden hierzu immer mehr Internetverbindungen eingesetzt. So können auch DSL Bandbreiten für den Zugang genutzt werden.

Derartige Verbindungen über das Internet müssen jedoch sicher und geschützt aufgebaut werden. Hierfür eignen sich idealerweise VPN Lösungen auf der Basis von IPsec. Die Einwahl erfolgt über einen lokalen Internetdiensteanbieter (Internet Service Provider, ISP). Dann wird ein IPsec Tunnel zu einem VPN Gateway der Firma aufgebaut und der Zugang hergestellt. Abbildung 1.15 zeigt ein derartiges Szenario.

Derartige Lösungen werfen gegenüber den oben bereits besprochenen Szenarien weitere neue Probleme auf, die hier kurz mit einem Stichwort erwähnt werden sollen:

- Keine statischen IP Adressen. Der Telearbeiter erhält bei seiner Einwahl bei seinem ISP eine beliebige IP Adresse. Die IP Adresse kann daher nicht zur Authentifizierung genutzt werden.

- Benutzerauthentifizierung bei der Anmeldung. Da nun das VPN durch einen Benutzer aufgebaut wird, wird auch häufig eine Authentifizierung gefordert, die einer Anmeldung entspricht.
- Unter Umständen eine Network Address Translation durch den ISP. Die IPsec Protokolle überprüfen die verwendeten IP Adressen der Kommunikation. Änderungen dieser IP Adressen führen häufig zu Problemen.
- Zuweisung einer IP Adresse aus dem internen Netz zur einfachen Kommunikation. Damit sich der Telearbeiter anschließend im Netz normal bewegen kann, soll ihm häufig eine virtuelle IP Adresse aus dem internen Netz zugewiesen werden.

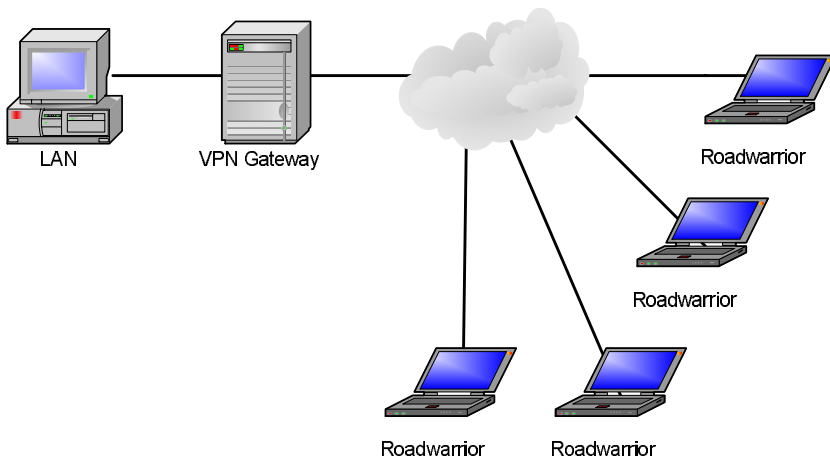


Abbildung 1.15 Zugriff von Telearbeitern auf ein Intranet (Roadwarrior)

Diese Punkte werden im weiteren in diesem Buch näher erläutert und Lösungen werden vorgestellt.

### 1.6.5 Anbindung von Außendienstmitarbeitern (Roadwarrior) an einen Standort

Dieses Szenario stellt im Grunde eine Kopie des letzten Szenarios dar. Auch hier besteht die Notwendigkeit, dass ein Außendienstmitarbeiter abends aus seinem Hotelzimmer seine Datenbanken mit den zentralen Firmendatenbanken synchronisieren möchte. Auch hier bietet sich ein VPN an. Der Außendienstmitarbeiter kann sich von seinem Telefonanschluss im Hotel bei einem lokalen oder nationalen ISP einwählen und so eine Internetverbindung aufbauen. Sie kann er dann anschließend nutzen um einen Tunnel zum Gate-

way der Firma aufzubauen. Über diese verschlüsselte Verbindung können dann alle Informationen ausgetauscht werden.

Dieses Szenario ist jedoch mit denselben Problemen behaftet, wie das Szenario mit den Telearbeitern.

## 1.6.6 Absicherung eines Wireless LAN

Wenn heute neue lokale Netzwerke (LAN) implementiert werden, so werden sie immer häufiger als Wireless LAN (WLAN) ausgeführt. WLANs können ohne Umbauarbeiten und dadurch verursachte Ausfallzeiten sehr einfach und schnell aufgesetzt werden. Mietverträge oder Denkmalschutz können die möglichen baulichen Änderungen bei einer Vernetzung stark einschränken. WLANs benötigen keine baulichen Veränderungen und können bereits mit einer Bruttobandbreite von 54 Mbit/s Daten transferieren. Für den Schutz dieser Daten wurde die Wired Equivalent Privacy (WEP) als Standard geschaffen. Sie verschlüsselt die übertragenen Daten mit 40 Bit oder 104 Bit<sup>6</sup>. Dieser Verschlüsselungsmechanismus wurde bereits Mitte 2001 von Scott Fluhrer, Itsik Mantin und Adi Shamir geknackt ([http://www.crypto.com/papers/others/rc4\\_ksaproc.ps](http://www.crypto.com/papers/others/rc4_ksaproc.ps)). Die entsprechenden Werkzeuge wurden kurze Zeit später als Open Source Werkzeuge zur Verfügung gestellt (AirSnort: <http://airsnort.shmoo.com/>; WEPCrack: <http://wepcrack.sourceforge.net/>). Die Verschlüsselung kann daher nicht als ausreichend sicher angesehen werden.

Ein Wireless LAN kann jedoch recht gut mit IPsec geschützt werden. Hierzu ist nur wenig mehr Aufwand als beim weiter oben beschriebenen Stern Szenario erforderlich. Ein Beispielprojekt, das dies in universitärem Rahmen durchführt, ist das MOPO Projekt: <http://mopoinfo.wlan.informatik.uni-freiburg.de/>. Hier ist ein etwa 20 Accesspoints umfassendes WLAN aufgebaut worden, welches eine Authentifizierung des Benutzers mit x509 Zertifikaten durchführt und anschließend den Aufbau eines verschlüsselten IPsec Tunnels ermöglicht.

## 1.6.7 Opportunistische Verschlüsselung

Die opportunistische Verschlüsselung ist eine neue Eigenschaft, die bisher nur unter Linux mit FreeS/WAN zur Verfügung steht (siehe Kapitel 5, »FreeS/WAN«). Hierbei ist Linux in der Lage den VPN Tunnel bei Bedarf und technischer Verfügbarkeit aufzubauen. Dazu ermittelt das VPN Gate-

---

6. Die Länge des Schlüssels wird teilweise unterschiedlich angegeben. Bei beiden Schlüssellängen wird zusätzlich ein 24 Bit langer Initialisierungsvektor zusätzlich verwendet. Daraus resultieren dann 64 Bit beziehungsweise 128 Bit.

way mit Hilfe des DNS Dienstes ob der entsprechende Kommunikationspartner möglicherweise durch ein VPN Gateway geschützt wird. Ist dies der Fall, so baut das Linux VPN Gateway zu dem entsprechenden zweiten VPN Gateway einen IPsec Tunnel auf und überträgt die Daten verschlüsselt. Steht kein VPN Gateway zur Verfügung, so werden die Daten in Klartext übertragen.

Diese Funktion ermöglicht den Aufbau von VPN Verbindungen mit beliebigen Partnern, die die entsprechenden Informationen in ihren DNS Servern hinterlegen. So kann eine schrittweise Migration erfolgen.

Das ist zum Beispiel sinnvoll, wenn innerhalb eines Netzwerkes die Kommunikation auf IPsec umgestellt werden soll. Es ist meist nicht möglich über Nacht die Konfiguration auf allen Systemen zu modifizieren und anzupassen. Bei opportunistischer Verschlüsselung ermitteln die Systeme selbst, mit welchen Systemen sie eine verschlüsselte Verbindung aufbauen können.

