Seminarband

Protokolle und Methoden zum Datenschutz im Internet

16.-18.2.1998

Lehrstuhl Informatik I
Universität Dortmund
Juni 1998

Veranstalter:

Prof. Dr. Bernd Reusch
Dipl.-Inform. Sascha Dierkes
Dipl.-Inform. Lars Hildebrand

Inhaltsverzeichnis

Tabellen- und Abbildungsverzeichnis

Datenschutz und Datensicherheit: Grundlagen und Motivation

Simone Weichert

1 Einleitung und Motivation

Durch die fortschreitende Technisierung ist es möglich geworden, große Datenmengen auf relativ kleinem Raum unterzubringen und mit Hilfe von Programmen gezielte Auswertungen zu machen. Dadurch werden immer mehr persönliche wie auch anonyme Daten gespeichert und es stellt sich die Frage wie diese Daten vor Mißbrauch zu schützen sind.

In einer KES-UTIMACO-Sicherheitsstudie wurde die Meinung der DV-Anwender zur Sicherheitsfrage untersucht [Hun96]. 183 Unternehmen wurden zum Stand der Sicherheits- und Informationstechnik befragt. Diese Unternehmen hatten in dieser Zeit einen relativ guten Sicherheitsstand. Es ergaben sich folgende Ergebnisse:

- Hauptmangel waren Irrtum und Nachlässigkeit von Mitarbeitern, sowie software-bedingte- und hardware-bedingte technische Defekte. Dabei kommt dem Irrtum und der Nachlässigkeit von Mitarbeitern und software-bedingten Defekten in Zukunft eine noch größere Rolle zu, wohingegen die hardware-bedingten Defekte abnehmen werden.
- Zur Befragung, ob der eigene Sicherheitsstandard in ihrer Firma nach ihrer Meinung ausreichend sei, antwortete nur knapp ein Drittel mit „ja“. D.h. das Problem der Informationssicherheit ist zwar bekannt und ist auch in vielen Unternehmen nicht unbedeutend, es hat aber keinen vorrangigen Stellenwert.
- Als Grund, warum der Sicherheitsstandard in den Firmen nicht verbessert wurde, wurde in der Hauptsache „Mangel an Bewußtsein“, „fehlende kompetente Mitarbeiter“ und „Geldmangel“ angegeben.
- Es ergab sich, daß viele Sicherheitsmaßnahmen und Schutzmöglichkeiten den Unternehmen angeboten werden bzw. schon verfügbar sind, sie jedoch vielfach einfach nicht genutzt würden.

1.1 Was ist Datenschutz?

Datenschutz bezieht sich in erster Linie nicht auf den Schutz der Daten, sondern soll vielmehr die Personen schützen, deren Daten gespeichert oder im Umlauf sind. Beispielsweise sollte es einem Arbeitgeber nicht möglich sein, an die Krankenakte der Person zu gelangen, die sich um eine neue Arbeitsstelle in seinem Unternehmen bewirbt. Im Sinne des Bundesdatenschutzgesetzes versteht man unter Datenschutz „den Schutz personenbezogener Daten vor Mißbrauch“. Der beste Datenschutz besteht

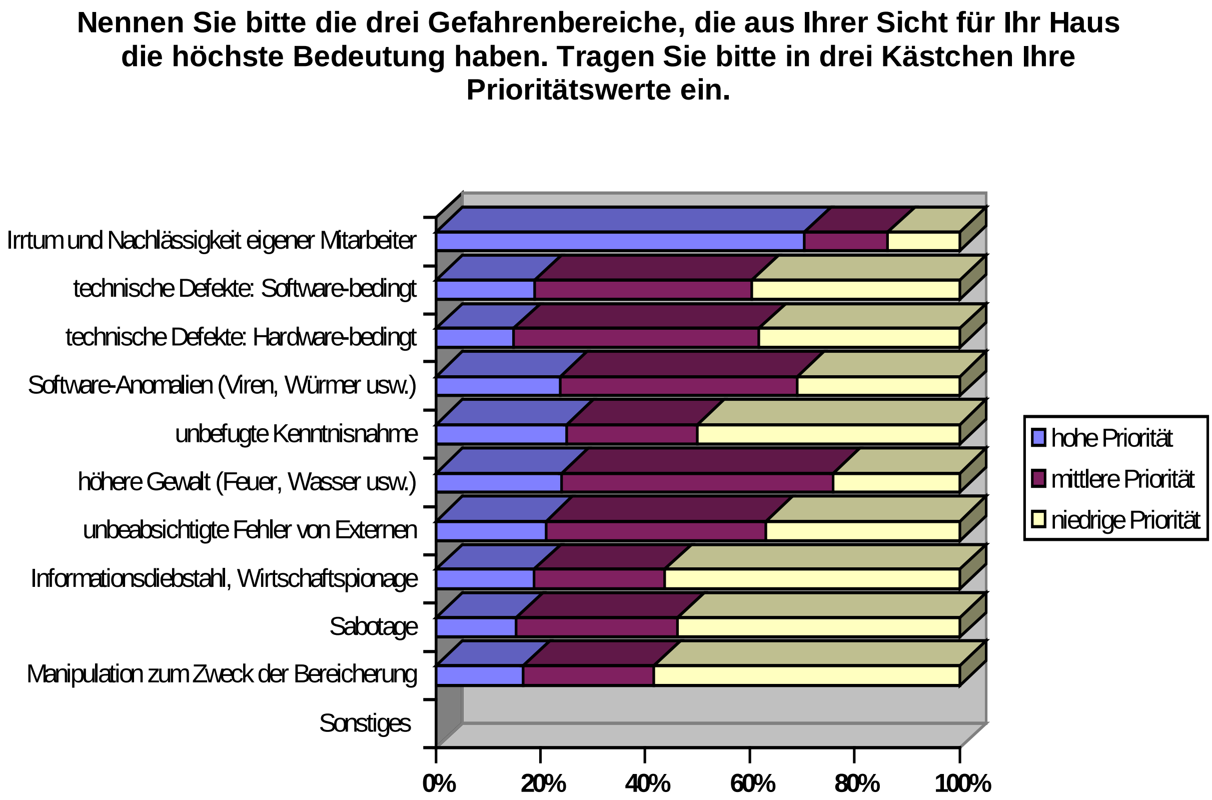

Abb. 1: KES/UTIMACO-Sicherheitsstudie, [Hunn96].

jedoch darin, Daten erst gar nicht zu erheben. Je mehr Daten vorhanden sind um so schwieriger ist es diese Daten zu schützen.

Die drei wichtigsten Stichwörter, die im Zusammenhang mit dem Datenschutz immer wieder genannt werden sind:

- Vertraulichkeit
- Verfügbarkeit
- Integrität.

Sind Informationen nur authorisierten Benutzern zugänglich und wird ein unbefugter Zugang zum System verhindert, so besitzen die Informationen eine hohe Vertraulichkeit. Gespeicherte, wie auch übertragene Daten, können Vertraulichkeit erfordern. Werden Informationen dennoch „abgehört", so ist der Verlust der Vertraulichkeit nicht immer nachzuweisen oder wird teilweise gar nicht bemerkt, da die Originaldaten beim „Abhören" unverfälscht bleiben.

Integrität beschreibt die Unverfälschtheit und Korrektheit von Informationen. Bei der Verarbeitung, der Übertragung, wie auch bei der einfachen Speicherung muß sichergestellt sein, daß Informationen weder absichtlich, noch unbeabsichtigt verändert oder verfälscht werden. Zur Änderung von Informationen ist nur der authorisierte Benutzer oder spezielle authorisierte Personen (z. B. der Systemadministrator) berechtigt.

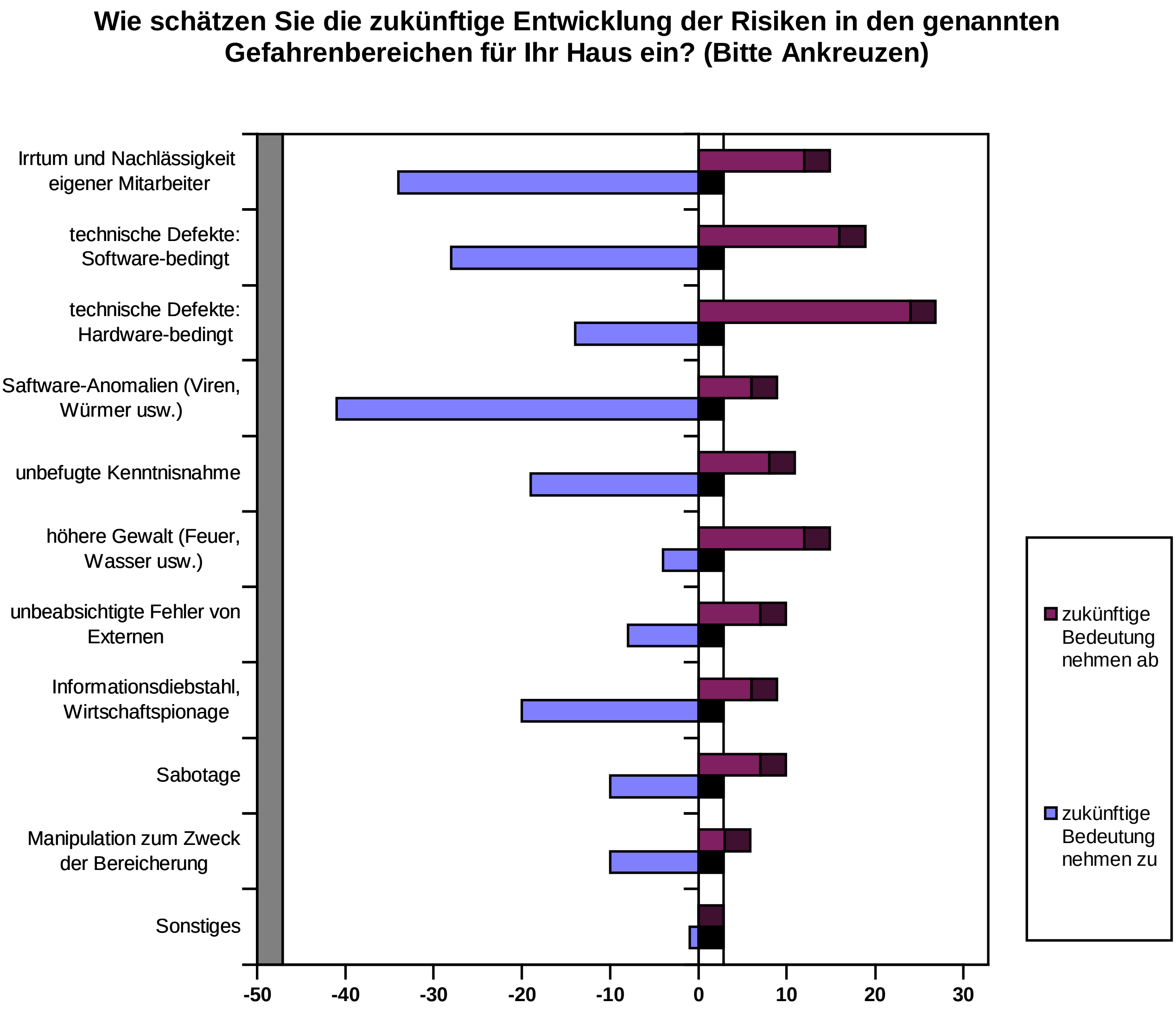

Abb. 2: KES/UTIMACO-Sicherheitsstudie, [Hunn96].

Unter der Verfügbarkeit versteht man den ungehinderten Zugang der Benutzer zu seinen Daten. Das System sollte jederzeit in der Lage sein Funktionen auszuführen und Informationen bereitzustellen. Das System soll also dauerhafte Bereitschaft und Funktionalität gewährleisten.

1.2 Was ist Datensicherheit?

Datensicherheit beinhaltet alle Maßnahmen um Daten vor unzulässigen Zugriffen zu schützen. Zusätzlich soll gewährleistet werden, daß die Daten auch bei Fehlfunktion oder Zerstörung der Rechenanlage gesichert sind.

Vor folgenden Bedrohungsformen sind die Daten zu schützen:

- Höhere Gewalt,
- aktive Angriffe,

- passive Angriffe,
- unbeabsichtigte Benutzerfehler,
- Hardware- und Softwarefehler.

Unter Höherer Gewalt versteht man unbeeinflußbare Dinge wie beispielsweise Feuer, Wasser oder Blitzeinschlag. Dagegen kann man sich nur bedingt schützen, z. B. durch feuerfeste Türen der Computerräume, automatische Sprinkleranlagen gegen Feuer oder Spannungsschutzeinrichtungen bei Blitzeinschlag.

Versucht ein Angreifer sich gezielt unberechtigten Zugang zu Informationen zu verschaffen, so versteht man darunter einen Angriff auf ein System. Es wird zwischen aktiven und passiven Angriffen unterschieden. Ein aktiver Angriff bezieht sich auf gespeicherte und / oder übertragene Daten, ein passiver Angriff nur auf übertragene Daten. Aktive Angriffe dienen dem unberechtigten Informationsgewinn wobei auch Daten verändert werden. Passive Angriffe dienen dem unberechtigten Informationsgewinn, verändern die Daten jedoch nicht. Beispielsweise wäre das Abhören von übertragenen Informationen mit dem Ziel an vertrauliche Daten zu gelangen oder das Abhören von Teilnehmeridentitäten ein passiver Angriff.

Weiterhin spielen unbeabsichtigte Benutzerfehler und Hardware- und Softwarefehler eine Rolle bei der Verletzung der Datensicherheit.

2 Kurze Einführung in die rechtlichen Rahmenbedingungen der neuen Informations- und Kommunikationsdienste

Das Anbieten von Informationen auf dem Internet fällt in den Regelungsbereich der nationalen Datenschutzgesetze und -regelungen. So ist es z. B. einem deutschen Anbieter eines World-Wide-Web-Servers verboten, ohne Wissen des Benutzers die vollständigen Angaben über die abgerufenen Seiten und heruntergeladenen Dateien zu speichern. Zusätzlich ist es in manchen Ländern Pflicht sich als Informationsanbieter bei der nationalen Datenschutzbehörde anzumelden.

In einer kurzen Übersicht werden im Folgenden die rechtlichen Rahmenbedingungen der neuen Informations- und Kommunikationsdienste eingehen. Spezieller wird der Vortrag „Rechtliche Regelung des Datenschutzes in Deutschland“ auf dieses Thema eingehen.

Das „Informations- und Kommunikationsdienste-Gesetz“ soll hierfür einen bundesgesetzlichen Rahmen schaffen. Das Gesetz soll zum einen grundlegende rechtliche Bedingungen für Angebot und Nutzung der Informations- und Kommunnikationsdienste insbesondere das Internet festlegen, zum anderen bestehende Bundesgesetze an die technische und wirtschaftliche Entwicklung in diesem Bereich anpassen. Das Gesetz befaßt sich in der Hauptsache mit folgenden Punkten:

- Bestimmung der Verantwortlichkeit, d.h. regelungsbedürftig sind Fragen der Verhaltenspflichten und damit der Verantwortlichkeit der Beteiligten

- Grundsatz der Zugangsfreiheit
- Grundsatz der Nichterhebung personenbezogener Daten, d.h. Prinzip der Datenvermeidung
- Unzulässige Angebote, Jugendschutz
- Digitale Signaturverfahren

Die neuen Informationsdienste erfordern darüber hinaus Anpassungen und Ergänzungen bestehender Bundesgesetze, wie der folgenden:

- Urheberrechtsgesetz (Schutz von Datenbanken und geistigen Eigentums)
- Strafgesetzbuch
- Bundesdatenschutzgesetz

Dies ist nur ein Auszug der rechtlichen Veränderungen die im speziellen das Internet mit sich bringt.

3 Datenschutz und Datensicherheit im Internet

Im Vorfeld eine kurze Bemerkung: „Datenschutz im Internet gibt es eigentlich nicht, da eine umfassende Kontrolle des Netzes nicht möglich wäre“ [URL-1]. Normalerweise sind bei der Verarbeitung personenbezogener Daten einzelne Behörden oder Unternehmen, die mit den Daten ihrer Kunden umgehen, dafür auch verantwortlich. Im Internet gibt es aber keine Einrichtung, der eine solche Gesamtverantwortung zugewiesen wäre. Daher muß jeder Benutzer ein gewisses Vertrauen in die Sicherheit des gesamten Netzes setzen, das bedeutet in jeden einzelnen Knoten des Netzes, egal in welchem Land er angesiedelt ist und wie er verwaltet wird. Da dies jedoch sehr fraglich ist, sollte sich jeder Nutzer so gut es geht und je nachdem wie seine Daten es verlangen schützen.

Der Wunsch ans „Netz“ angeschlossen zu werden, wächst seit einiger Zeit unter privaten wie auch öffentlichen Nutzern. Hierbei sollen nicht nur Daten zum Informationsgewinn dienen, sondern teilweise auch Daten zur Nutzung bereitgestellt werden. Damit stellt der Anschluß ein erhebliches Risiko beim Datenschutz und der Datensicherheit dar. Das Internet wurde ursprünglich nicht unter Sicherheitsaspekten entwickelt, insbesondere gibt es Schwächen in den Protokollen zur Datenübertragung (siehe Vortrag „Grundlagen des Internets und des TCP/IP Stacks“) und soll jedem weltweiten Zugang ermöglichen (z. Zt. mehr als 40 Millionen Inernet-Teilnehmer). Wenn keine besonderen Schutzmaßnahmen getroffen sind, kann sich ein Angreifer oft mit geringen Aufwand Zugang zu fremden Betriebsmitteln verschaffen und dabei unberechtigt Daten lesen, manipulieren oder zerstören.

3.1 Welche Daten werden erhoben?

Jeder Benutzer eines Internet-Dienstes hinterläßt eine „Datenspur“. Durch diese Spur läßt sich ein umfassendes Kommunikationsprofil und damit auch evtl. ein Benutzerprofil erstellen.

Man unterscheidet drei Kategorien von personenbezogenen Daten:

- Stammdaten: Sie werden von Internet-Anbieter erhoben und enthalten Name, Adresse sowie Login-Kennung und evtl. Bankverbindung des Benutzers. Teilweise werden auch Informationen

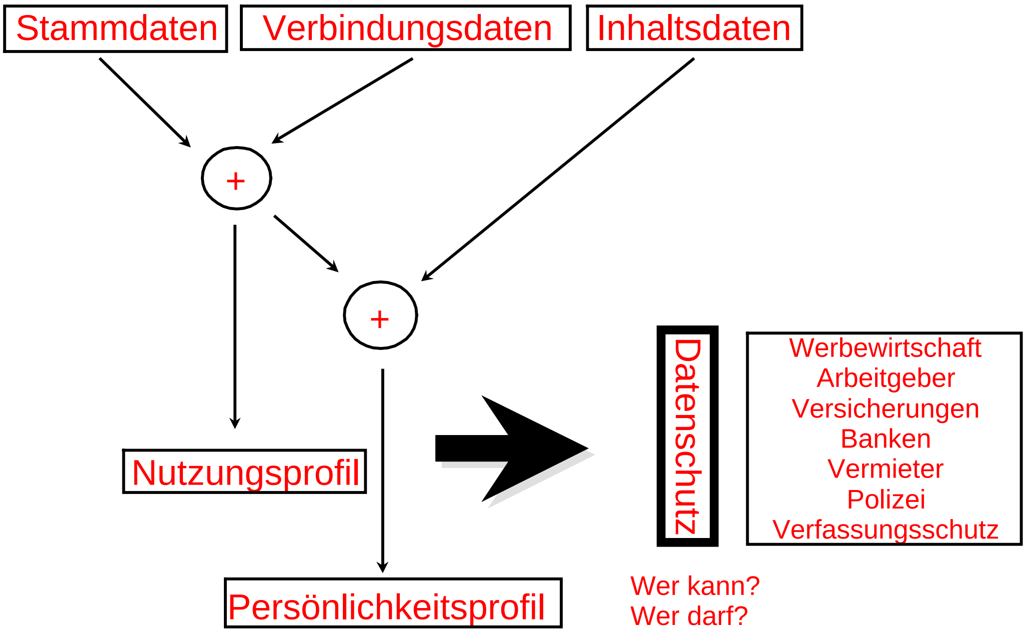

Abb. 3: Zusammenhänge der personenbezogenen Daten, [URL-1].

über den Status (Student, Arbeitsloser ect.) erhoben. Diese Daten sollten lediglich dem Internet-Provider zur Verfügung stehen.

- Verbindungsdaten: Sie geben an wer mit wem und wieviel Daten austauscht. Diese Daten, wie auch die folgenden „Inhaltsdaten" sind abhängig von der Nutzung des Internets. Geht der Benutzer beispielsweise in ein Forum, so werden seine Daten dort zeitweise gespeichert. Sieht man von den Daten ab die beim Sender oder Empfänger des anderen vorliegen, so können auch ohne viel Aufwand auf jedem Rechner entlang des Übertragungsweges oder direkt auf der Leitung Daten mitgehört werden.
- Inhaltsdaten: Sie beschreiben den eigentlichen Inhalt einer Nachricht, wie z.B. e-mail oder Artikel in einem Diskussionsforum.

Nimmt man die oben beschriebenen Daten nun zusammen, so kann ein Nutzungs- bzw. Persönlichkeitsprofil erstellt werden.

Die Abbildung zeigt schematisch an welchen Stellen im Netz die Erstellung von Nutzerprofilen und Persönlichkeitsprofilen möglich wäre.

Beim Client ist das Benutzerverhalten direkt auf dem Rechner zu beobachten. So war es z. B. beim Browser Netscape 2.0, mit den Scriptsprachen Livescript und Javascript, dem Anbieter von Internetseiten möglich, Informationen über die Konfiguration des Rechners und das Rezeptionsverhalten seines Nutzers zu erhalten.

Der Provider hat Informationen über die Identität des Nutzers. Er könnte Aufzeichnungen über das Abrufverhalten des Nutzers machen, und auch Inhaltsdaten einsehen. Gegen das Abhören von Inhaltsdaten kann sich der Nutzer weitgehend durch Verschlüsselung schützen, die Verbindungsdaten

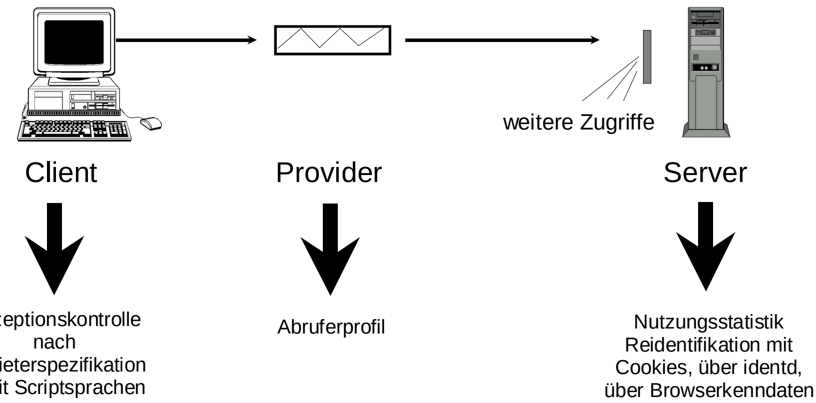

Abb. 4: Angriffspunkte im Netz, [URL-1].

fallen aber auf jeden Fall an, da der Provider sie für die Verbindungserstellung und teilweise zur Abrechnung benötigt,

Beim Server können Daten über die abgerufenen Seiten gesammelt werden. Normalerweise geschieht dies nur aus statistischen Zwecken. Mit Hilfe einiger Kniffe ist es jedoch auch möglich die Verbindung zum Benutzer herzustellen. Beispiele hierfür sind:

- Cookies
- Identd-Protokoll
- Verräterische Browserkenndaten.

Jedesmal, wenn man eine Webseite aufruft, wird vom Browser eine Anfrage an die entsprechende Seite geschickt und die Seite wird übertragen. Danach ist die Übertragung zunächst abgeschlossen und der Kontakt wird unterbrochen. Wird nun eine Folgeseite aufgerufen, so wird der Webserver mit einer neuen Anfrage kontaktiert und es wird die nächste Seite übertragen. Ein Cookie ist eine kleine Datei mit Textinformationen, die von einem Webserver an den Webbrowser übertragen wird. In ihr kann man beispielsweise kurze Informationen von einem Kontakt zum nächsten Kontakt zwischen Browser und Server speichern, so daß nicht alle Informationen wieder erneut ausgetauscht werden müssen. Beispielsweise finden Cookies auch Anwendung in sogenannten virtuellen Einkaufskörben. Nun können Cookies nicht nur die Verbindung erleichtern, sondern sie können auch mißbraucht werden. So kann ein Server Cookies dafür verwenden, um einen Benutzer beim Betreten einer Startseite eindeutig zu markieren und seine Zugriffe auf die Folgeseiten aufzeichnen.

Das Identd-Protokoll diente ursprünglich dazu den Server vor Mißbrauch durch Nutzer zu schützen. Es ermöglicht dem Server die Identität des Nutzers abzufragen. Dies kann natürlich auch gegen den Nutzer angewendet werden. Allerdings ist es dem Provider bei diesem Protokoll möglich, zu entscheiden wieviel er von den Informationen über den Nutzer enthüllen will.

Die schlechteste Methode der Benutzeridentifikation ist die Identifikation mit Hilfe von Browserkenndaten wie dem verwendeten Browserprogramm und seiner Seriennummer.

3.2 Sicherheitsrisiken im Internet

Es gibt eine Vielzahl von Sicherheitsrisiken im Internet. Der folgende Abschnitt ist ein Auszug daraus:

- Da das Internet die schnelle Übertragung von großen Informationsmengen auf viele andere an das Netzwerk angeschlossene Computersysteme ermöglicht, können personenbezogene Daten in andere Länder übertragen werden. Nicht jedes Land verfügt jedoch über das gleiche oder zumindest über ein angemessenes Datenschutzniveau. Man kann also ein Land wählen, daß keine oder nur eine geringe Datenschutzgesetzgebung hat und von dort Informationen anbieten auf die dann jeder weltweit zugreifen kann.
- Im Internet gibt es Tausende von speziellen news-groups in denen Artikel verbreitet werden. Beinhalten die Artikel Informationen über Dritte, so gibt es für den Betroffenen kaum eine Möglichkeit dies zu unterbinden.
- Wie den wenigsten bekannt ist, werden Nutzerkennung, wie auch das Paßwort bei gängigen Diensten über das lokale Netz wie z.B. das Ethernet im Klartext übertragen. Mit Abhörprogrammen (*Packet Sniffer*) können so leicht Knotenpunkte mitgehört werden und man bekommt Nutzerkennungen mit Paßwort geliefert.
- Ein weiteres Problem ist das sogenannte *IP-Spoofing*. Da bei vielen Internet-Diensten die Authentisierung der Rechner nur über die IP-Nummer des Nutzers erfolgt, kann ein Angreifer IP-Pakete mit gefälschten Absenderadressen verschicken. Bemerkt das System die gefälschte Adresse nicht, so kann man sich so Zugriff auf Daten verschaffen, für die man unter normalen Umständen keine Zugriffsrechte hätte. Dies ist auch möglich in dem man die Originalpakete abfängt und durch eigene ersetzt oder man leitet die Originalpakete einfach um. (siehe auch „Sicherheit auf Transport- und Sitzungsschicht“ und „Angriffsstrategien“).
- Natürlich können auch private Nachrichten wie e-mails mitgelesen werden, solange sie nicht verschlüsselt werden. Auch diese lassen sich verändern oder abfangen. „Adreßsammler“ nutzen dies auch um unaufgefordert Werbung zu verschicken.
- Auch der Telnet-Dienst und FTP wird häufig als Ausgangsbasis für Angriffe genutzt. So stellen schlecht gewartete FTP-Server ein Risiko dar. Fehler in der Konfiguration können dazu führen, daß es z. B. möglich wird die Datei mit sämtlichen verschlüsselten Paßwörtern herunterzuladen. Diese kann der Angreifer dann in aller Ruhe entschlüsseln und danach weitere Angriffe starten. Hat der Benutzer eines FTP-Servers die Möglichkeit auch Dateien in Verzeichnisse abzulegen, so kann ein Server schnell zum Umschlagplatz für Raubkopien werden.
- Beim Surfen im WWW können zahlreiche Daten über den Anwender und sein Verhalten gespeichert werden. Durch den Aufruf der verschiedenen Seiten kann ein Persönlichkeitsprofil des Nutzers erstellt werden.

- *Finger* ist ein Werkzeug zur Suche nach Informationen über Personen und Rechner. Damit können personenbezogene Daten gefunden werden und Nutzerkennungen in Erfahrung gebracht werden, so daß ein gezielter Angriff möglich wird. Bekannt geworden ist es durch den *Internet-Wurm* (siehe Vortrag „Angriffsstrategien“). Es wird hierbei ausgenutzt, daß beim Aufruf von Finger übergebene Parametern in einen Puffer fester Länge geschrieben wurde, die Daten die nicht mehr in den Arbeitsspeicher paßten überschrieben den Stack im Arbeitsspeicher, wo sie als Programmcode behandelt und ausgeführt wurden. Wählt man nun geschickt eine zu übergebene Zeichenreihe, so kann man beliebigen Code zur Ausführung bringen.

Man sollte sich also nur ans Netz anschließen, wenn vorher eine eingehende Analyse und Bewertung der damit verbundenen Risiken stattgefunden hat und die Gefahren durch technische und organisatorische Maßnahmen vermindert bzw. ausgeschaltet worden sind. Wie die Bewertung ausfällt ist natürlich auch immer von der Beschaffenheit der gefährdeten Daten abhängig.

3.3 Sicherheitsvorkehrungen

Folgende Sicherheitsvorkehrungen sind zu empfehlen:

- Es sollte zunächst geprüft werden, ob ein Anschluß überhaupt nötig ist. Falls ja, so sollte abgewogen werden, ob bei Vorhandensein eines eigenen Netzes das gesamte Netz an das Internet angeschlossen werden soll oder ob es nur bedingt anschließbare Teile geben soll.
- Je nach Beschaffenheit der zu schützenden Daten, sollte der Anschluß nur nach Vorhandensein eines schlüssigen Sicherheitskonzeptes erfolgen.
- Der Schutz von personenbezogenen Daten, die auf vernetzten Systemen verarbeitet werden, ist durch geeignete Firewall-Systeme sicherzustellen, die eine differenzierte Kommunikationssteuerung unterstützen. Firewall wird eine Schwelle zwischen zwei Netzen genannt. Der Sinn des Firewalls ist es nur die Aktivitäten zwischen den Netzen zu ermöglichen die auch zulässig sind. Besonders gestaffelte Firewalls sind für Netze die aus einer Vielzahl von Teilnetzen mit unterschiedlicher Datensensibilität bestehen gut geeignet. (Mehr zu diesem Thema im Vortrag „Firewalls“).
- Auch beim Einsatz von Firewalls bleibt ein Restrisiko, so sollten sensible Daten nur verschlüsselt übertragen werden. Außerdem sind natürlich Paßwörter und Authentifikationsdaten notwendig (siehe auch insbesondere Vorträge über „Kryptographie“, „Quantenkryptographie“)
- Ein weiterer Punkt zur Sicherheit beizutragen ist es, Personal einzusetzen die sich um die Sicherheit des Netzes bemühen. Diese Personen müssen genauso spezialisiert sein wie die zu erwartenden Angreifer.

4 Vorstellung des *Orange Book*

Das *Orange Book* wurde vom Department of Defense, USA herausgegeben. Es ist ein Kriterienkatalog für die Beurteilung von Betriebssystemen und wendet sich:

a) An Benutzer von Computersystemen, um ihnen die Risiken der Verwaltung von sensiblen Daten mit dem System darzustellen

b) An die Hersteller von Computersystemen, um ihnen einen Leitfaden bei der Herstellung zu geben. Es soll ihnen zeigen, was bei der Verarbeitung von sensiblen Daten zu beachten ist

c) An diejenigen, die grundlegende Richtlinien für die notwendigen spezielle Sicherheitsvorkehrungen suchen.

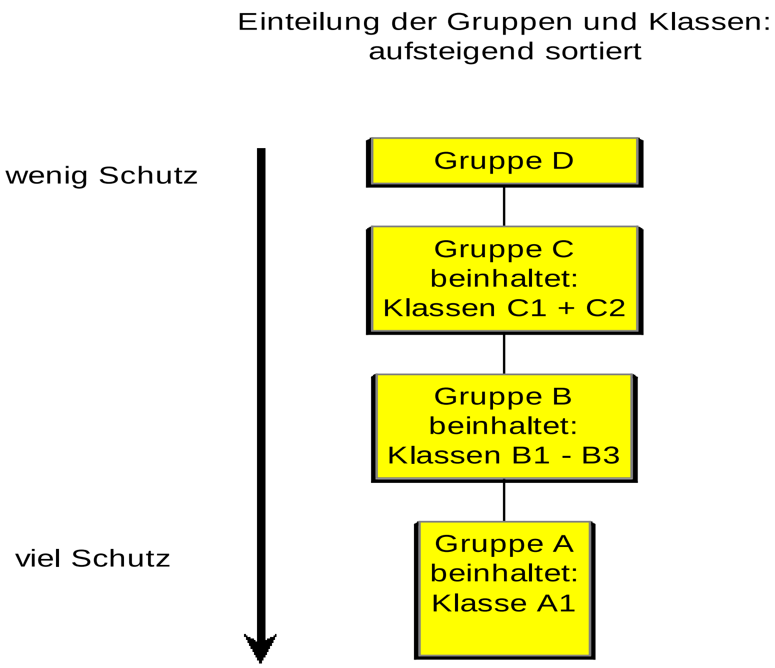

Abb. 5: Einteilung des Orange Book in Klassen.

Das Buch teilt die verschiedenen Schutzkriterien in Sicherheitsgruppen D, C, B und A auf. Innerhalb dieser Gruppen werden noch einmal verschiedene Klassen (C1,C2,B1,B2,B3 und A1) unterschieden. Die Gruppen werden im folgenden nach ihren Sicherheitsanforderungen aufgelistet, wobei mit den geringsten Anforderungen begonnen wird und die Gruppen, was die Sicherheitsanforderungen angeht, aufsteigend sortiert sind. Sie unterscheiden sich wie folgt:

Gruppe D:

Diese Gruppe besteht nur aus einer Klasse und bietet minimalen Schutz. In diese Klasse werden diejenigen Systeme eingeordnet die auf Grund ihrer Sicherheitsanforderungen keiner anderen Gruppe gerecht werden konnten.

			Klasse						
			D	C1	C2	B1	B2	B3	A1
Sicherheits-grundsätze	Benutzerbestimmbare Zugriffskontrolle		-	+	+	=	=	+	=
	Wiederverwendung von Objekten		-	-	+	=	=	=	=
	Kennzeichen		-	-	-	+	+	=	=
		Integrität der Kennzeichen	-	-	-	+	=	=	=
		Ausgabe gekennzeichneter Info.	-	-	-	+	=	=	=
		Sensitivitätskennzeichen der Subjekte	-	-	-	-	+	=	=
		Gerätekennzeichen	-	-	-	-	+	=	=
	Vorgeschriebene Zugriffskontrolle		-	-	-	+	+	=	=
Nachweis-führung	Identifizierung und Authentifizierung		-	+	+	+	=	=	=
		Vertrauenswürdiger Zugriffspfad	-	-	-	-	+	+	=
	Revision		-	-	+	+	+	+	=
Garantie	Funktionsgarantie	Systemarchitektur	-	+	+	+	+	+	=
		Systemintegrität	-	+	=	=	=	=	=
		Analyse der verdeckten Kanäle	-	-	-	+	+	+	+
		Management der vertrauenswürdigen Betriebstelle	-	-	-	+	+	+	=
		Verstrauenswürdige Wiederherstellung	-	-	-	-	-	+	=
	Lebensdauer-garantie	Sicherheitsprüfung	-	+	+	+	+	+	+
		Entwurfsspezifikation und –verifikation	-	-	-	+	+	+	+
		Konfigurationsmanagement	-	-	-	-	+	=	+
		Vertrauenswürdige Verteilung	-	-	-	-	-	-	+
Dokumen-tation	Benutzerhandbuch über Sicherheitsvorrichtungen		-	+	=	=	=	=	=
	Handbuch über die vertrauenswürdige Betriebsstelle		-	+	+	+	+	+	=
	Prüfdokumentation		-	+	=	+	+	=	+
	Entwurfsdokumentation		-	+	=	+	+	+	+

- Keine Anforderungen an diese Klasse

\+ neue oder gegenüber der nächstniedrigeren Klasse erweiterte Anforderungen

= keine Zusatzanforderungen (gleiche Anforderungen wie in der nächstniedrigeren Klasse)

Abb. 6: Überblick über das Orange Book, [Heid96].

Gruppe C:
Diese Gruppe besteht aus mehreren Schutzklassen. Es gibt die sog. *Trusting Computer Base* (TCB). Sie ist die Gesamtheit der Mechanismen in einem Computersystem die notwendig sind um die Sicherheit in einem System zu gewährleisten. In der Gruppe C wird die Verantwortlichkeit von Subjekten und deren Aktionen die sie auslösen, kontrolliert. Hier werden Benutzer voneinander getrennt und jeder bekommt eine individuelle Zugangsberechtigung. Es besteht also die Möglichkeit den Benutzern lesenden, schreibenden oder gar keinen Zugriff auf Objekte zuzuteilen.
Es ist außerdem möglich gewisse Ressourcen isoliert zu halten.
Gruppe B:
Auch diese Gruppe besteht aus mehreren Klassen. Zur TCB gibt es eine sogenannte *Security Policy*[1]. Sie beinhaltet die Gesetze, Regeln und Praktiken die befolgt werden müssen um die Sicherheit sensibler Daten zu gewährleisten. Die TCB soll das ausführen und gewährleisten, was in der *Security Policy* beschrieben und verlangt wird. Außerdem muß ein Beweis erbracht werden, daß das System auch tatsächlich die verlangten Sicherheitsansprüche erfüllt. Jedes Objekt erhält ein Kennzeichen in dem seine Schutzbedürftigkeit angegeben wird, also ob es sensible Daten enthält oder allgemein zugängliche. Diese Gruppe ist schon relativ sicher gegen Eindringlinge.
Gruppe A:
Sie stellt die höchsten Ansprüche an die Sicherheit. Diese Gruppe ist durch formale Sicherheitsüberprüfungsmethoden gekennzeichnet, so daß sichergestellt wird, daß das System effektiv sensible Daten schützen kann. Ausführliche Dokumentation der Sicherheitsvorkehrungen und Demonstration, daß TCB die Anforderungen auch tatsächlich in Design, Entwicklung und Implementation ausführt, sind notwendig.
In der folgenden Tabelle erfolgt eine Vorstellung der Entwicklungskriterien des Orange Book, die notwendig sind um Daten zu schützen. Dabei wird eine Einteilung in die verschieden Klassen vorgenommen.

5 Zusammenfassung

Abschließend sei zu bemerken, daß man vor dem Anschluß an das Internet eingehend prüfen sollte, ob der Anschluß tatsächlich von Nutzen ist. Ist ein Anschluß eines internen Netzes notwendig oder könnte auch ein isolierter Rechner zum Internetzugriff genügen? Wenn diese Fragen geklärt sind, muß man sich weiterhin überlegen welche Daten gefährdet sind und wie groß ihr Schutzanspruch ist. Nach diesen Kriterien müssen sich die erforderlichen Schutzmaßnahmen orientieren. Man sollte sich auf keinen Fall auf die „Ehrlichkeit“ der anderen Internetbenutzer verlassen, sondern sich ausreichend selbst schützen, d.h. auch veraltete Schutzmechanismen verbessern oder austauschen.

[1] engl. Sicherheitspolitik

Bei der Benutzung des Internetzuganges durch mehrere Anwender sollte man sich nicht nur vor Angriffen aus dem Internet schützen, sondern auch internen Mißbrauch nicht außer Acht lassen.

6 Literatur- und Quellenverzeichnis

6.1 Printmedien

[Hunn96] G. Hunnius, „So schätzen DV-Anwender ihre Sicherheit ein“. *KES Zeitschrift für Kommunikations- und EDV-Sicherheit*. SecuMedia Verlag, Ingelheim, Nr.3, 1996.

[Heid96] B. Heidecke, Diplomarbeit: *„Analyse und Bewertung von Sicherheitskonzepten in Client/Server-Systemen am Beispiel von Novell NetWare“*, Dortmund, 1996.

6.2 Elektronische Dokumente

[URL-1] First Surf, Computer & Net - Die Datenschnüffler, „So einfach geht das“, (gesichtet 16.10.97):
`http://www.firstsurf.com/koeln1.html`

[URL-2] Berlin-Memorandum, „Datenschutz und Privatsphäre im Internet“, (gesichtet 16.10.97):
`http://www.datenschutz-berlin.de/diskus/13_19.html`

[URL-3] „Site Security Policy Handbook Working Group“, 7.1991 (gesichtet 18.10.97):
`http://sunsite.cnlab-switch.ch/ftp/doc/standart/rfc/12xx/1244`

[URL-4] Department of Defense, „Trusted Computer System Evaluation Criteria“ (orange book), 1996 (gesichtet 16.10.97):
`http://iaks.www.ira.uka.de/`
`ta/Security/Diverses/DoD_Security_OrangeBook.txt.gz`

[URL-5] Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie, „Rechtliche Rahmenbedingungen für neue Informations- und Kommunikationsdienste“, 2.5.1996, (gesichtet 16.10.97):
`http://www.iid.de/rahmen/eckwerte_bmbf.html`

[URL-6] Der Hamburgische Datenschutzbeauftragte, „Orientierungshilfe zu Datenschutzfragen des Anschlusses von Netzen der öffentlichen Verwaltung an das Internet“, *BDSG-Broschüre*, 1.12.95, (gesichtet 16.10.97):
`http://www.rewi.hu-berlin.de/`
`Datenschutz/DSB/HmbDSB/Material/intern.html`

Grundlagen des Internets und des TCP/IP Stacks

Oliver Schröder

1 Einleitung

Dieser Beitrag beschäftigt sich innerhalb eines Kompaktseminars zum Thema Protokolle und Methoden zum Datenschutz im Internet mit den Grundlagen des Internets. Das Internet wird in diesem Zusammenhang kurz vorgestellt und es wird danach eine Betrachtung des TCP/IP Stacks vorgestellt, um dann eine Abgrenzung zu anderen Protokollen vorzunehmen.

1.1 Eine kurze Geschichte des Internets

Ende der späten 60er Jahre begann in den USA die staatliche Unterstützung von Experimenten zur Vernetzung von Computern unter Aufsicht des Verteidigungsministeriums. Das Interesse an einer solchen Vernetzung resultierte aus dem Bedürfnis des Militärs, über ein Kommunikationsmittel zu verfügen, das auch unter ungünstigen Bedingungen mit hoher Zuverlässigkeit den Austausch von Daten ermöglichen würde. Dies sollte auch gewährleistet bleiben, wenn ein Teil des Systems ausfallen würde. Die ARPA (*Advanced Research Projects Agency*), seit 1972 *Defence* ARPA (*DARPA*), forcierte mit erheblichen finanziellen Mitteln die Entwicklung in diesem Bereich der Computertechnologie.

Neben der militärischen Anwendung eröffnete die Vernetzung von Computern im wissenschaftlichen und immer mehr auch im kommerziellen Bereich völlig neue Anwendungsmöglichkeiten. Gerade im Bereich der Wissenschaft sind für viele Probleme Rechner mit hoher Rechenleistung und ganz spezieller Software nötig. Diese Ausstattungen sind sehr teuer und können daher nicht für jede Einrichtung beschafft werden. Die Idee besteht also nun darin, daß mit Hilfe einer Rechnervernetzung möglichst vielen Wissenschaftler ein Zugang zu diesen Rechnern zu ermöglichen.

Der Vorteil eines Rechnernetzes besteht allgemein darin, daß alle im Netz vorhanden Ressourcen durch jeden Netzteilnehmer unabhängig von seinem Standort genutzt werden können. Es besteht die Möglichkeit der gemeinsamen Nutzung von unterschiedlicher Hardware, Programmen, Daten und Peripheriegeräten (*resource sharing*).

So gingen die ersten Bestrebungen dahin, daß man wenige an geographischen unterschiedlichen Punkten verteilte, verschiedenartige Rechner miteinander zu verbinden versuchte. Dies geschah über angemietete Leitungen. Diese Leitungen verfügten über eine Übertragungsrate von 50 Kbit/s.

Die ersten Herausforderungen waren:

- ein Subnetz aus Telefonleitungen und Vernittlungsknoten aufzubauen, dessen Zuverlässigkeit, Kapazität und Kosten das Resource Sharing erlaubte

- die erforderlichen Protokolle zu verstehen, zu gestalten und für die unterschiedlichen Rechnertypen zu implementieren, um die neuen Subnetze zur Kommunikation nutzbar zu machen

Ende 1969 war eine erste Implementierung von *telnet*, d. h. Durchführen von Sitzungen auf entfernten Rechnern und von *ftp*, d.h. Transfer von Dateien zwischen entfernten Rechnern, vorhanden. Damit war das *Arpanet* geboren. Aufgrund vieler Verbesserungen entwickelte sich das Arpanet von einem Laborexperiment zu einem funktionsfähigem System, in dem verschiedenartige Computersysteme untereinander verbunden waren. Allerdings waren diesen Hostrechnern [2] , da sie auf unterschiedlichen Systemen basierten, kleinere Rechner vorgeschaltet. Diese Rechner bezeichnet man als IMP´s (*Interface Message Processor*). Die eigentliche Vernetzung erfolgte durch diese IMP´s. Die IMP´s waren für den eigentlichen Transport der Daten zwischen den Hosts verantwortlich [SBGK94; S. 2 ff.].

1.2 Einführung des Protokolls TCP/IP

Jetzt mußte die Entwicklung eines Kommunikationsprotokolls voran getrieben werden, daß es ermöglichte die unterschiedlichen Hardwarearchitekturen der Hersteller in Einklang zu bringen. Der wichtigste Schritt in diese Richtung war die Entwicklung des TCP/IP Protokolls (*Transmission Control Protocol/Internet Protocol*), das von den verantwortlichen Forschern des NCC (*Network Control Center*) und des NIC (*Network Information Center*) in den Jahren 1973/1974 implementiert wurde. Es sollte das bis dahin verwendete NCP (*Network Control Protocol*) als Transportprotokoll ablösen. Mit TCP/IP wurde es möglich eigenständige Netzwerke so miteinander zu verbinden, daß jeder Hostrechner einen Netzwerkes mit allen anderen Rechnern des Netzes kommunizieren konnte. So waren die IMP´s nicht mehr notwendig, weil eine Möglichkeit der direkten Kommunikation geschaffen worden war. Es wurde großer Wert darauf gelegt, daß bei der Implementierung dieses Protokolls eine Unabhängigkeit vom Übertragungsmedium gewährleistet war. So ist das Galsfaserkabel dem Kupferkabel bei der Übertragungsgeschwindigkeit um ein vielfaches überlegen. TCP/IP wurde unter folgenden Prämissen entwickelt:

- Unabhängigkeit vom Übertragungsmedium
- Interoperabilität zwischen unterschiedlichen heterogenen Systemen
- Ende-zu-Ende –Kommunikation über unterschiedliche Netzwerke
- Robustheit gegenüber Verbindungstörungen

In dieser Zeit waren als Anwendungen lediglich *ftp* und *telnet*. Erst als sich um 1971 zwei Programmierer nicht nur Daten sondern auch Nachrichten senden wollten, entwickelten sie *Electronic Mail*, die auch heute noch am weitesten verbreitete Anwendung. Im Juli 1975 wurde die Verwaltung des *Arpanet* an die DCA (*Defense Communications Agency*) des US-Verteidigungsministeriums über-

---

[2] Rechner, die dem Benutzer innerhalb eines Netzwerkes Anwendungen zur Verfügung stellen

geben. Zu Beginn der 80er Jahre wurde der militärische Teil ins *Milnet* ausgegliedert, die zivilen Teile, Forschung, Entwicklung und Lehre blieben im *Arpanet*.

Im Jahre 1978 wurde von der US-Regierung beschlossen in öffentlichen und vom Staat geförderten Projekten bei der Dateiübertragung zwischen Computern nur noch TCP/IP einzusetzen. 1983 schließlich wurde im *Arpanet* nur noch TCP/IP als Übertragungsprotokoll verwendet. Seit dieser Zeit etwas besteht auch der allgemeine Begriff *Internet* für das auf TCP/IP basierende *Arpanet* und dessen Netze.

1.3 Die Weiterentwicklung des Internet und die Entwicklung in Europa

Der Siegeszug von TCP/IP als Protokoll im Internet wurde auch dadurch gefördert, daß sich die US-Regierung entschloß bei offenen Systemen in staatlichen Organisationen und vom Staat geförderten Projekten TCP/IP als Protokoll zu verwenden. Als offene Betriebssystemumgebung wurde UNIX gefordert. Speziell entschied man sich hier für Berkeley UNIX (BSD, *Berkeley Software Distribution*) weil sowohl TCP/IP als auch die darauf basierenden Anwendungen *ftp, telnet und E-Mail* Teil des BSD-Betriebssystems waren. Damit war der Grundstein für die Verbreitung von TCP/IP gelegt.

Auch in Europa wurde die Notwendigkeit erkannt den Wissenschaftlern an Universitäten eine schnelle und kostengünstige Kommunikationsinfrastruktur zur Verfügung zu stellen. Die europaweite Koordination wurde 1986 RARE (*Réseaux Associés pour la Recherche Europeénne*) übertragen, die dafür extra gegründet wurde. Das erste Projekt COSINE (*Cooperation for an Interconnection Networking in Europe*) diente der Bereitstellung einer auf ISO/OSI-Normen (ISO: *International Standards Organisation*, siehe 2.1) basierenden Infrastruktur für den akademischen Bereich innerhalb Europas. In Europa sollten im Gegensatz zu den USA vorwiegend Applikationen, die auf den ISO/OSI-Normen basieren, zum Einsatz kommen. Das wichtigste Ergebnis aus COSINE-Projekt war das erste paneuropäische Netzwerk auf der X.25-Basis IXI (*International X.25 Interconnect*), das seit Februar 1993 als EuropaNET in einem Multiprotokoll-Backbone fortgeführt wird. Natürlich konnte man sich gegenüber den Entwicklungen in den USA nicht verweigern, schließlich mußte man bedenken, daß eine Verbindung der Netzwerke sehr sinnvoll sein würde. RIPE (*Réseux IP Européens*) übernahm die Koordinierung des Internetverkehrs in Europa [SBGK94].

1.4 Organisation im Internet

Im Gegensatz zu einem lokalen oder auch globalen Firmennetzwerk gibt es im Internet keine zentralen Administratoren. Aber auch für das Internet ist es wichtig, daß sich Gruppen um den laufenden Betrieb und die Entwicklungen kümmern. So steht durch das rasante Wachstum des Internets in den letzten Jahren unter anderem das Problem an, daß die Adressen für die einzelnen Rechner inzwischen knapp werden. Solche Betriebsgruppen sind zum Beispiel, die in 1992 als Non-Profit-Organisation gegründete Internet Society (ISOC). Ihre Ziele:

„Die technische Entwicklung des Internet als Infrastruktur für die Forschung und Lehre zu forcieren und akademische, wissenschaftliche und technische Gesellschaften in die Weiterentwicklung des Internet einzubinden,

Den Wissenschaftlern, aber auch der Öffentlichkeit, die Technik, den Umgang und die Anwendungen des Internet nahezubringen,

Wissenschaftliche und lehrebezogene Anwendungen der Internettechnologie zum Nutzen von Lehranstalten aller Stufen, der Industrie und der gesamten Öffentlichkeit zur Verfügung zu stellen,

Ein Forum zu sein für die Entwicklung neuer Internet-Anwedungen und die Zusammenarbeit von Organisationen, die sich mit der Entwicklung, dem Betrieb und der Gestaltung neuer Anwendungen beschäftigen, untereinander zu fördern.“([SBGK94]).

Für die weitere technische Entwicklung ist das Internet Architecture Board verantwortlich. Sie soll laut ISOC die weitere Entwicklung der Architektur eines Multiprotokoll-Internet beaufsichtigen.

Eine weitere wichtige Organisation ist das NIC (*Network Information Center*), denn in einem Netzwerk ist es wichtig, daß jeder Rechner über einen eindeutigen Namen verfügt, um ihn im Netz eindeutig zu identifizieren. Innerhalb eines TCP/IP-Netzwerkes wurde hierzu eine 32-Bit Adresse eingeführt (bestehend aus vier 8 Bit Worten, die durch Punkte von einander getrennt werden, z. B. 193.140.4.14). Diese Adressen werden in jedem Land von einer speziellen Organisation vergeben, in Deutschland übernimmt diese Aufgabe das DE-NIC (Deutsches Network Information Center). Da Namen einfacher zu handhaben sind als Zahlen, hat man den *Domain Name Service (DNS)* eingerichtet. Hier wird zu jeder IP-Adresse im Internet der logische eindeutige Namen vergeben [SBGK94].

2 Protokollarchitektur

TCP/IP ist zur Zeit eines der wichtigsten Softwarepakete in den UNIX-Systemen, aber auch in den anderen Betriebsystemen wie zum Beispiel MS Windows 95, MS Windows NT 4.0 und IBM OS/2 Warp werden die TCP/IP-Implementierungen immer leistungsfähiger. Gerade erst hat IBM eine neue Implementierung des TCP/IP-Stacks für IBM OS/2 Warp freigegeben. Die Vorteile von TCP/IP, die dazu führten, daß es zum Standard innerhalb des Internets wurden, sind:

„Offene Protokollspezifikationen, die frei zugänglich und unabhängig von der Hardware und dem Betriebssystem sind. Aufgrund seiner weitreichenden Unterstützung eignet sich TCP/IP in idealer Weise dazu, unterschiedliche Hardware und Software miteinander zu verbinden – auch wenn kein direkter Anschluß an das Internet geplant ist.

Unabhängigkeit von einer bestimmten Netzwerkhardware. So ist es möglich, mittels TCP/IP viele unterschiedliche Netze miteinander zu verbinden. TCP/IP kann über ein Ethernet, einen Token-Ring, eine Wählleitung, ein X.25-Netz und beinahe jedes andere physikalische Übertragungsmedium betrieben werden.

Ein einheitliches Adressierungsschema, das es jedem Rechner in einem TCP/IP-Netz ermöglicht jeden beliebigen anderen Rechner in diesem Netz eindeutig zu identifizieren. Dies gilt sogar für das weltweite Internet.

Standardisierte Protokolle der höheren Schichten, die dem Benutzer einheitliche und weithin verfügbare Dienste zur Verfügung stellen" ([Hunt95;S. 3]).

2.1 Modell zur Datenkommunikation

Zur Erklärung der Realisierungen von Netzwerkprotokollen dient das ISO-OSI-Modell (*International Standards Organisation, Open System Interconnection*). Diese Realisierungen werden in sieben Schichten (*Layer*) beschrieben, wobei jede Schicht eine spezielle Funktionalität hat und auf die darunterliegende Schicht zugreift oder Informationen an die darüberliegende Schicht weiter reicht. Einen direkten Zugriff auf nicht benachbarte Schichten über eine andere Schicht hinweg ist nicht erlaubt. Da diese übereinander liegenden Schichten wie Stapel von Ziegelsteinen wirken, spricht man auch von einem *Protokoll-Stack*. Diese Schichten sind im einzelnen:

- **Anwendungsschicht:** (*Application Layer*) Der Anwender hat die Möglichkeit direkt auf Anwendugsprozesse zu zugreifen. Dies können Prozesse sein, mit denen der Benutzer direkt agiert oder auch Prozesse von denen er direkt nichts merkt. Beispiel: *FTAM (File Transfer Access and Managment)*: Ein OSI-Protokoll für den Dateizugriff. Oder *FTP (File Transfer Protocol)*: Ein Internet Protokoll zur Dateiübertragung.
- **Darstellungsschicht:** (Presentation Layer) Kooperierende Anwendungen müssen zum Zweck des Datenaustauschs sich auf eine gemeinsame Darstellung der Daten einigen. In diese Schicht fallen auch die Aufgaben, wie Datenkompression und Alphabetumwandlungen.
- **Kommunikationssteuerungsschicht**: (*Session Layer*) In dieser wird die Verbindung zwischen den kooperierenden Anwendungen realisiert.
- **Transportschicht**: (*Transport Layer*) Sie stellt sicher, daß der Empfänger die Daten genauso bekommt, wie der Absender sie abgeschickt hat. Sie garantiert die netzwerkunabhängige, gesicherte Übertragung von Daten zwischen zwei Prozessen. Dazu gehören der Aufbau und Unterhalt der Verbindung, Multiplexing, Fehlerbehandlung und das Ordnen der Daten.
- **Vermittlungsschicht**: (*Network Layer*) Dieser Schicht kommt die Aufgabe zu, die Verbindung zu anderen Rechnersystemen aufzubauen. Dies umfaßt die Bereitstellung geeigneter Adressierung, die Vermittlung, den Verbindungsaufbau und –abbau, Rücksetzung, Unterbrechung, Fehlererkennung und den transparenten Datentransport zwischen den Netzwerkendpunkten. Unter dem Aspekt Transparenz fallen Anpassungen der Eigenarten verschiedener Sicherungsschichten und auch Anpassungen an die sich ändernden Netzwerktopologien.
- **Sicherungsschicht**: (*Data Link Layer*) Die Aufgabe der Sicherungsschicht ist die zuverlässige Übertragung von Daten über das zugrundeliegende physikalische Netzwerk. Notwendige Funktionen sind: Segmentieren, Kontrollieren und die Behandlung von Fehlern.
- **Bitübertragungsschicht**: (*physical Layer*) Diese Schicht definiert die Charakteristika der Hardware, die zur Übertragung der Daten benötigt wird. Signalpegel und die Anzahl von Kontakten in

Steckern sind in dieser Schicht spezifiziert. Als Beispiel stellt hier die Norm IEEE 802.3 für lokale Netzwerke dar.

2.2 Protokollarchitektur von TCP/IP

Die Spezifikation des TCP/IP Protokolls ist älter als das ISO-OSI-Modell, dies wurde erst 1983 eingeführt, daher unterscheidet sich das Internet Modell an einigen Stellen. Es gibt im Aufbau allerdings au ch einige Gemeinsamkeiten, so wird die TCP/IP-Netzwerkschicht im ISO-OSI-Modell zu einer Bitübertragungs- und Sicherungsschicht. Zwischen Transport und Anwendungsschicht sind noch die beiden Schichten Kommunikation und Darstellung enthalten. Diese sind bei TCP/IP-Software oft schon in den Protokollen der Anwendungsschicht integriert [HoBr95]. So entsteht ein Modell mit nur noch 4 Schichten:

4	**Anwendungsschicht**
	Enthält Anwendungen und Prozesse, die auf das Netzwerk zugreifen.
3	**Transportschicht**
	Stellt End-zu-End-Datendienste zur Verfügung.
2	**Internet – Schicht**
	Definiert den Aufbau von Datagrammen und routet Daten.
1	**Netzzugangsschicht**
	Enthält Routinen für den Zugriff auf physikalische Netze.

Abb. 1: Protokoll TCP/IP.

Analog zum ISO-OSI-Modell werden die Daten im Stack nach unten weitergereicht, wenn Daten verschickt werden. Jede Schicht fügt ihre eigenen Kontrollinformationen hinzu, um eine korrekte Datenübertragung zu gewährleisten. Diese Informationen werden als *Header* (Kopf) bezeichnet, weil sie den eigentlichen Daten vorangestellt werden. Jede Schicht betrachten die von oben erhaltenen Daten als die zu übertragenden Daten und fügt den eigenen *Header* hinzu. Dieses Hinzufügen bezeichnet man als Kapselung. Beim Empfang von Daten wird diese Kapselung rückgängig gemacht [Hunt95].

2.2.1 Netzzugangsschicht des TCP/IP-Stacks

Die unterste Schicht in dieser Protokollhierarchie ist die Netzzugangsschicht (*Network Access Layer*). Sie deckt die Funktionalitäten Vermittlung, Sicherung und Bitübertragung aus dem ISO-OSI-Modell ab. Diese Schicht definiert, wie ein IP-Datagramm[3] über das Netzwerk transportiert wird. Anders als die höheren Schichten, muß die Netzzugangsschicht den Aufbau des zugrundeliegenden Netzes kennen. Dazu gehört die Struktur seiner Pakete, seine Adressierung usw. So ist es auch nötig mit der Einführung von neuer Hardware neue Netzzugangsprotokolle zu entwickeln, um diese in TCP/IP-Netzwerken zu nutzen. Zu den Funktionen dieser Schicht gehört es ebenfalls, daß die eindeutige IP-

Adresse in eine Adresse umgewandelt wird, die das physikalische Netzwerk versteht. Zum Beispiel gibt es das *Adress Resolution Protokol* (ARP), das IP-Adressen auf Ethernet-Adressen abbildet oder ein Protokoll, das beschreibt, wie IP-Datagramme für den Transport über Ethernet-Netze gekapselt werden .

2.2.2 Internet-Schicht des TCP/IP-Stacks

Das wichtigste Protokoll dieser Schicht ist das Internet Protokoll (IP; RFC 791). Es bildet die Grundlage für TCP/IP-Netzwerke, in dem es den Transport von Datagrammen definiert. Alle Protokolle oberhalb und unterhalb dieser Schicht benutzen IP für den Transport von Daten. Alle TCP/IP-Daten durchlaufen das IP, unabhängig von ihrem Ziel. Dies gilt gleichfalls für ausgehende als auch für eingehende Daten. Die Funktionen des IP sind die wichtigsten Funktionen, die das Internet erst ermöglichen:

- Definition des Datagramms, der kleinsten Einheit für die Übertragung im Internet
- Definition der Adressierung im Internet
- Datentransport zwischen der Netzzugangsschicht und den Protokollen der Transportschicht
- Routing von Datagrammen zu fremden Rechnern
- Fragmentierung und Defragmentierung von Datagrammen

IP ist ein verbindungsloses Protokoll, das bedeutet, daß vor der Datenübertragung keine Kontrollinformationen zwischen zwei Rechnern ausgetauscht werden müssen, um eine Übertragung zu ermöglichen. Dagegen ist ein verbindungsorientiertes Protokoll ein Protokoll, daß vor der Übertragung von Daten mit dem entfernten Rechner zuerst Informationen austauscht. Der Sender fragt beim Empfänger an, ob er bereit ist, Daten zu empfangen. Wenn der Empfänger mitteilt, daß er bereit ist, beginnt die Datenübertragung. Man spricht davon, daß eine Verbindung aufgebaut worden ist. Sollte innerhalb eines TCP/IP-Netzwerkes ein verbindungsorientierter Dienst benötigt werden, so muß dies durch die Protokolle anderer Schichten geleistet werden.

Die Erkennung und Korrektur von Fehlern wird ebenfalls Protokollen anderer Schichten überlassen. IP sendet die Daten korrekt, es überprüft allerdings nicht, ob die Daten genauso empfangen wurden, wie sie gesendet wurden. Das Internet Protokoll wird daher auch als unzuverlässiges Protokoll bezeichnet [Hunt95].

Datagramm

Die TCP/IP-Protokolle wurden entworfen, um im *Arpanet* Daten zu transportieren. Hierbei handelte es sich um ein *Paketvermittlungsnetz.* Dieses Paket, das die Daten enthält, wird durch weitere Informationen ergänzt. Die wichtigsten Aufgaben von IP sind es die Adressinformationen für die einzelnen Pakete zu ergänzen, so daß sie möglichst schnell den richtigen Empfänger erreichen. Ähnlich wie bei

[3] Die Internet-Schicht betrachtet alle Daten als Blöcke, die Datagramm genannt werden

einem Brief, bei dem der Umschlag die Adressinformation enthält, wird den Paketen eine Adresse mitgegeben. Diese Adressinformation wird genutzt um die Pakete von einem physikalischen Netzwerk in das andere zu senden.

Ein *Datagramm* ist das vom Internet-Protokoll definierte Paketformat.

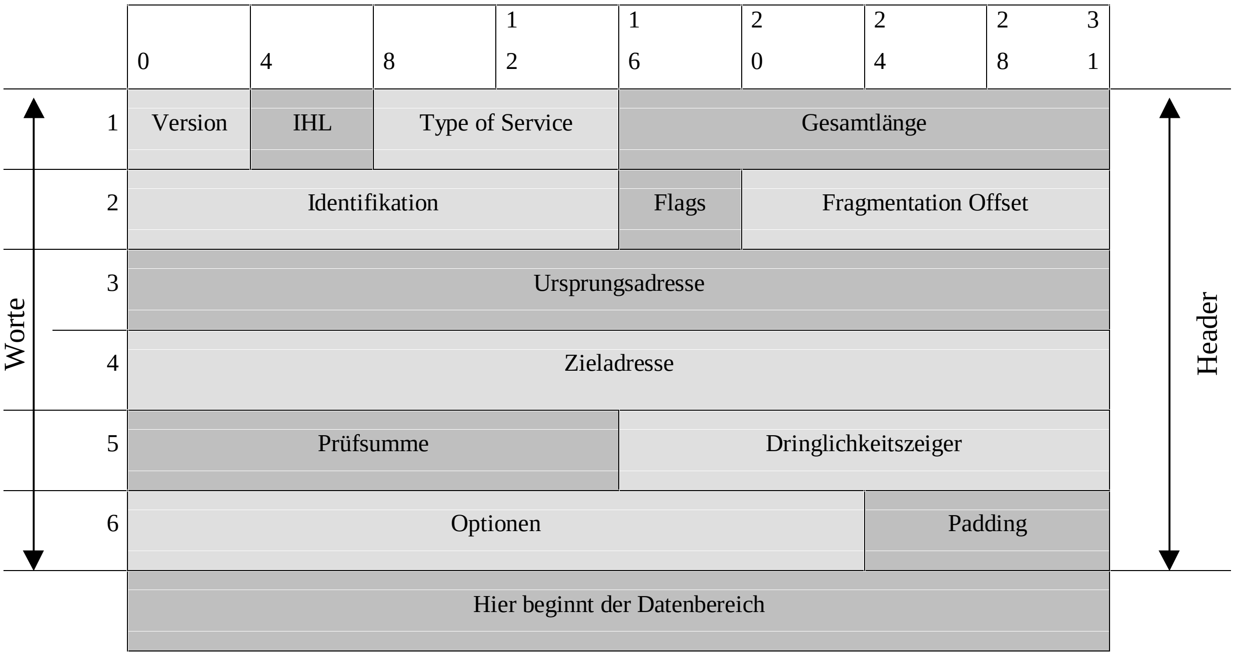

Abb. 2: IP-Datagramm

Die ersten fünf oder sechs 32-Bit-Wörter enthalten Kontrollinformationen und werden als *Header* bezeichnet. Die Größe des Header ist nicht fest vorgegeben. Normalerweise ist er fünf Worte groß, das sechste Header-Wort ist optional. Die Größe des Headers läßt sich über das Feld *Internet Header Length (IHL)* variieren. Im Header sind alle Informationen enthalten um ein Paket zustellen zu können. Die Datagramme werden nun mittels des Internet-Protokoll transportiert, indem das fünfte Wort des Headers gelesen wird. Hier steht die *Destination Address (Zieladresse)* als Standard-IP-Adresse mit einer Länge von 32 Bits (für Aufbau und Format der IP-Adressen siehe 3.1). Liegt die IP-Adresse im lokalen Netzwerk, dann wird das Paket direkt zugestellt. Sonst wird das Paket an ein *Gateway* übergeben. Gateways sind Rechner, die Pakete zwischen verschiedenartigen physikalischen Netzwerken umsetzen. Die Wahl eines geeigneten Gateways nennt man *Routing*. IP bewerkstelligt dieses Routing für jedes einzelne Datenpaket.

Routing von Datagrammen

Innerhalb des Internets werden die Gateways in der Regel als *IP-Router* bezeichnet, dies hängt damit zusammen, daß sie das IP-Protokoll benutzen, um Pakete durch die Netzwerke zu routen.

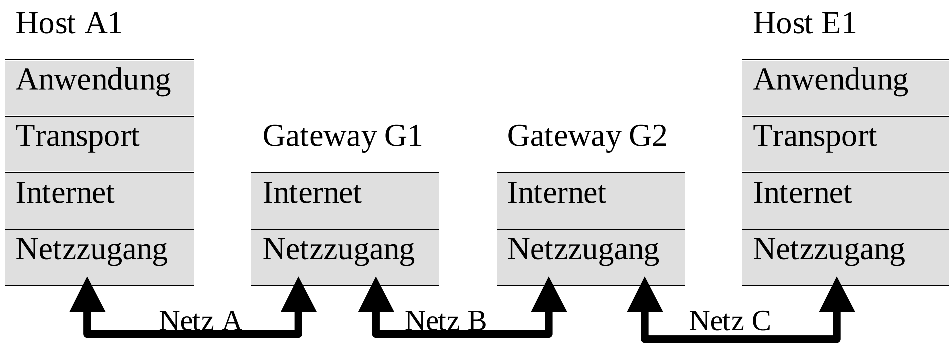

Abb. 3: Routing mit Hilfe von Gateways.

Diese Abbildung zeigt wie Pakete mit Hilfe von Gateways zwischen den Netzwerken transportiert werden. Dabei werden in den Hosts Pakete durch alle vier Protokollebenen gereicht, während in den Gateways das Routing in der Internet-Schicht durch das Internet-Protokoll abgewickelt wird. Ein anderer Aspekt bei diesem Routing ist, daß auch unterschiedliche physikalische Netzwerke (z. B. Token-Ring oder Ethernet) verbunden werden.

Fragmentierung von Datagrammen

Sollen Pakete durch verschiedene physikalische Netzwerke geroutet werden, so kann es notwendig werden, daß das IP in einem Gateway das Paket in kleinere Einheiten aufteilen (*fragmentieren*) muß. Dies hängt damit zusammen, daß Netzwerke unterschiedliche Paketgrößen verwenden: X.25, eine Schnittstelle in der Vermittlungsschicht für den Zugang zu öffentlichen Paketvermittlungen, wie Datex-P arbeitet beispielsweise mit 128 Bytes, Ethernet dagegen mit 1526 Bytes [HoBr95]. Im Format des Datagramms sind im zweiten Wort die Information für die Defrafragmentierung und Wiederherstellung gespeichert. Das Feld *Identification* zeigt an, zu welchem Datagramm ein Fragment gehört und das Feld *Fragmentation Offset* gibt die Position eines Fragmentes im Datagramm an. Das Feld *Flags* enthält ein *More-Fragments-Bit*, an dem IP ablesen kann, ob alle Fragmente eines Datagramms empfangen wurden [Hunt95]. Alle Datagramme haben das gleiche Format, es ist unabhängig davon, ob es fragmentiert ist oder nicht.

Übergabe von Datagrammen an die Transportschicht

Empfängt das Internet-Protokoll ein Datagramm, das für den eigenen Rechner bestimmt ist, so muß es dieses Datagramm an das entsprechende Protokoll in der Transportschicht weiterleiten. Mit Hilfe der *Protokollnummer* im dritten Wort des Headers wird dies gewährleistet. Alle Protokolle der Transportschicht werden anhand einer eindeutigen Protokollnummer identifiziert [ChZw96].

Internet Control Message Protocol (*ICMP*)

Das zweite wichtige Protokoll in der Internet-Schicht ist das in RFC 792 definierte *Internet Control Message Protocol*. Mit Hilfe der Datagramm-Dienste des Internet-Protokolls versendet es seine Meldungen. Diese Meldungen dienen der Kontrolle und Information und erfüllen im einzelnen folgende Funktionen:

- **Flußkontrolle:** Kann ein Gateway oder ein Zielrechner eingehende Datagramme nicht schnell genug bearbeiten, schickt er an den Absender die ICMP-Source-Quench-Meldung (Überlauf). Damit wird der Absender aufgefordert das Senden von Datagrammen vorübergehend einzustellen.
- **Erkennen unerreichbarer Ziele**: Falls ein Ziel nicht erreichbar ist, wird an den Absender des Datagramms die Meldung Destination-Unreachable (Ziel nicht erreichbar) gesendet. Handelt es sich dabei um einen unerreichbaren Rechner oder ein Netzwerk, wird diese Meldung von einem Gateway generiert. Sollte es sich um einen nicht erreichbaren Port, dann sendet der Zielrechner diese Meldung (Ports werden in Kapitel 3 genauer behandelt).
- **Änderungen im Routing**: Gateways können durch eine Route-Redirect-Meldung einem Rechner mitteilen, daß er seine Datagramme über ein anderes Gateway senden soll, weil dieses Gateway eine günstigere Position hat. Dies kann nur erfolgen, wenn die beiden Gateways im gleichem physikalischem Netzwerk sind.
- **Statusabfrage bei fremden Rechnern**: Um festzustellen, ob ein anderer Rechner über IP erreichbar ist, kann eine ICMP-Echo-Meldung an diesen Rechner abgesandt werden. Empfängt ein Rechner dieses Meldung, dann sendet er gleiche Paket zurück. [Hunt95],[ChZw96].

2.2.3 Transportschicht des TCP/IP-Stacks

Betrachten wir nun die über der Internet-Schicht liegende Transportschicht. Sie wird auch als *Rechner-zu-Rechner-Transportschicht* bezeichnet. In dieser Schicht stellen das *Transmission Control Protocol (TCP)* und das *User Datagram Protocol (UDP)* die beiden wichtigsten Protokolle dar. TCP stellt einen zuverlässigen Datenübertragungsdienst mit Fehlererkennung und –korrektur von einem Ende der Verbindung bis zum anderen bereit. Dagegen bietet UDP eine verbindungslose Übertragung mit geringem Verwaltungsaufwand. Beide Protokolle stellen eine Schnittstelle zwischen der Anwendungsschicht und der Internet-Schicht zur Verfügung. Der Anwendungsprogrammierer hat die Entscheidung zu treffen, welcher Dienst für ihn der geeignete ist.

User Datagram Protocol (UDP)

UDP bietet einen einfachen verbindungslosen Datagrammdienst, der gegenüber IP lediglich zusätzlich Portnummnern und eine Prüfsumme bietet. Dies geschieht durch Nutzung des ersten Wortes im Header, im ersten 16-Bit Wort ist der Ursprungsort codiert, im zweiten der Zielport. Das zweite 32 Bit Wort des Headers enthält die Länge und die Prüfsumme. Sinnvoll ist UDP vor allem dann, wenn nur eine geringe Datenmenge zu übertragen ist und das erneute Senden der Daten mit weniger Aufwand verbunden ist, als der Aufwand für das Herstellen einer Verbindung und das Sicherstellen einer korrekten Übertragung. Gleichsam kann es sein, daß andere Anwendungen oberhalb dieser Schicht ebenfalls effiziente Methoden für eine sichere Datenübertragung zur Verfügung stellen und somit käme es zu einer ineffizienten Datenübertragung [Hunt95].

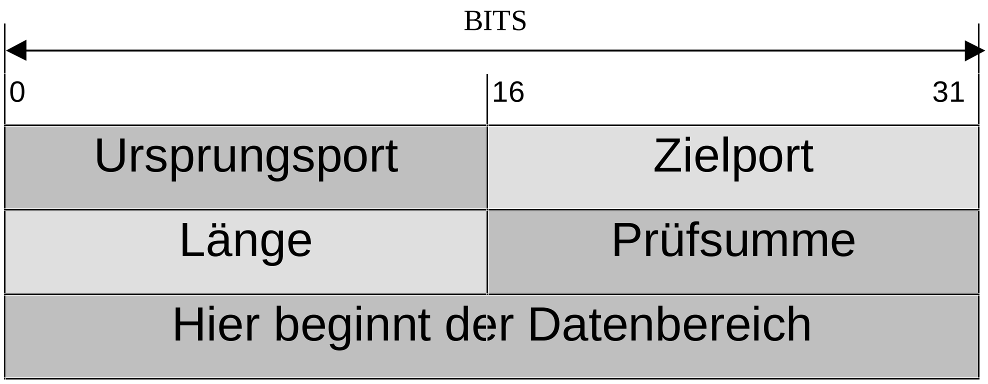

Abb. 4: Aufbau eines UDP Datagramms

Das Transmission Control Protocol (TCP)

Anwendungen, die auf eine zuverlässige Datenübertragung angewiesen sind, nutzen das Transmission Control Protocol (TCP). Es stellt sicher, daß die Daten korrekt und in der richtigen Reihenfolge über das Netz transportiert werden. Bei TCP handelt es sich um ein *zuverlässiges, verbindungsorientiertes Byte-Stream-Protocol*. Betrachten wir nun die drei Begriffe: zuverlässig, verbindungsorientiert und Byte-Stream (Datenstrom).

Die Zuverlässigkeit wird durch einen Mechanismus erreicht, der als *Positive Acknowledgement with Re-Transmission (PAR*; positive Bestätigung mit Neu-Übertragung) bezeichnet wird. Das heißt, daß ein Rechner die Daten solange noch einmal sendet, bis er vom Empfänger mitgeteilt bekommt, daß er die Daten richtig empfangen hat. Die Dateneinheit, die TCP-Module bei der Kommunikation untereinander verwenden, heißen *Segment*. In diesem Segment ist eine Prüfsumme enthalten, diese wird auf der Seite des Empfängers ausgewertet um zu testen, ob die Daten richtig eingetroffen sind.Iist dies der Fall sendet der Empfänger an den Absender eine *positive Bestätigung* zurück. Trifft ein beschädigtes Datensegment ein, dann wird es ignoriert. Der Absender reagiert darauf indem er nach einer gewissen Wartezeit alle Segmente noch einmal senden wird, für die er keine positive Bestätigung erhalten hat.

Man spricht davon, daß TCP verbindungsorientiert arbeitet, weil es eine logische Verbindung zwischen Sender und Empfänger herstellt. Dazu werden zu Beginn einer Verbindung erst einige *Kontrollinformationen (handshake)* zwischen Sender und Empfänger ausgetauscht und startet damit den Dialog zwischen den Kommunikationspartnern. Ein Kontrollsegment wird durch Setzen des entsprechenden Bits im *Flags*-Feld des *Segment-Headers* geschaffen.

Bei dem von TCP benutzen Handshake handelt es sich um einen *3-Wege-Handshake*. Diese Bezeichnung kommt daher, da zum Verbindungsaufbau 3 Segmente ausgetauscht werden. Diese 3 Segmente haben die folgende Bedeutung: Rechner A will mit Rechner B eine Verbindung aufbauen, dazu sendet Rechner A ein Segment in dem das Bit Synchronize sequence numbers (SYN; Seqenznummern synchronisieren) gesetzt ist. Empfängt Rechner B dieses Segment, so weiß er das Rechner A eine Verbindung aufbauen will und mit welcher Sequenznummer Rechner A sein erstes Segment, das es über-

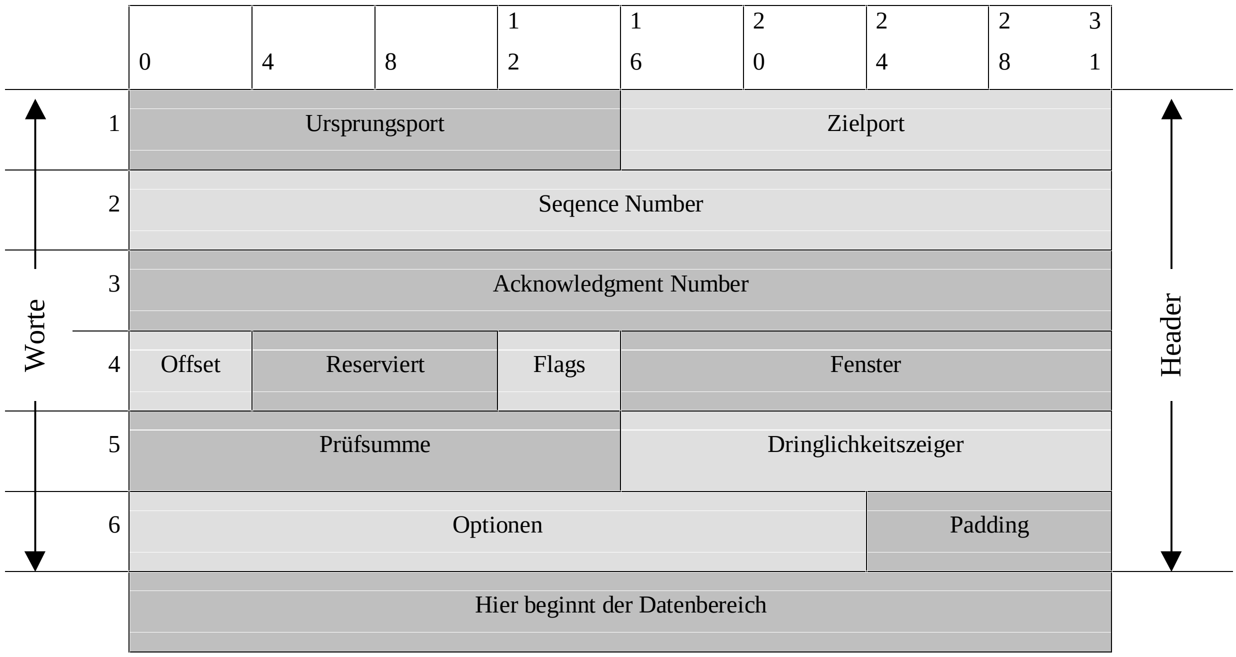

Abb. 5: Aufbau des TCP-Segments

tragen will, versenden wird. Die Seqenznummern sorgen dafür, daß die Daten in der richtigen Reihenfolge bleiben. Rechner B sendet nun ein Segment, in die Bits *Acknowledgment* (ACK) und *Synchronize sequence numbers* (SYN) gesetzt sind. So quittiert Rechner B den Erhalt des Segments von Rechner A und teilt gleichzeitig die Sequenznummer mit die Rechner B für sein ersten Versenden eines Segmentes nutzen wird. Rechner A quittiert daraufhin dieses Segment und beginnt mit der eigentlichen Übertragung der Daten. Nach dem Übertragen dieser Kontrollsegmente hat Rechner A die Gewißheit, daß das TCP von Rechner B bereit ist, Daten zu empfangen. Nach der Übertragung aller Daten leitet Rechner A einen Verbindungsabbau ein, wieder durch einen 3-Wege-Handshake. Dazu sendet Rechner A ein Segment mit gesetztem Bit *No more data from Sender* (FIN; keine weiteren Daten vom Absender).

Die übertragenen Daten werden von TCP als ununterbrochener Datenstrom betrachtet und nicht als Reihe unabhängiger Pakete. Mit Hilfe von *Sequence Number* und *Acknowledgment* werden die Daten in der richtigen Reihenfolge gehalten. Bereits in der Phase des Verbindungsaufbaus mit Segment 1 hat Rechner A mitgeteilt, mit welcher Sequenznummer er die Datenübertragung starten will, diese Sequenznummer wird als *Intitial Sequence Number* (ISN; Anfangssequenznummer, diese Sequenznummer ist vom Protokoll nicht vorgegeben, es wird aber in der Regel mit 0 begonnen).

Im weiteren Verlauf der Datenübertragung werden die Datenbytes fortlaufend durchnumeriert, so hat das erste Datenbyte die Nummer 1 und wird mit Sequenznummer 1 im Header des Datensegmentes versandt. Die Sequenznummer kennzeichnet somit die Position des Datenbytes im Datenstrom.

Mit Hilfe des Acknokledgment-Segment (ACK) werden zwei Funktionen realisiert. Zum einen die *positive Bestätigung (positive acknowledgment)* und die *Flußkontrolle (flow control)*. Die Bestätigung teilt dem Absender mit, wieviele Daten bereits empfangen wurden, und wieviele mehr noch empfangen werden können. Die Bestätigungsnummer (acknowledgment number) ist die Sequenznummer des letzten Bytes, das am anderen Ende empfangen wurde. Es ist vom Protokoll nicht vorgesehen, daß jedes Paket einzeln zu bestätigen ist. Die Bestätigungsnummer zeigt lediglich an, daß bis zu dieser Nummer die Bytes richtig empfangen wurden.

Dem Empfänger ist es möglich über das Feld Window (Fenster) den Datenfluß zu steuern. Dieses Feld enthält den Wert, wieviele weitere Bytes der Empfänger noch empfangen kann. Setzt der Empfänger diesen Wert auf 0, so wird der Sender solange keine weiteren Daten mehr senden, bis er einen anderen Wert im Feld Windows empfängt. Solange dieser Wert im Feld Windows für den Sender kleiner ist, als die für ihn noch nicht bestätigten Daten, wird er Daten senden. Kommt es dazu, daß der Empfänger nun die maximale Anzahl an Bytes empfangen hat, aber keine Bestätigung sendet, so wird der Sender nach einer angemessenen Wartezeit, mit der Neuübertragung nach dem letzten bestätigten Byte beginnen. So wird sichergestellt, daß die Daten ihr Ziel sicher erreichen.

Die Transportschicht ist auch dafür verantwortlich, daß ein korrekter Datenaustausch mit der Anwendungsschicht erfolgt. Die Anwendungen werden durch eine 16 Bits lange Nummer namens *Port Number* identifiziert. Der *Source Port* und *Destination Port* (Ursprungs- und Zielport) sind im ersten Wort des Segment-Headers kodiert [Hunt95].

2.2.4 Anwendungsschicht des TCP/IP-Stacks

Die Anwedungsschicht (*application layer*) befindet sich auf der obersten Ebene der TCP/IP-Protokollarchitektur. Hier sind alle Prozesse angesiedelt, die zur Datenübertragung die Protokolle der Transportschicht nutzen. In der Anwendungsschicht gebt es bereits zahlreiche Protokolle, und es kommen laufend neue hinzu. Viele dieser Protokolle stellen dem Benutzer direkt Dienste zur Verfügung. Die bekanntesten sind sicherlich:

- Telnet, das Network Terminal Protocol, es ermöglicht dem Benutzer das Anmelden auf fernen Rechnern im Netz,
- FTP, das File Transfer Protocol ermöglicht die interaktive Dateiübertragung,
- SMTP, das Simple Mail Transfer Protocol ist für die Zustellung elektronischer Post verantwortlich.

Dies sind sicherlich die bekanntesten TCP/IP-Anwendungen mit denen ein Benutzer konfrontiert wird. Es gibt aber noch andere wichtige Anwendungen:

- *Domain Name Service (DNS)* auch *Name Service* genannt, hat die Aufgabe numerische IP-Adressen logischen Namen zu zuordnen,
- *Routing Information Protocol (RIP)* wird von Geräten im Netz genutzt um Routing-Information auszutauschen,

- *Network File System (NFS)* ist ein gemeinsames Dateien-System zur Nutzung der Dateien unabhängig von der verwendeten Hardwareplattform.

Unterscheiden lassen sich hier zwei Gruppen von Diensten, einmal Dienste, die der Benutzer direkt benutzt, wie zum Beispiel die ersten drei Dienste telnet, ftp und SMTP. Dagegen arbeiten die drei anderen Dienste als Prozesse weitgehendst unbemerkt vom Anwender [ChZw96].

3 Übertragen von Daten

Bis jetzt haben wir nur den grundlegenden Aufbau der TCP/IP-Protokollhierachie betrachtet. Nun werden wir betrachten, wie die Daten zwischen den einzelnen Protokollschichten und den einzelnen Rechnern transportiert werden.

Um Daten zwischen zwei Rechnern im Internet zu übertragen, muß sichergestellt sein, daß die Daten über das Netzwerk zum richtigen Empfänger gelangen, und daß sie dort den richtigen Anwendungsprozeß erreichen. Um dies zu gewährleisten verfügt TCP/IP über drei Mechanismen:

Adressierung: Durch eindeutige IP-Adressen ist es möglich, jeden Rechner im Internet eindeutig zu identifizieren und somit die Daten korrekt zu zustellen.

Routing: Durch Gateways werden Datagramme an das richtige physikalische Netzwerk weitergeleitet.

Multiplexing: Protokoll- und Port-Nummern stellen sicher, daß die Daten im Zielrechner an den richtigen Anwendungsprozeß übergeben werden.

3.1 Aufbau und Format der IP-Adressen

Das Internet-Protokoll überträgt Daten zwischen Rechnern in Form von Datagrammen. Jedes Datagramm wird an die Adresse im Internet weitergeleitet, die im Feld Zieladresse (das fünfte Wort) des Datagramm Headers angegeben ist. Die IP-Adresse ist eine 32 Bits lange Adresse. Sie enthält genug Informationen, um einen Rechner im Internet eindeutig zu identifizieren. Diese Adressen bestehen aus zwei Teilen: einer *Netzadresse* und der *Adresse des Rechners* innerhalb des Netzes. Das Format dieser beiden Teile ist allerdings nicht in allen IP-Adressen dasselbe. Die drei wichtigsten Adressklassen sind die Klassen A, B und C. Wenn IP-Software die Adresse liest, kann es an Hand der ersten Bits feststellen, zu welcher Klasse eine Adresse gehört. IP-Adressen werden im allgemeinen als vier durch Punkte getrennte Dezimalzahlen geschrieben. Folgende Regeln gelten zum Aufbau der IP-Adressen:

- Das erste Bit einer IP-Adresse ist 0, so handelt es sich um ein Adresse der Klasse A. Das erste Bit einer Adresse der Klasse A kodiert die Klassenzugehörigkeit. In den weiteren 7 Bits wird das Netzwerk identifiziert. Und die restlichen 24 Bits kodieren den Rechner innerhalb des Netzwerkes. Es gibt weniger als 128 Netze der Klasse A, aber jedes von ihnen kann Millionen von Rechner enthalten. (Erstes Byte kleiner als 128 ist das erste Byte die Nummer des Netzwerkes, die anderen drei Bytes sind die Rechnernummer.)

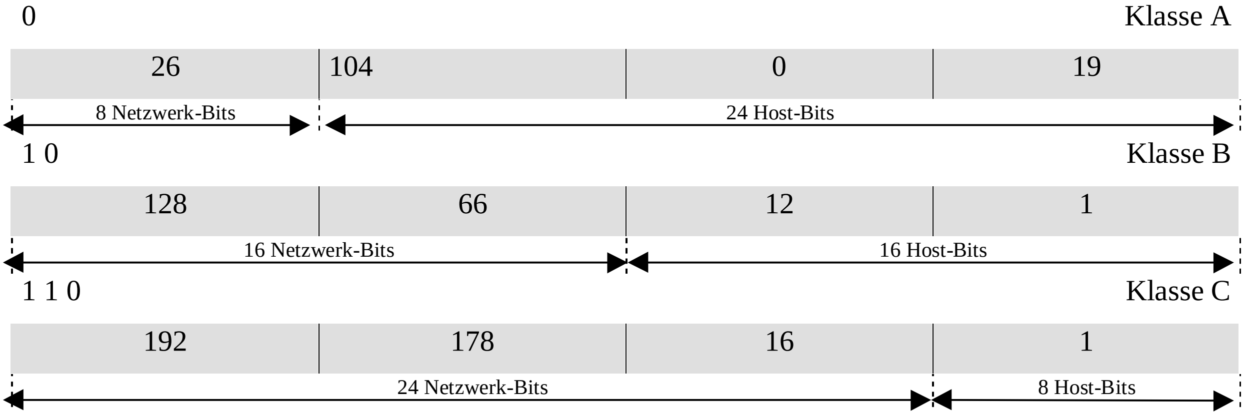

Abb. 6: Netzklassen

- Wenn die ersten beiden Bits einer IP-Adresse 1 0 sind, handelt es sich um eine Adresse in einem Netzwerk der Klasse B. Die ersten beiden Bits einer Klasse-B-Adresse bestimmen die Adressklasse, die nächste 14 Bits identifizieren das Netz, und die letzten 16 Bits den Rechner. Es gibt Tausende von Netzen der Klasse B und jedes von ihnen kann Tausende von Rechnern enthalten.(Erstes Byte zwischen 128 und 191, die ersten beiden Bytes bezeichnen das Netzwerk die letzten beiden Bytes den Rechner.)
- Wenn die ersten drei Bits einer Adresse 1 1 0 sind, handelt es sich um eine Adresse in einem Netzwerk der Klasse C. Die ersten drei Bits dieser Adresse dienen der Bestimmung der Klasse. Die nächsten 21 Bits bestimmen das Netzwerk und die letzten 8 Bit bestimmen den Rechner. Es gibt Millionen von Netzen der Klasse C, aber jedes dieser Netze kann nur 254 Rechner enthalten. (Wert zwischen 192 und 223, die ersten drei Bytes bilden die Netzwerknummer, das letzte Byte ist die Rechnernummer.)
- Sind die ersten drei Bits einer Adresse 1 1 1 , handelt es ich um eine speziell reservierte Adresse. Diese Adressen werden manchmal als Adressen der Klasse D bezeichnet. Sie spezifizieren aber eigentlich kein Netzwerk. Die in diesem Bereich vergebenen Adressen sind sogenannte Multicast-Adressen, mit denen lassen sich Gruppen von Computern adressieren, die ein gemeinsames Protokoll benutzen, aber nicht im gleichen physikalischen Netzwerk sind (Werte größer als 223).

Es gibt noch zwei andere reservierte Adressen, das ist in der Klasse A die Adresse 0 sie bezeichnet die *Default-Route (Standard*- oder *voreingestellte Route)* und das Netzwerk 127 ist die *Loopback-Adresse* (etwa: auf sich selbst verweisende Adresse). Die Default-Route dient zur Vereinfachung des Routings. Die Loopback-Adresse vereinfacht Netzwerkanwendungen, weil der lokale Rechner genauso adressiert werden kann, wie ein fremder Rechner.

Es gibt weitere Rechnernummern, die einen besonderen Zweck erfüllen. Diese Adressen haben als erstes Byte die 0 oder die 255. Eine IP-Adresse in der alle Rechnerbits auf 0 stehen, identifizieren das Netzwerk selber. Eine IP Adresse 28.0.0.0 bezeichnet demnach das Netzwerk 28. Solche Adressen werden in Routing-Tabellen verwendet. Stehen in einer IP-Adresse alle Bits auf 1 bezeichnet man diese Adresse als *Broadcast-Adresse* (Rundfunk-Adresse). Diese Adresse wird benutzt um ein Datagramm an jeden Rechner im Netzwerk zu senden.

3.2 Subnet (Teilnetze)

Es besteht die Möglichkeit die vorgegebene Struktur des Netzes zu ändern. So ist es möglich aus einem großen Netzwerk mehrere kleinere logische Netzwerke zu machen. Dadurch bekommt man Netzwerke, die leichter zu administrieren sind. Oder auch einfach nur logisch besser zu einander passen. Aus technischen Gründen kann es auch sinnvoll sein, Teilnetze zu installieren. So können IP-Router physikalisch verschiedene Netzwerke miteinander verbinden. Dazu muß aber jedes Netzwerk eine eindeutige Netzwerknummer haben. Durch das Subnetting teilt man eine einzige physikalische Netzadresse in viele eindeutige Subnet-Adressen auf. So bekommt jedes physikalische Netz seine eigene Adresse. Bei Subnetting wird die Grenze zwischen Netzadresse und Rechneradresse verschoben [Hunt95].

Ein Subnet wird definiert, in dem man die IP-Adresse mit einer sogenannten Subnetmaske verknüpft. Diese Subnetmaske ist auch wieder, wie die IP-Adresse ein 32-Bit-Wort. Ist ein Bit der Subnetmaske gesetzt, so wird das entsprechende Bit als Netzadresse interpretiert. Ist das Bit nicht gesetzt, wird es als Rechneradresse gewertet. Das Subnet ist immer nur lokal bekannt, für den Rest der Welt, ist nur die IP-Adresse bekannt.

Ein Beispiel: Betrachten wir ein Netzwerk der Klasse B mit der Subnetmaske 255.255.0.0. Die Adresse wird nun durch die Subnetmaske 255.255.255.0 erweitert. Die Adresse ist dann so zu verstehen, daß die ersten beiden Bytes das Netzwerk bezeichnen, das dritte Byte das Subnet und das vierte Byte die Rechneradresse.

3.3 Routing im Internet

Zu Beginn des Internets gab es eine Hierarchie von Gateways. Dies resultiert daraus, daß zu Beginn des Internets das ARPANET als Backbone für das Netz genutzt wurde. Dieses System nannte man Core (Kern) und die Gateways dazwischen wurden als *core gateways* bezeichnet. In dieser hierarchischen Struktur werden die Routing-Informationen über alle Netzwerke an die core gateways weitergegeben. Das ganze geschieht über das *Gateway to Gateway Protocol (GGP)*. Außerhalb dieses Internet Kerns gibt es Gruppen von eigenständigen Netzen, die man als *autonomous systems* bezeichnet. Hier gibt es ebenfalls ein Protokoll mit dessen Hilfe die Routing Informationen ausgetauscht werden, das *Exterior Gateway Protocol (EGP)*. Das Defense Data Network (DDN) im Internet benutzt weiterhin das Kernmodell, um Routing-Informationen zu verteilen. Das hierarchische Modell hat den Nachteil, daß jede einzelne Route vom Kern bearbeitet werden muß. Die führt zu einem erheblichen Aufwand. So daß zur Zeit an einem neuen Modell für das Routing gearbeitet wird. Dieses Modell faßt

mehrere autonome Systeme zu gleichberechtigten *routing domains* zusammen. Zwischen diesen Domains werden Routing Informationen mit Hilfe des *Border Gateway Protocol (BGP)* oder EGP ausgetauscht. Im Gegensatz zu GGP verlassen sich die Routing-Domains nicht auf ein zentrales System zur Berechnung der besten Route, sondern jede Routing-Domain übernimmt diese Aufgabe selber. Dadurch ist das System einfach zu erweitern, wenn neue Netzwerke oder Rechner hinzukommen.
Nicht nur Gateways routen Daten durch die Netzwerke. Jeder Rechner muß ebenfalls Routing-entscheidungen treffen. Sollte der Zielrechner im lokalen Netzwerk liegen, werden die Daten zu diesem gesandt. Sonst werden die Daten an ein lokales Gateway geleitet. Das Routing orientiert sich an den Netzwerken, daher trifft IP seine Routing-Entscheidungen anhand des Netzwerkteils der Adresse. Aus der Adressklasse ergibt sich, welcher Teil der Adresse gelesen wird, um das Zielnetzwerk zu bestimmen. Nachdem das Zielnetzwerk bestimmt wurde, sucht IP dieses Netz in der *Routing-Tabelle*. Mit Hilfe der Angaben der Routing-Tabelle werden die Daten dann weitergeleitet. Diese Routing-Tabellen können vom Systemadministrator oder auch von Routing-Protokollen erstellt worden sein. IP trifft seine Entscheidungen für das Routing durch Lesen dieser Tabellen.

3.4 Multiplexing

Wenn die Daten durch das Netz gesandt worden sind, muß auch sicher gestellt werden, daß sie die richtige Anwendung oder den richtigen Prozeß im Zielrechner erreichen. Natürlich müssen auch die Daten zwischen den einzelnen Protokollschichten richtig weiter gereicht werden. Dieser Mechanismus muß Daten aus vielen Anwendungen zu wenigen Transportprotokollen zusammenfassen. Das Verdichte von Daten aus mehreren Quellen zu einem Datenstrom nennt man *multiplexen*. Beim Empfänger müssen die aus dem Netzwerk eintreffenden Daten durch IP *demultiplext* werden. Um dies zu bewerkstelligen kennzeichnet IP die Transportprotokolle mit *Protokollnummern*. Damit die Anwendungen eindeutig zu identifizieren sind, kennzeichnen die Transportprotokolle die Anwendungen über Port-Nummern. Für häufig benötigte Protokolle und Anwendungen *(well-known services)* gibt es bestimmte Protokoll- und Port-Nummern. Im dritten Wort des Datagramm-Headers ist codiert an welches Protokoll der Transportschicht die Daten zu übergeben sind. Mit Hilfe der 16 Bits langen Portnummer identifiziert die Transportschicht den Anwendungsprozeß für den die Daten bestimmt sind. Diese Portnummern stehen im ersten Wort eines jeden TCP- und UDP-Paketes. Genauer gesagt, stehen hier die *source port number (Ausgangs-Portnummer)* und auch die *destination port number (Ziel-Portnummer)*. So kann aus Protokollnummer und Portnummer der Prozeß eindeutig bestimmt werden, der die Daten zu erhalten hat. Es gibt für bestimmte Anwendungen reservierte Ports (*well-known-ports)*. Dies vereinfacht den Aufbau einer Verbindung, weil sowohl Empfänger als auch Absender vorab wissen, daß die Daten für einen bestimmten Prozeß an einen bestimmten Port geleitet werden müssen, so ist beispielsweise für TELNET der Port 23 reserviert. Es gibt aber nicht nur statisch vergebene Port-Nummern, sondern auch dynamische Portnummern *(dynamically allocated* *ports)*. Somit ist es möglich zwei Anwendern zum Beispiel den Zugriff per TELNET auf einen Rechner zu ermöglichen.

4 Name Service

Im letzten Kapitel haben wir gesehen, wie Routen und Adressen im Netzwerk von TCP/IP verwaltet und gehandhabt werden. Jedes Netzwerk-Interface in einem TCP/IP-Netz wird durch eine eindeutige, 32 Bits lange IP-Adresse identifiziert. So kann nun jedem Interface mit einer IP-Adresse ein *Hostname (Rechnername)* zu geordnet werden. Diese Rechnernamen sind einfacher zu merken und erleichtern somit dem menschlichen Benutzer die Arbeit. Um einen Rechner anzusprechen, kann sowohl die IP-Adresse als auch der Rechnername verwendet werden. Beim Aufbau einer Verbindung wird der Name in die eindeutige IP-Adresse gewandelt. Diese Namenswandelung ist nicht nur eine lokale Angelegenheit, sondern sie muß für das gesamte Internet eindeutig sein. So gibt es im Internet zwei gebräuchliche Methoden um Namen in IP-Adressen zu konvertieren. Bei der älteren der beiden Methoden wird der Name in einer Tabelle namens *host table (Rechnertabelle)* gesucht, zu dem Namen ist in der Tabelle die 32 Bit lange IP-Adresse vermerktDie Rechnertabelle ist eine einfache Textdatei, in der IP-Adressen Namen zugeordnet werden. Natürlich müssen die Rechnernamen im Internet eindeutig sein, daher müssen diese Namen zentral verwaltet werden. Diese Aufgabe übernimmt das *Network Information Center (NIC)* [4]. Da das Internet gerade in der letzten Zeit sehr rasant gewachsen ist, ist diese Rechnertabelle so groß geworden, daß mit ihr nicht mehr effizient gearbeitet werden kann. Der jetzige Standard ist der *Domain Name Service (DNS)*. . Er benutzt eine verteilte Datenbank um Namen zu Adressen zu zuordnen und behebt somit die beiden großen Schwachpunkte der Rechnertabelle.

- Der DNS beruht auf einem weltweit verteilten Datenbanksystem. So läßt er sich leicht erweitern.
- Der DNS stellt sicher, daß Informationen über neue Rechner bei Bedarf an den Rest des Internets verteilt werden.

Das Weiterleiten von Informationen bei Bedarf erfolgt nicht automatisch, sondern nur dann, wenn es nötig ist. Erhält ein DNS-Server eine Anfrage, die er nicht bearbeiten kann, sendet er diese Anfrage an einen *autoriativen Server* weiter. Dieser Server hat die Aufgabe für die fragliche Domain genauere Informationen zu haben. Die Antwort, die der DNS-Server erhält, merkt er sich und kann so bei einer gleichen Anfrage diese direkt beantworten. Auf diese Art und Weise erhält der DNS-Server seine neuen und aktualisierten Informationen.

Der DNS ist ein verteiltes, hierarchisches System zur Konvertierung von Rechnernamen in IP-Adressen. Die Informationen sind über Tausende von Name-Servern verteilt. Diese Name-Server sind ähnlich wie das UNIX-Dateisystem organisiert. An der Spitze steht die *Root-Domain* mit einer kleinen Gruppe von Name-Servern, diese werden als *Root-Server* bezeichnet. Direkt unterhalb der Root-Do-

[4] Es gibt mehr als ein NIC. Im allgemeinen bezeichnet man mit NIC das InternetNIC

main befinden sich die *top level domains*. Diese unterscheiden sich in zwei Typen, zum einen in die geographischen und zum anderen in die organisatorischen. Die geographischen Domains werden durch einen zweistelligen Code, zum Beispiel **de** für Deutschland, gekennzeichnet. Organisatorische Top-Level-Domains sind zum Beispiel **com** für kommerzielle Organisationen.

Wie erhält man nun aber eine Domain? Diese werden durch *Network Information Center (NIC)* vergeben. Dazu stellt man entweder bei einem NIC oder bei einem Internet-Provider einen Antrag. In diesem Antrag sind mindestens der Name und die Adresse von zwei Servern anzugeben werden. Diese übernehmen dann für die neue Domain den Name Service. Hat man die Domain erhalten, kann man beliebig viele Subdomains einrichten. Die Administratoren sind für die Verwaltung des Namensraum verantwortlich. Die Domain-Namen spiegeln die Domain-Hierarchi wider. Domain-Namen schreibt man von der untersten Ebene (dem Rechnernamen) zur obersten Ebene (der Top-Level-Domain). Die einzelnen Teile des Namens werden durch Punkte voneinander getrennt.

5 Abgrenzung zu anderen wichtigen Protokollen

TCP/IP ist nicht das einzige Protokoll, das es ermöglicht Daten zwischen Computern zu übertragen. Es gibt noch andere so zum Beispiel von den Firmen Novell SPX/IPX oder von IBM NetBEUI. Dies soll nicht bedeuten, daß es keine anderen Protokolle gibt.

5.1 TCP/IP im Vergleich zu SPX/IPX

Es ist äußerst schwierig, Netzwerkprotokolle miteinander zu vergleichen. TCP/IP Experten würdigen SPX/IPX meistens nicht richtig, da sie es für ein proprietäteres Protokoll halten. Allerdings wird SPX/IPX auf über 60 Prozent aller Desktop-Rechner eingesetzt. IPX ist gegenüber TCP/IP besser auf LAN (*Local Area Network)* Umgebung abgestimmt, dem gegenüber hat TCP/IP Vorteile im Bereich von WAN *(Wide Area Network)* Umgebungen. In den folgenden Protokollschichten können TCP/IP- und SPX/IPX Protokolle auf Gemeinsamkeiten und Unterschiede untersucht werden.

- Bitübertragungs und Sicherungsschicht
- Netzwerkschicht
- Transportschicht
- Anwendungsschicht

Auf der Bitübertragungs und Sicherungsschicht sind TCP/IP und SPX/IPX identisch. Beide Protokolle arbeiten nun mit den meisten üblichen Netzwerktopologien, wie zum Beispiel Ethernet, Token-Ring zusammen.

Innerhalb der Netzwerkschicht enthält TCP/IP zusätzlich zum Basis IP-Protokoll, Dienste zur Auflösung der Adressen. SPX/IPX hat dies nicht, weil es seine Adressen aus den physikalischen Adressen des Netzes ableitet.

Auf der Netzwerkschicht besitzen IPX und IP die gleichen Eigenschaften. Beide sind verbindungslose Datagrammdienste, die auf die Protokolle der Bitübertragungsschicht, wie Ethernet, aufsetzen. Beide Protokolle haben ähnliche Routingprotokolle. Sie unterscheiden sich aber in den Adreßplänen.
Auf der Transportschicht haben SPX und TCP ähnliche Eigenschaften. Beide bieten auf der Basis von IPX oder IP einen verbindungsorientierten und zuverlässigen Übertragungsdienst an. IPX ist wie UDP ein verbingungsloses Protokoll.
Auf der Anwendungsschicht gibt es in einer Netware Umgebung keine Entsprechung zwischen beiden Protokollen. Lediglich die Nachrichtenprotokolle, wie das *Simple Mail Transfer Protocol (SMTP)* und das *Message Handling System (MHS)* ähneln einander [SFLa95].

5.2 TCP/IP im Vergleich zu NetBEUI

Als eines der ersten Protokolle stand *NetBEUI* zur Verwendung in Personal-Computer-Netzwerken zur Verfügung. IBM stellte *NetBEUI* als Protokoll vor, das in Verbindung mit Software-Programmen für die *NetBIOS*-Oberfläche (*Network Basic Input/Output System*) verwendet werden konnte. NetBIOS definiert neben einer Software-Schnittstelle auch noch eine Namenskonvention für Computer im Netzwerk.
NetBEUI wurde als kleines effizientes Protokoll entworfen, das eine einfache und effiziente Vernetzung von Computern in einem Local Area Netzwerk ermöglichen sollte. Es ist nicht vorgesehen, daß durch dieses Protokoll Daten in andere Subnetze gesendet werden können, sprich in NetBEUI sind keine Routingfunktionalitäten vorgesehen. Mittlerweile wird NetBEUI ausschließlich in kleineren Netzwerken eingesetzt, die aus verschiedenen Computern mit unterschiedlichen Betriebssystemen arbeiten. So unterstützen zum Beispiel alle netzwerkfähigen Betriebssysteme von Microsoft, IBM PCLAN und LAN Server von IBM dieses Protokoll.
In der Anwendungsschicht greift ein Programm über die NetBIOS-Software-Schnittstelle auf das NetBEUI-Protokoll zu.
Das NetBEUI-Protokoll stellt ähnlich wie das TCP/IP-Protokoll sowohl eine unzuverlässigen verbindungslose als auch eine zuverlässige verbindungsorientierte Datenübertragung zur Verfügung.
NetBEUI übernimmt weiterhin die Aufgaben der Verbindungseinrichtung, Wartung und Beendigung, Rahmenabfolge und –bestätigung, Rahmenflußkontrolle und die verbindungslose Datenübertragung.
Daten werden in sogenannten Rahmen und Angabe der *SSAP (Source Service Access Point)*[5] und *DSAP (Destination Service Access Point)*[6] versendet.
In frühen Implementierungen von NetBEUI wurde eine 1-Byte Zahl verwendet um NetBIOS Sitzungen zu identifizieren, daher waren die Sitzungen auf 254 beschränkt. Das entspricht den Protokoll-, Portnummern und Sockets bei TCP/IP [Mic196].

[5] sendender Client, der den Rahmen generiert hat
[6] empfangender Client des Rahmens

In einfachen kleinen Netzwerken zum Beispiel in einem Büro mit weniger als 20 Rechnern ist NetBEUI ein geeignetes Netzwerkprotokoll. Aber spätestens wenn alle Netzwerkteilnehmer einen Zugang zum Internet wünschen, wird man nur schwer auf TCP/IP verzichten können. Ein ähnliches Problem stellt sich auch bei der Nutzung von Novells IPX/SPX-Protokollstandards. Novell wird in der nächsten Version seines Novell Netware Betriebssystems ebenfalls TCP/IP als Protokoll verwenden.

6 Literatur- und Quellenverzeichnis

[SBGK94] M. Scheller; K. P. Boden; A. Geenen; J. Kampermann. *Internet: Werkzeuge und Dienste*. Springer Verlag, Berlin (Deutschland), 1994.

[Hunt95] C. Hunt. *TCP/IP Netzwerk Administration.* O'Reilly/International Thomson Verlag, Bonn (Deutschland), 1995

[ChZw96] D. Chapmann; E. Zwicky. „*Einrichten von Internet Firewalls*". O'Reilly/International Thomson Verlag, Bonn (Deutschland), 1996

[HoBr95] F. Hosenfeld; K. Brauer. „Kommunikation ohne Grenzen". *C´t Magazin für Computertechnik.* Heise Verlag, Hannover, Nr. 12, 1995

[GaSp95] S. Garfinkel, G. Spafford. „*PRACTICAL UNIX SECURITY*". O'Reilly & Associates, Inc., Sebastopol (USA), 1995.

[SFLa95] P. Singh; R: Faiweather; D. Laderman. "*Mit Netware ins Internet*".Markt & Technik Buch und Software Verlage, Haar bei München (Deutschland),1995

[Micr96] „*Grundlagen des Netzwerkbetriebs*". Microsoft Press, Unterschleißheim (Deutschland), 1996

[Mic196] „*Die technische Referenz Microsoft Windows NT Server Version 4.0 Netzwerk*". Microsoft Press, Unterschleißheim (Deutschland), 1996

Kryptographie: Grundlagen und Algorithmen

Oliver Brühl

1 Grundlagen

1.1 Einleitung

In diesem Beitrag werden kryptographische Algorithmen vorgestellt. Dazu gehören Verschlüsselungsverfahren, die grob in symmetrische und asymmetrische Verfahren eingeteilt werden können. Ebenfalls besprochen werden Message-Digest-Algorithmen, einwegige Hash-Funktionen, die einen eindeutigen Fingerabdruck eines Dokumentes liefern.

Mit diesen Verfahren können komplexe Protokolle entwickelt werden, die die Bedürfnisse nach sicherer Kommunikation erfüllen können. Neben der offensichtlichen Funktion, die Vertraulichkeit von Daten zu bewahren, können kryptographische Algorithmen und darauf aufbauende Protokolle auch sicherstellen, daß Authentizität, Integrität und Verbindlichkeit der Daten gewährleistet ist.

Bei der Authentizitätsprüfung möchte man sicherstellen, daß der Gegenüber derjenige ist, für den er sich ausgibt. Sollen zum Beispiel auf einem Server bestimmte Dateien nur einzelnen Gruppen zugänglich sein, so wird der Server den Benutzer auffordern, sich zu authentifizieren, etwa mit einer Benutzerkennung und einem Paßwort. Auch in umgekehrter Richtung sollte Authentifikation möglich sein. Der Benutzer soll sicher sein können, daß die Daten vom richtigen Server stammen.

Die Forderung nach Integrität bedeutet, daß Veränderungen der Daten während der Übertragung festgestellt werden können.

Verbindlichkeit ist z. B. im elektronischen Handel wichtig. Der Kommunikationspartner soll nicht behaupten können, daß die Kommunikation nicht stattgefunden hat. So muß ein Verkäufer einem Käufer nachweisen können, daß er eine bestimmte Bestellung getätigt hat. Andererseits hat auch der Käufer ein Interesse daran, daß Lieferzusagen und Angebote verbindlich sind.

1.2 Substitution und Transposition

In der Kryptographie werden die Methoden, um Klartext zu verschleiern, in zwei Kategorien eingeteilt: Substitution und Transposition.

Bei der Substitution wird jedes Zeichen des Klartextes durch ein anderes ersetzt. Die älteste bekannte Chiffre ist die Cäsar-Chiffre, die, wie man schon vermuten kann, Julius Cäsar zugeschrieben wird. Hier wird jeder Buchstabe des durch denjenigen ersetzt, der sich an der drittnächsten Stelle (modulo 26) im Alphabet befindet. So wird z.B. a zu d, b zu e und y zu b. Allgemeiner kann man die Buchstaben um n modulo 26 in eine Richtung verschieben. Hier wäre n dann der Schlüssel.

Bei der Transposition werden nicht die die Symbole des Alphabets durch andere ersetzt, sondern deren Reihenfolge im Klartext wird vertauscht. Ein einfacher Algorithmus ist die Spaltentransposition. Hier wird der Klartext auf ein Papier mit einer fixen Anzahl von Spalten geschrieben. Der Chiffretext besteht nun aus den vertikalen Spalten, die hintereinander aufgeschrieben werden.
Aus dem Klartext

DERURLAUBMUSSVERSCHOBENWERDEN

wird mit der Spaltenanzahl 6

DASHEEUVORRBEBDUMREERUSNNLSCW,

wie das folgende Bild verdeutlicht:

D	**E**	**R**	**U**	**R**	**L**
A	**U**	**B**	**M**	**U**	**S**
S	**V**	**E**	**R**	**S**	**C**
H	**O**	**B**	**E**	**N**	**W**
E	**R**	**D**	**E**	**N**	

Abb. 1: Beispiel zur Spaltentransposition.

Die heutigen, auf Computern verwendeten kryptographischen Algorithmen basieren immer noch auf dieses beiden Grundprinzipien. Nur hatte das Alphabet damals 26 Elemente, während die Algorithmen heute auf Bitebene arbeiten. Substitutionen werden dabei durch sogenannte S-Boxen, Transpositionen durch P-Boxen realisiert. In S-Boxen werden dabei n Eingabebits durch m Ausgabebits ersetzt, während in einer P-Box die Reihenfolge der Bits vertauscht wird.

1.3 Kryptoanalyse

Wenn man einen verschlüsselten Text hat und versucht, aus ihm den Klartext oder den verwendeten Schlüssel zu gewinnen, nennt man das Kryptoanalyse. Im Allgemeinen wird davon ausgegangen, daß der Verschlüsselungsalgorithmus frei zugänglich ist. Hierbei gibt es drei Szenarien: Im ersten Fall hat man nur Chiffretext. Im zweiten Fall sind sowohl der Klartext als auch der Chiffretext vorhanden. Dies nennt sich dann bekannter Klartext. Der dritte Fall heißt gewählter Klartext. Hier hat der Kryptoanalytiker die Möglichkeit, Klartexte seiner Wahl verschlüsseln zu lassen.
Ein sicheres Kryptosystem sollte auch der Attacke durch gewählten Klartext standhalten.

2 Symmetrische Verschlüsselung

2.1 Grundlagen

Bei der symmetrische Verschlüsselung wird vom Sender und vom Empfänger der gleiche geheime Schlüssel für die Ver- und Entschlüsselung benutzt (siehe Abb. 2). Dieser muß daher vorher über einen sicheren Kanal ausgetauscht werden.

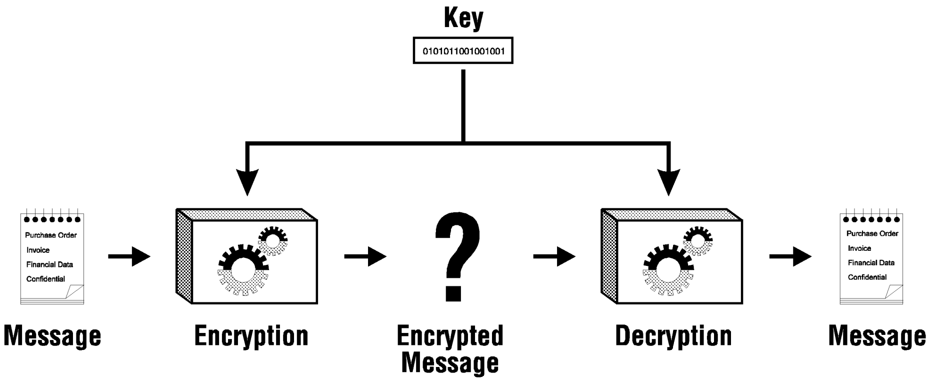

Abb. 2: Symmetrische Verschlüsselung (aus [Iann97]).

Man unterscheidet zwischen Block- und Streamchiffren. Erstere verschlüsseln jeweils einen Klartextblock (typischerweise 64 Bit) in einen Chiffreblock gleicher Länge, während Streamchiffren einen kontinuierlichen Bit- oder Bytestrom verschlüsseln.

2.2 Blockchiffren

2.2.1 Grundlagen

Ein Großteil der verwendeten Blockchiffren sind iterierte Blockchiffren. Bei diesen wird eine Funktion mehrere Runden hintereinander auf den zu verschlüsselnden Block angewandt. Diese Funktion wird durch einen Teilschlüssel parametrisiert, der, für jede Runde auf eine andere Weise, aus dem geheimen Schlüssel abgeleitet wird. Die Anzahl der Runden ist ein Kompromiß zwischen der Sicherheit des Algorithmus und seiner Schnelligkeit, da im allgemeinen die Kryptoanalyse bei steigender Rundenanzahl schwieriger wird.

Feistel-Chiffren [Feis73] sind spezielle iterierte Blockchiffren. Bei diesen Algorithmen wird der Eingabeblock in zwei Hälften aufgeteilt. Auf eine Hälfte wird die Rundenfunktion angewandt und diese dann per exklusiv-oder mit der anderen Hälfte verknüpft. Danach werden die beiden Hälften vertauscht. Einfach ist bei diesen Chiffren die Entschlüsselung, da der Algorithmus genauso durchlaufen werden kann, nur werden die Teilschlüssel in umgekehrter Reihenfolge angewandt. Der bekannteste Feistel-Chiffren ist DES.

2.2.2 DES

Die bekannteste Blockchiffrier ist DES (Data Encryption Standard). Sie ist eine Weiterentwicklung von Lucifer, einem Algorithmus, der Anfang der siebziger Jahre von IBM entwickelt wurde [1482]. Die Entwicklung von DES wurde durch einen Aufruf des *National Bureau of Standards* (NBS), welches heute *National Institute of Standards and Technology* (NIST) heißt, initiiert. Hier wurde dazu aufgerufen, Algorithmen für einen Kryptographiestandard vorzuschlagen. IBM entwickelte DES, und dieser wurde mit Hilfe der *National Security Agency* (NSA) auf seine Sicherheit hin untersucht. DES wurde 1977 als offizielle Norm übernommen [NIST93a].
DES ist ein monoalphabetischer Ersetzungscode, das heißt jeder 64-Bit-Eingabeblock wird (abhängig vom Schlüssel) genau einem 64-Bit-Chiffreblock zugeordnet.

2.2.2.1 Verschlüsselung mit DES

Der Algorithmus besteht grob aus 18 Schritten (siehe Abb. 3). Der erste Schritt ist eine schlüsselunabhängige Permutation. Dann wird der 64-Bit-Block in zwei 32-Bit-Hälften aufgespalten. Diese durchlaufen sechzehnmal die gleiche Funktion. In diesen sechzehn Runden findet die eigentliche Verschlüsselung statt. Zum Schluß wird die Initialpermutation rückgängig gemacht.

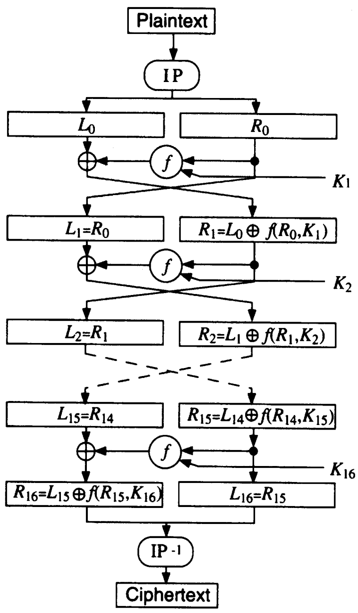

Abb. 3: Die Verschlüsselung mit DES (aus: [Schn96]).

Eine Runde hat als Eingabe jeweils zwei 32-Bit-Zahlen und produziert wiederum zwei 32-Bit-Zahlen. Dabei sind die linken Ausgabebits einfach eine Kopie der rechten Eingabehälfte. Die andere Hälfte ist eine Kombination aus den 64 Eingabebits und dem Schlüssel. Diese Funktion besteht aus vier Stufen (siehe Abb. 4). In der ersten Stufe wird der rechte 32-Bit-Block zu einem 48-Bit-Block expandiert. Dieser wird in der zweiten Stufe per exklusiv-oder mit einem 48-Bit-Teilschlüssel verknüpft.

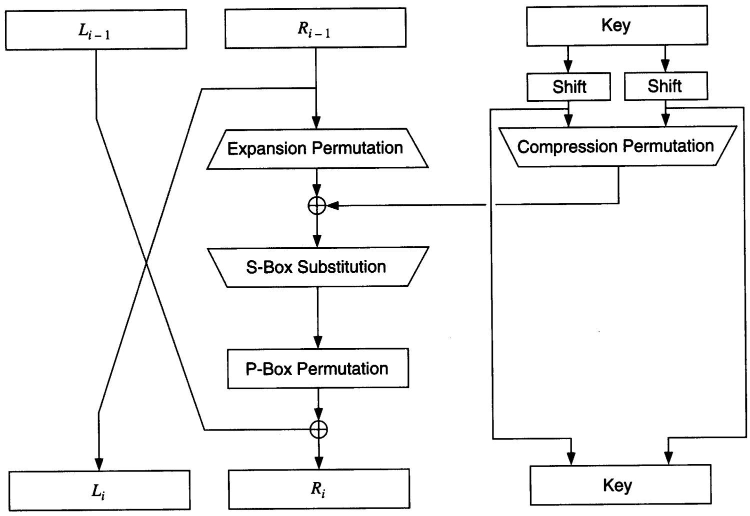

Abb. 4: Eine Runde in DES (aus: [Schn96]).

Letzterer wird aus dem 56-Bit-Schlüssel gewonnen und ändert sich jede Runde. Danach folgt eine S-Box-Substitution, die aus dem 48-Bit Ergebnis der letzten Stufe wieder eine 32-Bit-Zahl macht. Diese Zahl durchläuft noch eine P-Box, dessen Ergebnis per exklusiv-oder mit der linken 32-Bit-Zahl verknüpft wird und als rechter 32-Bit-Block die Runde verläßt.

2.2.2.2 Entschlüsselung mit DES

DES wurde als symmetrischer Algorithmus entworfen. Das heißt, daß die gleiche Funktion sowohl zur Ver- als auch zur Entschlüsselung benutzt werden kann. Nur die Teilschlüssel müssen in der umgekehrten Reihenfolge benutzt werden.

2.2.2.3 DES knacken

Die Schlüssellänge von 56-Bit ist nach heutigen Erkenntnissen zu kurz und kann, wenn auch mit hohem Aufwand, durch Absuchen des gesamten Suchraums geknackt werden. Die Kosten für eine Maschine, die eine mit DES chiffrierte Nachricht (*Brute-Force-Suche*) innerhalb von durchschnittlich 3,5 Stunden entschlüsselt, wurden 1993 von Michael Wiener mit 1 Million US-Dollar angegeben [Wien94], eine auch von größeren Firmen aufzubringende Summe.

2.2.3 Andere BlockChiffren

IDEA (International Data Encryption Algorithm) wurde von den Schweizern Lai und Massey entwickelt und 1990 veröffentlicht [LaMa90]. Die Schlüssellänge ist mit 128 Bit deutlich höher als die von DES und wird aus heutiger Sicht als ausreichend betrachtet. Anders als DES, dessen Operationen in Software nur sehr schlecht implementiert werden können, bedient sich IDEA einfacher Operationen wie exklusiv-oder, Addition modulo 2^{16} und Multiplikation modulo $2^{16}+1$ auf 16-Bit-Teilblöcken, so daß dieser Algorithmus sogar auf 16-Bit Prozessoren läuft. So ist IDEA etwa doppelt so schnell wie DES.

Bekannt wurde IDEA durch seinen Einsatz im dem E-Mail-Verschlüsselungspaket *Pretty Good Privacy* (PGP).

Andere Blockchiffren sind z. B. RC5 [Riv95] und SAFER [Mas93]

2.2.4 Block Chaining

Blockchiffren arbeiten mit einzelnen Blöcken fester Größe. Um diese Blöcke aneinanderzuketten gibt es mehrere Verfahren, sogenannte kryptographische Modi.

2.2.4.1 Electronic Codebook Mode (ECB)

Die einfachste Modus ist der *Electronic Codebook Mode*. Eine Datei wird in 64 Bit lange Blöcke aufgeteilt, jeder Klartexblock wird unabhängig von den anderen Blöcken chiffriert und an den vorherigen Block angehängt. Wenn nötig wird der letzte Block auf 64 Bit aufgefüllt. Der Name dieses Modus kommt daher, daß es theoretisch möglich ist, ein Codebuch mit allen Klartext-Chiffretext-Paaren zu erstellen, da ein Block bei gleichem Schlüssel immer den gleichen Chiffretext ergibt. Der Vorteil diese Verfahrens ist, daß auf Dateien an beliebigen Stellen zugegriffen werden kann. Dies ist bei den später vorgestellten Modi nicht möglich, da die chiffrierten Blöcke voneinander abhängen.

Dieser Modus kann aber, auch wenn weder Schlüssel noch Algorithmus, sondern nur die Blockgröße bekannt ist, per *Block Replay* mißbraucht werden. Dazu ein Beispiel aus [Tane97, S. 609]:

In Abbildung 5 sieht man den Klartext einer in 16 DES-Blöcken verschlüsselten Datei, die Jahresboni verschiedener Mitarbeiter enthält.

Name		Position	Bonus
Adams, L	eslie	Clerk	$ 10
Black, R	obin	Boss	$500,000
CollINs,	Kim	Manager	$100,000
Davis, B	obbie	Janitor	$ 5

Abb. 5: Beispiel zum *Block Replay*.

Leslie weiß, daß er dieses Jahr keinen hohen Bonus erwarten darf. Er hat aber Zugriff auf die verschlüsselte Datei und kennt auch ihren Aufbau, er weiß also, welche Informationen in welchem Block gespeichert sind. Auch ohne Schlüssel kann er diese Situation ausnutzen, indem er den Block 11, den mit dem Bonus von Kim, in den Block 3 kopiert. Wenn niemand genau in die Abrechnungen schaut, kann er sich auf ein schönes Weihnachtsfest freuen.

2.2.4.2 Cipher Block Chaining Mode (CBC)

Eine Möglichkeit, den oben beschriebenen Angriff zu verhindern, ist das *cipher block chaining* (CBC). Beim CBC wird ein Block vor der Verschlüsselung per exklusiv-oder mit dem vorherigen Chiffreblock verkettet. Der erste Block wird dabei mit einem zufällig initialisierten Initialisierungs-

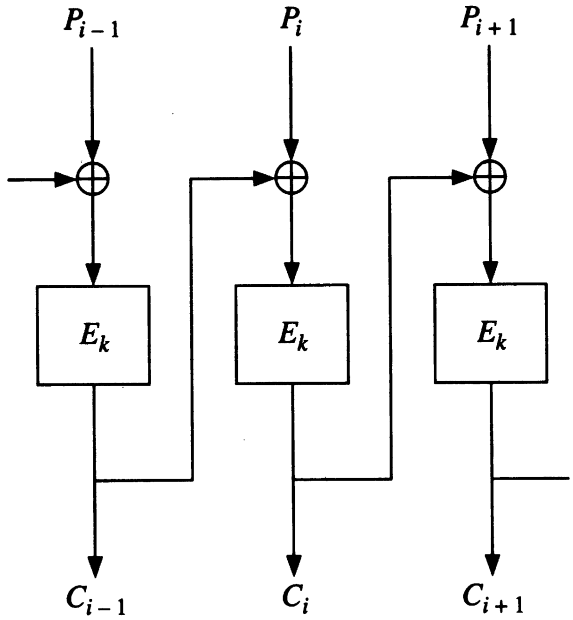

Abb. 6: Verschlüsselung von CBC (aus: [Schn96]).

block, der mit übertragen wird, verkettet (siehe Abb. 6). Als Ergebnis dieses Verfahrens wird der gleiche Klartextblock nicht auf den gleichen Chiffreblock abgebildet.

Um aus den chiffrierten Blöcken wieder Klartext zu gewinnen, werden sie dechiffiert und dann per exklusiv-oder mit dem vorherigen chiffrierten Block verknüpft. Dabei benutzt man bei dem ersten Block wiederum den Initialisierungsblock.

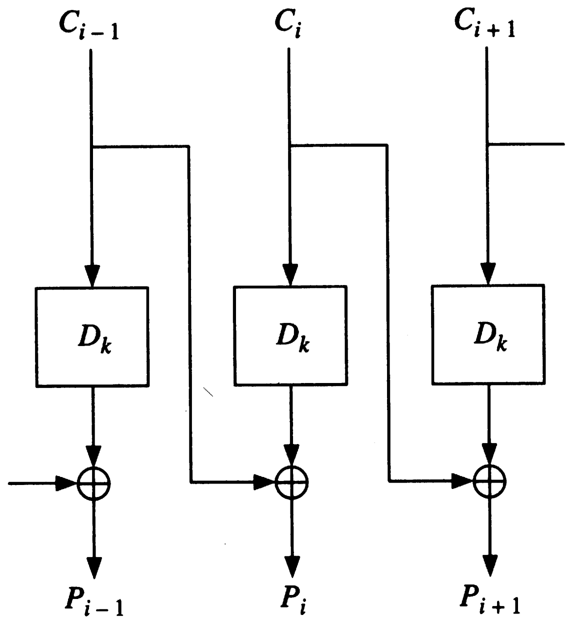

Abb. 7: Entschlüsselung von CBC (aus: [Schn96]).

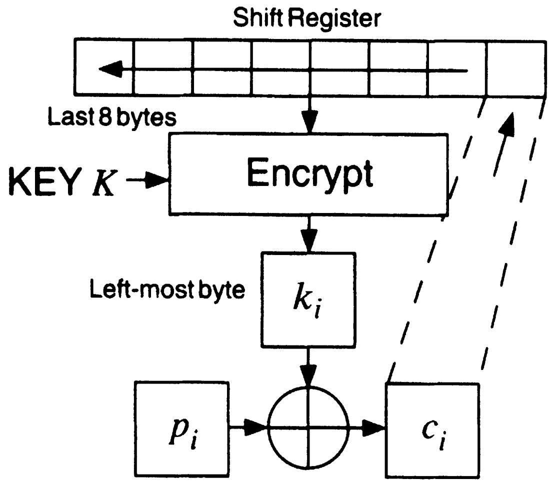

Abb. 8: Verschlüsselung von CFB (aus: [Schn96]).

2.2.4.3 Cipher-Feedback Mode (CFB)

Es gibt auch die Möglichkeit, einen Blockchiffre als *self-synchronizing stream cipher* (s. u.) zu verwenden. Damit umgeht man ein Problem beim *cipher block chaining*: Mit der Dechiffrierung kann erst begonnen werden, wenn ein ganzer 64-Bit-Block angekommen ist. In interaktiven Umgebungen, wie zum Beispiel bei Terminals, kann dieses Verhalten unerwünscht sein. Hier kann man mit dem *cipher-feedback mode* Abhilfe schaffen. Hier wird der Chiffreblock in einem 64-Bit-Schieberegister gehalten. Beim 8-Bit-CFB wird ein ankommendes Byte mit dem linken Byte des Chiffretexts verknüpft und das Ergebnis wird sowohl in das Schieberegister geschoben als auch über die Übertragungsleitung gesendet. Die linken 8 Bit aus dem Schieberegister werden jeweils weggeworfen. Allgemeiner ist der n-Bit-CFB Mode. Hier ist n eine Bitanzahl zwischen eins und der Blockgröße.
Die Entschlüsselung erfolgt analog. Dabei wird sowohl bei der Ver- als auch der Entschlüsselung der Blockchiffre im Verschlüsselungsmodus betrieben.

2.2.4.4 Output-Feedback Mode (OFB)

Der *output-feedback mode* ist vom Aufbau her ähnlich dem *cipher-feedback mode*, nur wird hier das Byte (beziehungsweise die n Bits beim n-Bit OFB), welches rechts in das Schieberegister eingefügt wird, *vor* dem exklusiv-oder entnommen. So wird der Blockchiffre *als synchronous stream cipher* (s. u.) benutzt.

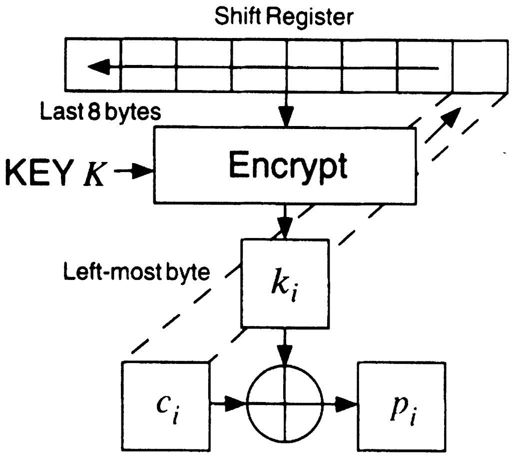

Abb. 9: Entschlüsselung von CFB (aus: [Schn96]).

2.3 Streamchiffren

2.3.1 Grundlagen

Im Gegensatz zu Blockchiffren, die jeweils einen ganzen Klartextblock chiffrieren, verschlüsseln Streamchiffren einen Klartextstrom aus einzelnen Bits oder Bytes per exklusiv-oder mit einem *key-stream*. Key-Stream-Generatoren sind im Prinzip Pseudozufallszahlengeneratoren, deren Bitfolge durch den Schlüssel bestimmt wird. Beim Empfänger wird der *key-stream* mit Hilfe des Schlüssels reproduziert, und durch ein simples exklusiv-oder wird der Klartext wiedergewonnen.

Bei einem *self-synchronizing stream cipher*, zu denen die meisten Streamchiffren gehören, hängt der interne Zustand von den letzten n Bits des bearbeiteten Chiffrestroms ab. Wenn also beim Entschlüsseln diese n Bits empfangen wurden, hat sich der Generator des Empfängers synchronisiert und der Rest der Nachricht kann entschlüsselt werden.

Bei einem Bitfehler in verschlüsselten Strom werden, da dieses Bit ja auch den internen Zustand verändert, n Bits im entschlüsselten Klartext verstümmelt. Danach wird der Strom wieder richtig entschlüsselt.

Wenn der *key-stream* unabhängig von den verschlüsselten Daten ist, hat man einen *synchronous stream cipher*.

2.3.2 Algorithmen

Algorithmen sind hier zum Beispiel RC4 von Ron Rivest [Rive92a] und SEAL (Software-optimized Encryption Algorithm) von Rogaway und Coppersmith [RC93].

3 Asymmetrische Verschlüsselung

3.1 Grundlagen

Algorithmen für die asymmetrische Verschlüsselung (auch *public key encryption* genannt) unterscheiden sich von symmetrischen Verfahren dadurch, daß es nicht nur einen Schlüssel zur Ver- und Entschlüsselung gibt, sondern ein Schlüsselpaar, wobei ein Schlüssel zur Verschlüsselung, der andere zur Entschlüsselung benutzt wird. Macht man den ersten Schlüssel öffentlich bekannt (den *public key*), können Kommunikationspartner ihre Nachrichten mit diesem Schlüssel chiffrieren, und nur der Inhaber des anderen Schlüssels (des *private key*) kann diese Nachricht wieder dechiffrieren.

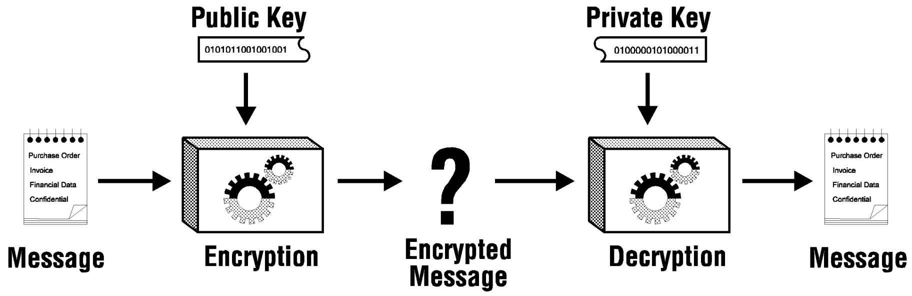

Abb. 10: Asymmetrische Verschlüsselung (aus: [Iann97]).

Sinnvoll ist ebenfalls die Möglichkeit, eigene Nachrichten mit seinem *private key* zu verschlüsseln. Ein Empfänger dieser Nachricht kann diese nur mit dem zugehörigen *public key* entschlüsseln. Auf diese Weise werden digitale Signaturen ermöglicht.

Ein Algorithmus zur asymmetrischen Verschlüsselung muß also die Anforderung erfüllen, daß

- ein mit dem *public key* verschlüsselter Klartext mit dem *private key* wieder entschlüsselt werden kann.
- es extrem schwierig ist, den *private key* aus dem *public key* abzuleiten.
- ein dem *private key* verschlüsselter Klartext mit dem *public key* wieder entschlüsselt werden kann.

Die letzte Voraussetzung ist nötig, wenn der Algorithmus zur Erzeugung von digitalen Signaturen benötigt wird.

Diese Art von Kryptosystem schlugen erstmalig zwei Wissenschaftler der Stanford-Universität, Diffie und Hellman, 1976 vor [DiHe76].

3.2 RSA

Der bekannteste Algorithmus ist RSA. RSA wurde von Rivest, Shamir und Adleman 1977 entwickelt [RSA78] und wird in vielen offiziellen Standards wie SSL [FKK96] und PEM [Kali93] [Kent93] [Linn93] und Programmen wie PGP benutzt.

Laut [Schn96] ist RSA sowohl der am einfachsten zu verstehende als auch zu implementierende Algorithmus mit öffentlichen Schlüsseln. Um die Schlüssel zu generieren geht man wie folgt vor:

- Es werden zwei große zufällige Primzahlen p und q generiert, die ungefähr gleich groß sein sollten.
- Berechne $n = pq$ und $z = (p-1)(q-1)$.
- Generiere eine Zahl e so, daß e und z teilerfremd sind.
- Berechne d, so daß $ed = 1$ *(modulo z)* gilt. Dies kann mit Hilfe des erweiterten Euklidischen Algorithmus gemacht werden (siehe [Schn96, S 246ff]).

Der *public key* besteht nun aus dem Paar e und n, der *private key* aus d und n. Die Zahlen p und q werden nicht mehr gebraucht, sie dürfen aber nicht öffentlich zugänglich gemacht werden.

Um einen Klartext m zu verschlüsseln, wird er in einzelne Blöcke m_i aufgespalten. Bei Zahlen müssen diese Blöcke kleiner als n sein, bei Binärdaten nimmt man als Blockgröße die größte Potenz von 2 kleiner als n. Jeder Block wird jetzt mit der Formel $c_i = m_i^e \mod n$ verschlüsselt. Entschlüsselt werden die Blöcke mit der Formel $m_i = c_i^d \mod n$.

Um das etwas klarer zu machen hier ein Beispiel aus [Schn96, S 467f]. Die gewählten Zahlen sind natürlich viel zu klein als das sie Sicherheit bieten wurden:

- Nehme $p = 47$ und $q = 71$.
- Berechne $n = pq = 3337$.
- Berechne $z = (p-1)(q-1) = 46 * 70 = 3220$.
- Wähle e, hier 79.
- Berechne $d = 79^{-1} \mod 3220 = 1019$.
- Die zu verschlüsselnde Nachricht sei $m = 6882326879666683$. Diese wird nun in Dreierblöcke gespalten. Jeder von ihnen ist kleiner als $n = 3337$. Die entstehenden Blöcke sind nun $m_1 = 688$, $m_2 = 232$, $m_3 = 687$, $m_4 = 966$, $m_5 = 668$ und $m_6 = 003$.
- Der erste Block wird verschlüsselt: $c_1 = 688^{79} \mod 3337 = 1570$.
- Die komplett verschlüsselte Nachricht lautet $c = 1570\ 2756\ 2091\ 2276\ 2423\ 158$.
- Um zum Beispiel m_1 zu dechiffrieren berechnet man $1570^{1019} \mod 3337 = 688$.

Die Sicherheit dieses Verfahrens hängt mit der Schwierigkeit zusammen, große Zahlen ((500 Stellen) zu faktorisieren. Könnte man die öffentliche Zahl n faktorisieren, hätte man p und q. Mit dem erweiterten Euklidischen Algorithmus und e könnte man nun den geheimen Schlüssel d berechnen.

RSA ist, wie alle asymmetrischen Verfahren, um Größenordnungen langsamer als die symmetrischen Verschlüsseungsverfahren. So sind Softwareimplementierungen von DES ungefähr hundertmal schneller als RSA. Public-key-Verfahren werden deshalb meist im Kombination mit symmetrischen Verfahren benutzt: Eine Nachricht wird mit einem symmetrischen Verfahren verschlüsselt. Der Schlüssel dafür kann zufällig gewählt werden. Dieser Schlüssel, der um einiges kleiner ist als die

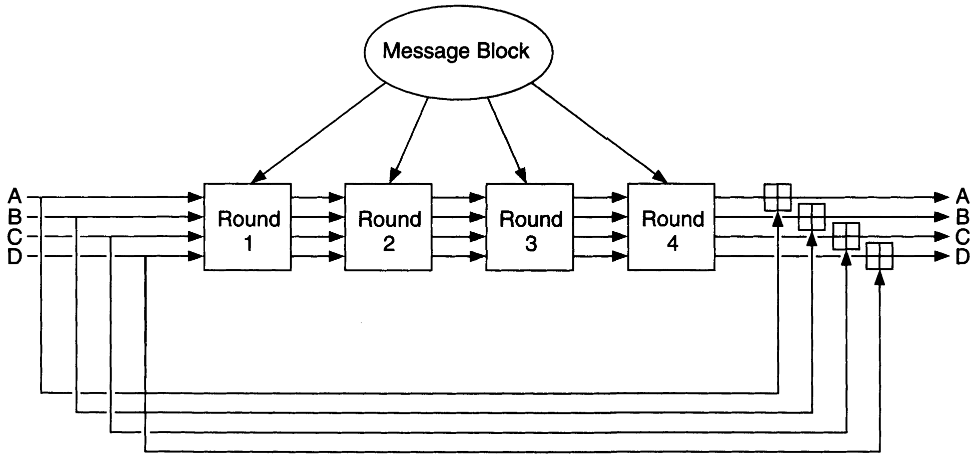

Abb. 11: Die Hauptschleife von MD5 (aus: [Schn96]).

Nachricht, wird dann mit dem asymmetrischen Verfahren verschlüsselt und dann an die Nachricht gehängt.

3.3 Andere Public-Key-Verfahren

Der erste Public-Key-Algorithmus stammt von Ralph. Merkle und Martin Hellman [MeHe78]. Er basiert auf dem NP-vollständigen Rucksackproblem. Der Algorithmus von El Gamal [ElGa85] beruht auf der Schwierigkeit, diskrete Logarithmen zu berechnen.

4 Message-Digests

4.1 Grundlagen

Um digitale Signaturen zu erzeugen, werden Message-Digest benötigt. Diese sind eine Art Fingerabdruck eines Dokuments und sind im allgemeinen mit 128 bis 160 Bit deutlich kürzer als das Dokument. Message-Digest-Algorithmen sind Hash-Funktionen mit den Eigenschaften, daß

- sie leicht aus dem Dokument zu berechnen sind,
- aus dem Message-Digest nicht auf das Ursprungsdokument geschlossen werden kann und
- nur mit extrem hohem Aufwand zwei Nachrichten erzeugen werden können, die den gleichen Message-Digest haben.

Zu den am häufigsten benutzten Algorithmen gehören MD5 und SHA-1.

4.2 MD5

MD5 wurde 1991 von Ron Rivest entwickelt und ist der fünfte Algorithmus einer ganzen Reihe von ihm entwickelter Funktionen. Die Definition und eine Beispielimplementation findet sich in [Rive92b]. MD5 erzeugt einen 128 Bit langen Hash-Code.

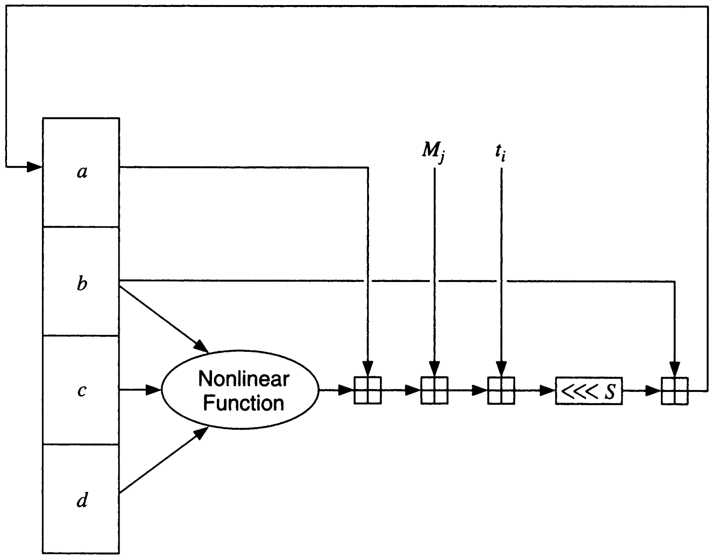

Abb. 12: Eine MD5 Operation (aus: [Schn96]).

Die Ausgangsnachricht wird auf eine Länge von 448 Bit (modulo 512) aufgefüllt und dann die Länge der Originalnachricht als 64-Bit-Zahl angehängt. So ergibt sich als Gesamtlänge ein Vielfaches von 512 Bit. Der 128-Bit-Puffer, der nachher den message-digest enthält, wird mit einem festen Wert initialisiert.

Die Nachricht wird nun in 512 Bit lange Blöcke aufgeteilt und in mehreren Durchgängen mit dem 128 Bit Puffer durchmischt, bis alle Blöcke aufgebraucht sind. Für jeden 512-Bit-Eingabeblock werden dabei 4 Runden durchlaufen (siehe Abb. 11).

Jede Runde wendet eine unterschiedliche Operation (siehe Abb. 12) an. Diese Operation wird jeweils 16 mal durchgeführt. In jeder dieser Operationen wird der bisherige message-digest mit einem 32-Bit-Teilstück des derzeit bearbeiteten 512-Bit-Blocks verknüpft.

4.3 SHA

SHA-1 (Secure Hash Algorithm 1) [NIST94] ist die korrigierte Version des von der National Security Agency (NSA) entwickelten SHA [NIST93b]. Es wurde vom National Institute of Standards and Technology (NIST) zum Standard erhoben. Der Hash-Code ist mit 160 Bit um 32 Bit länger als der von MD5, und damit laut [Tane97] um den Faktor 232 sicherer als MD5.

5 Literatur- und Quellenverzeichnis

[Bal93] D. Balenson. *RFC 1423: Privacy Enhancement for Internet Electronic Mail Part III.* Network Working Group, 1993.

[DiHe76] W. Diffie and M.E. Hellman. „*New directions in cryptography*“. IEEE Transactions on Information Theory, IT-22: 644–654, 1976.

[ElGa85] T. ElGamal. „*A public-key cryptosystem and a signature scheme based on discrete logarithms*“. IEEE Transactions on Information Theory, IT-31: 469 472, 1985

[Fei73] H. Feistel. „*Cryptography and Computer Privacy*“ *Scientific American*, Mai 1973.

[FKK96] A. O. Freier; P. Karlton; P. C. Kocher. *Internet Draft: The SSL Protocol Version 3.0*. Internet Engineering Task Force, 1996 .

[Iann97] M. Iannamico. *PGP User's Manual*. Pretty Good Privacy, Inc, 1997.

[Kali93] B. Kaliski. *RFC 1424: Privacy Enhancement for Internet Electronic Mail Part IV.* Network Working Group, 1993.

[Kent93] S. Kent. *RFC 1422: Privacy Enhancement for Internet Electronic Mail Part II.* Network Working Group, 1993.

[LaMa90] X. Lai; J. Massey. „*A Proposal for a New Block Encryption Standard*“. Advances in Cryptology – Eurocrypt ´90 Proceedings. Springer-Verlag, New York, 1990.

[Linn93] J. Linn. *RFC 1421: Privacy Enhancement for Internet Electronic Mail Part I.* Network Working Group, 1993.

[Mass93] J.L. Massey. „*SAFER K-64: A byte-oriented block ciphering algorithm*“. In Proceedings of 1st Workshop on Fast Software Encryption. S. 1-17, Springer-Verlag, 1993.

[MeHe78] R. C. Merkle; M. Hellman. „*Hiding Information and Signatures in Trapdoor Knapsacks*“. IEEE Transactions on Information Theory, v. 24, n. 5 September 1978.

[NIST93a] National Institute of Standards and Technology (NIST). *FIPS Publication 46-2: Data Encryption Standard*. 1993.

[NIST93b] National Institute of Standards and Technology (NIST). *FIPS Publication 180: Secure Hash Standard (SHS)*. Mai 1993.

[NIST94] National Institute of Standards and Technology (NIST). *Announcement of Weakness in the Secure Hash Standard*. Mai 1994.

[RC93] P. Rogaway and D. Coppersmith. „*A software-optimized encryption algorithm*“. Proceedings of 1st Workshop on Fast Software Encryption. Springer-Verlag, 1993.

[Rive92a] R. L. Rivest. *The RC4 Encryption Algorithm*. RSA Data Security, Inc, 1992.

[Rive92b] R. L. Rivest. *RFC 1321: The MD5 Message-Digest Algorithm*. Internet Activities Board. 1992.

[Rive95] R. L. Rivest. „*The RC5 encryption algorithm*“. CryptoBytes, 1(1): 9-11, 1995.

[RSA78] R. L. Rivest; A. Shamir; L.M. Adleman. „*A method for obtaining digital signatures and public-key cryptosystems*“. Communications of the ACM, 21(2), Februar 1978.

[Schn96] B. Schneier. *Applied Cryptography*. Wiley, New York, 1996.

[Tane97] A. S. Tanenbaum. *Computernetzwerke*. Prentice Hall Verlag, München, 1997.

[Wien94] M. J. Wiener. *Efficient DES Key Search*. TR-244, School of Computer Science, Carleton University, 1994.

Quantenkryptographie

Lars Werbeck

1 Motivation

Praxisrelevante Kryptographiesysteme basieren heutzutage auf der Komplexität mathematischer Probleme. RSA wird zur Verschlüsselung eingesetzt, da es zur Zeit nicht effizient möglich ist, aus dem öffentlichen Schlüssel die privaten Schlüssel zu berechnen, die zur Entschlüsselung notwendig sind. In den letzten Jahren erfolgte auf Basis der Quantenmechanik die Entwicklung neuer Ideen zur Informationsverarbeitung, die zunehmend auch praktische Relevanz bekommen. Quantencomputer sind im Gegensatz zu heutigen klassischen Rechnern theoretisch in der Lage, harte mathematische Probleme, wie die Faktorisierung einer natürlichen Zahl, effizient zu lösen. Sie beruhen auf Prinzipien der Quantenmechanik (siehe Kapitel 2.2).
Kleinere Erfolge in jüngster Zeit (kohärente Zustände im Millisekundenbereich) lassen vermuten, daß die praktische Realisierung nur eine Frage der Zeit ist. Dennoch sind einige Wissenschaftler nicht davon überzeugt, daß Quantencomputer jemals Realität sein werden.
Die Idee, Quanten als Informationsträger zu benutzen, wird unter dem Begriff Quanteninformation zusammengefaßt. Quantenkryptographie benutzt Quanteninformation und quantenmechanische Effekte zur Generierung geheimer Schlüssel zwischen Kommunikationsteilnehmern. Die Quantenkryptographie beruht nicht auf mathematischen, sondern auf physikalischen Prinzipien.

2 Einführung

Die Quantenphysik beschäftigt sich mit den kleinsten bekannten Objekten: Quanten. Während in der klassischen Physik relativ große Systeme Gegenstand der Untersuchungen sind und waren, über die Gesetzmäßigkeiten durch Experimente herausgefunden wurden, stellte sich Anfang dieses Jahrhunderts heraus, daß diese Gesetzmäßigkeiten sich nicht auf beliebig kleine Systeme übertragen ließen. Die Regeln, nach denen die Mikrowelt funktioniert, waren zunächst nur von theoretischem Interesse. Doch schon bald waren praktische Anwendungen, die auf Grundlagen der Quantentheorie beruhen, realisiert. Beispiele hierfür sind Lasertechnik, Kernkraftnutzung (Energiegewinnung, Atombombe) oder Halbleitertechnik. In der heutigen Zeit werden auch verstärkt Experimente möglich, die nicht auf einer großen Menge Quanten operieren, sondern bis in den Bereich isolierter Atome reichen. Daher sind zunehmend auch praktische Anwendungen möglich, die auf Ergebnissen der Quantentheorie kleiner Quantenmengen basieren, die ursprünglich jenseits jeglicher praktischer Relevanz hergeleitet wurden. Erst durch diese Experimente (EPR-Nachweis) sind selbst skeptische Physiker gezwungen, die teilweise paradoxen Folgerungen aus der Quantentheorie zu akzeptieren.

2.1 Quanteninformation

Die Miniaturisierung im Halbleiterbereich mit immer größeren Taktfrequenzen und damit verbundenen kürzeren möglichen Wegstrecken stößt schon in naher Zukunft (zwangsläufig) in Bereiche, in denen fundamentale Auswirkungen durch das Verhalten von Quanten unumgänglich berücksichtigt werden müssen. Quanten sind physikalische Objekte. Sie können unterscheidbare Zustände annehmen. Es ist daher möglich, mit diesen Zuständen Informationen in Relation zu setzen, ähnlich den Spannungspotentialen klassischer Physik in klassischen Computern und deren Interpretation als Informationen.

2.1.1 Qubits

Qubits werden die kleinsten Informationseinheiten auf Quantenbasis genannt. Das Wort ergibt sich aus den Begriffen ‘Quant’ und ‘Bit’. Qubits können in zwei diskreten Zuständen sein, die genau wie in der klassischen Informationstheorie die Werte ‘0’ und ‘1’ oder ‘A’ und ‘B’ oder ‘hoch’ und ‘tief’ repräsentieren können. Jedoch sind sie auch in der Lage, eine Superposition oder Überlagerung von Zuständen anzunehmen. In diesem Fall liefert eine Messung (bezüglich dieser zwei Basen) einen der beiden diskreten Zustände mit jeweils 50-prozentiger Wahrscheinlichkeit. Jedoch befindet sich das Qubit während der Superposition in beiden Zuständen gleichzeitig. Diesen Effekt kann man sich zu Nutzen machen, er macht die vergleichbare Mächtigkeit von Quantensystemen aus.

2.1.2 Lichtpolarisation

Am Beispiel der Polarisation des Lichtes und der Interpretation dieser Polarisationszustände als Qubits sollen die Möglichkeiten der Quanteninformation analysiert werden. Dabei kann man Licht als Teilchenstrom aus Photonen bestehend interpretieren. Im folgenden sollen linear polarisierte Photonen als Qubits eingesetzt werden. Eine Möglichkeit besteht darin, vertikal und horizontal polarisierte Photonen mit den Zuständen 0 und 1 in Relation zu setzen. Durch Polarisatoren ist es möglich, einen Polarisationszustand herzustellen. Als Polarisatoren werden Schirme aus dünnen Drähten (Mikrowelle), Kristalle oder Polaroid-Filter, die aus einer gestreckten Plastikfolie bestehen, die im wesentlichen aus langen parallelen Ketten von Molekülen aufgebaut ist, eingesetzt [Orea79].

Von Interesse ist hier jedoch nicht die physikalische Realisierung, sondern die Eigenschaften linear polarisierter Photonen. Vertikal polarisiertes Licht kann ungestört durch einen Vertikalfilter gelangen, wird jedoch von einem Horizontalfilter blockiert. Daher kann durch zwei senkrecht zueinander hintereinander gestellte Polarisatoren kein Licht dringen. Wird jedoch ein um 45 Grad versetzter Filter zwischenpositioniert, treten 25 Prozent der ursprünglichen Photonen durch die drei Filter.

Werden horizontal polarisierte Photonen durch einen Diagonalfilter bewegt, treten 50 Prozent der Photonen aus. Hier wird ein Problem deutlich: Wie verhält sich ein einzelnes Photon beim Durchtritt durch den Filter. Es stellt sich heraus, daß hierüber nur Wahrscheinlichkeitsaussagen

getroffen werden können. Eine Messung, die ein Polarisator darstellt, zwingt das Photon unvorhersehbar in einen Zustand. Entweder es wird diagonal polarisiert passieren, oder blockiert, mit jeweils 50-prozentiger Wahrscheinlichtkeit [GrGr87].
Vertikalfilter und Horizontalfilter bilden eine Basis im Polarisationsraum, da sie einander ausschließen.

2.1.3 Orthogonale Quantenzustände

Vertikale und horizontale Polarisation sind Photoneneigenschaften, die orthogonale Quantenzustände repräsentieren. Ein weiteres Beispiel bilden die Diagonalpolarisationen 45 Grad und 135 Grad.

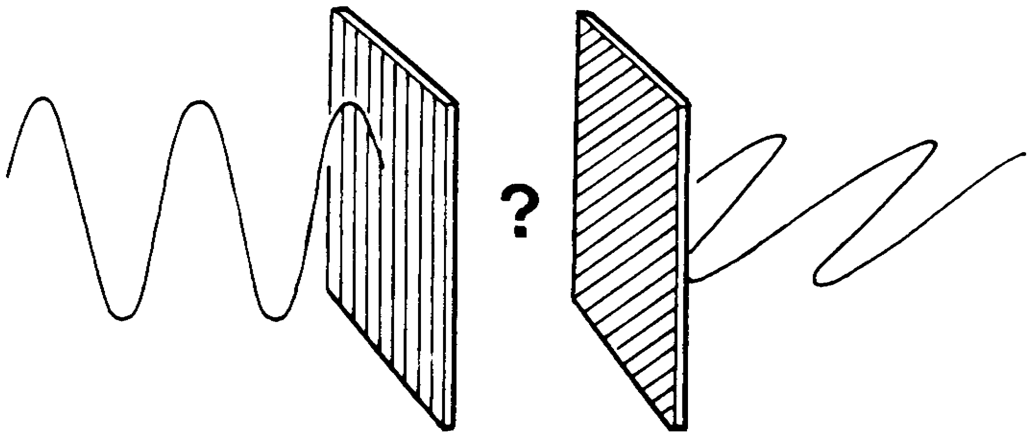

Abb. 1: Zwei um 45 Grad versetzte Polarisatoren lassen 50 Prozent der Photonen passieren

Wann immer Quantenzustände durch eine Messung exakt bestimmt werden können, durch die jeweils andere Messung jedoch keine Aussage möglich ist, liegen orthogonale Quantenzustände vor. Im Bereich der Polarisationsfilter bedeutet dies, daß die jeweiligen Basen sich in einem Winkel von 90 Grad unterscheiden. Diese Eigenschaften werden in der Quantenkryptographie ausgenutzt.

2.1.4 Notation

Um Quantenzustände präzise zu beschreiben, ist eine Notation gebräuchlich, die dem Namen ihres Entwicklers entsprechend Dirac-Notation genannt wird. Dabei werden Quantenzustände durch abstrakte Vektoren wie beispielsweise |V> (vertikale Photonenpolarisation) oder |H> (horizontale Photonenpolarisation) im Hilbert-Raum dargestellt. Ein Hilbert-Raum ist einem komplexen Vektorraum ähnlich. Die oben genannten Vektoren werden ‘kets’ genannt (der rechte Anteil des Wortes ‘bracket’[7]). Die formalen Eigenschaften des Hilbert-Raumes sind für die Quantenkryptographie nicht so relevant wie für den Bereich Quantencomputer. Es sei hier nur darauf aufmerksam gemacht, daß eine der wichtigsten Eigenschaften der Vektoren ihre Länge ist, insbesondere die Länge des Skalar-

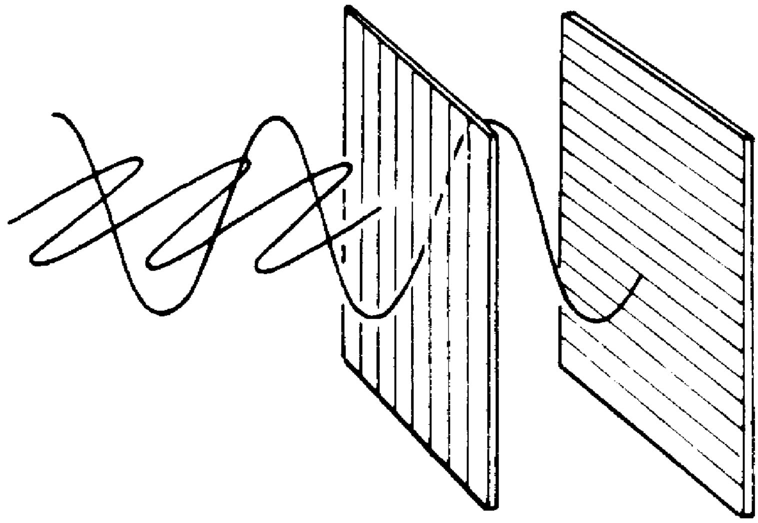

Abb. 2: Zwei senkrecht zueinander gestellte Polarisatoren lassen kein Photon passieren

produktes zweier Vektoren. In der Dirac-Notation werden die komplex konjugierten Vektoren <V| und <H|, die als Gegenstücke zu den 'kets' 'bras' genannt werden, zur Definition des Skalarproduktes wie folgt eingeführt:

$$\langle V|V\rangle = \langle H|H\rangle = 1, \quad \langle V|H\rangle = \langle H|V\rangle = 0.$$

Die Länge eines Quantenzustand-Vektors entspricht der Wahrscheinlichkeit eines experimentellen Ergebnisses oder einer Messung, wobei mathematisch die Idee der Projektion verwirklicht ist [URL-2].

2.2 Quantencomputer

In den frühen 80er Jahren begann Richard Feynman (u.a.) die konventionelle Informationstheorie auf quantenphysikalische Prozesse zu übertragen. Binärzahlen sollten dabei durch mehrere orthogonale Quantenzustände (|0> und |1>) dargestellt werden (siehe Kapitel 2.1.1). Diese Qubits können dabei in beliebiger Superposition sein. Logische Gatter sollten mehrere Qubits als Eingänge haben, wie klassische Gatter auch, jedoch sollte das Ergebnis ebenfalls in von den Eingangsqubits abhängiger Superposition stehen. Dadurch sind nicht nur Operationen auf diskreten Zuständen möglich, sondern auf einer zur Anzahl der Eingänge exponentiellen Anzahl vielen Eingaben simultan. Ein Register der Länge N mit N Qubits der geforderten Form ist in der Lage, gleichzeitig 2^N Eingaben darzustellen. Auf all diesen Eingaben kann nun durch ein Gatter eine Funktion angewendet werden, die die Funktionswerte aller 2^N Eingaben in einem Schritt berechnet.

Dabei treten folgende Probleme auf, die Schuld an der bisher fehlenden praktischen Realisierung sind:

- Superpositionen lassen sich zur Zeit nur 1 Millisekunde stabil halten

---

[7] engl. für Klammer.

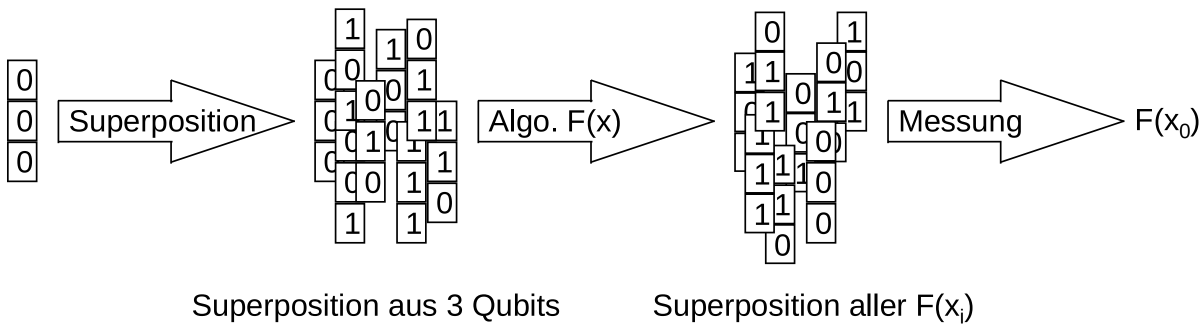

Abb. 3: Funktionsprinzip eines Quantencomputers

- Wechselwirkungen zwischen den Qubits zerstören die Superposition
- Gatter mit vielen Eingängen sind zur Zeit noch zu instabil

Ein Problem der Messung aller Ausgänge ergibt sich dadurch, daß eine Messung die Superpositionen kollabieren läßt. Daher können nur allgemeine Aussagen über die Ausgangsmenge ermittelt werden, wie beispielsweise Perioden, die einfach ausgedrückt durch Interferenzen der Superpositionen meßbar gemacht werden können. Diese Eigenschaften nutzt Shor's Algorithmus zur Primfaktorzerlegung zweier Zahlen aus, dessen Grundgedanken im folgenden Kapitel erläutert werden.

2.2.1 Shor's Algorithmus

Shor's Algorithmus basiert auf der Tatsache, daß die Primfaktorzerlegung auf die Berechnung einer Periode einer Funktion reduziert werden kann [Schr90]. Diese Periode kann mit einem Quantencomputer berechnet werden. Sei die zu faktorisierende Zahl N.

Gesucht ist die Periodizität ($f(x)=f(x+r)=...=f(x+n*r)$) der Funktion $f^N(x) = a^x \bmod N$, wobei $1<a<N$. Für steigende Potenzen von x ist die Funktion periodisch mit Periode r. Die größten gemeinsamen Teiler von $a^{r/2}\pm 1$ und N sind Faktoren von N. Beispiel: N=15, a=11 gewählt. Für steigende x zeigt 11^x mod 15 die Sequenz 1,11,1,11,1,11,...

Daraus folgt für r: r=2. Aus $a^{r/2}=11$ folgt, daß 10 und 12 Kandidaten zur GGT[8] Bestimmung mit 15 sind. Nach dem euklidischen Algorithmus werden GGT(10,15) = 5 und GGT(12,15) = 3 berechnet. Sie liefern die Primfaktoren von N=15.

Shor's Algorithmus hat auf einem Quantenrechner eine polynomielle Laufzeit, im Gegensatz zu den besten Algorithmen auf klassischen Rechnern, die exponentielle Laufzeiten erfordern. Dieser Geschwindigkeitsgewinn würde bisherige Verschlüsselungsmethoden, die auf der Komplexität der Faktorisierung basieren, für den Einsatz in der Kryptographie unbrauchbar machen. Dies motiviert Forschungen in der Quantenkryptographie.

---

[8] GGT = Größter Gemeinsamer Teiler.

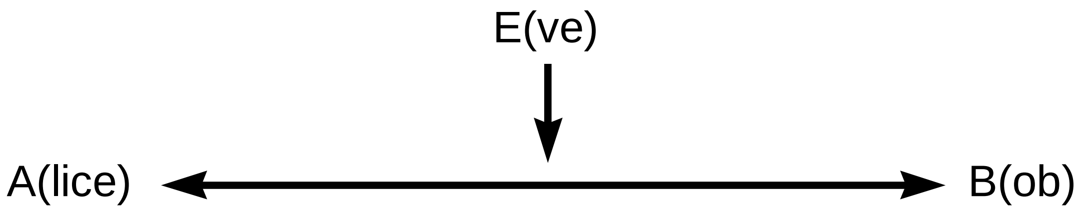

Abb. 4: Traditionelle Bezeichnung der Interaktionsteilnehmer in der Kryptographie

3 Quantenkryptographie

Eine Grundaufgabe der Kryptographie ist der Austausch geheimer Nachrichten zwischen zwei Interaktionspartnern, die traditionell Alice und Bob genannt werden, so daß keine andere Partei den Inhalt dieser Nachrichten lesen kann. Die einzige Möglichkeit, dies auf sichere Weise zu tun, ist ein „one-time pad“[9] zu benutzen, das im Vorfeld zwischen Alice und Bob ausgetauscht werden muß. Dieses sollte die Länge der Nachricht nicht unterschreiten, um selbst Kombinationsrückschlüsse zu vermeiden. Eine so verschlüsselte Nachricht ist nicht zu entschlüsseln, da sie jede Nachricht sein könnte. Dies macht deutlich, daß ein Hauptsicherheitsaspekt von dem Schlüsselgeheimnis abhängt. Da ein Austausch von immer neuen Schlüsseln durch direkte Übergabe oder einen glaubwürdigen Kurier unpraktisch ist, wurden Verfahren auf Basis mathematisch komplexer Probleme entwickelt, von denen das RSA-Verfahren das bekannteste ist. Das vorherige Kapitel hat jedoch deutlich gemacht, daß diese Verfahren nur solange als sicher eingestuft werden können, solange die zugrundeliegenden Probleme nicht effizient gelöst werden können. Für das Problem der Faktorisierung wurde gezeigt, daß es nach der Entwicklung von Quantencomputern nicht mehr zur Verschlüsselung eingesetzt werden könnte. Jedoch bietet die Quantenmechanik nicht nur eine Möglichkeit, bestehende Kryptographiesysteme zu gefährden, sondern auch eine Möglichkeit neuer Arten von Kryptographie, die nicht auf mathematischen Problemen beruhen, sondern auf physikalischen Gegebenheiten. Diese Verfahren beruhen alle darauf, immer wieder ein „one-time pad“ zu generieren, mit dem die Nachrichten auf klassische Weise verschlüsselt werden. Die folgenden Kapitel beschreiben den Vorgang der Generierung dieses geheimen Schlüssels.

3.1 Generierung geheimer Schlüssel

Übliche Verschlüsselung findet zur Zeit über „öffentliche“ Kanäle statt. Dies ist so zu verstehen, daß sämtliche ausgetauschte Daten durch Dritte protokolliert werden können. Die Person, welche diese Angriffe vornimmt, wird traditionell Eve genannt (Abkürzung für *Eavesdropper*[10]). Alice und Bob, die Nachrichten austauschen wollen, können bei heutigen Verfahren nicht feststellen, ob jemand ihre Kommunikation protokolliert. In der Quantenkryptographie ist die Existenz von Eve im Gegensatz

[9] engl. Einmalstempelkissen, Ausdruck für einen Schlüssel, der nur ein einziges Mal benutzt wird.

dazu feststellbar. Dies ist ein Hauptvorteil der Quantenkryptographie, da hierdurch die Sicherheit der Verbindung bezüglich ungewollter Dritter geprüft werden kann, und dadurch ein Austausch ständig neu generierter „one-time pads“ möglich ist.

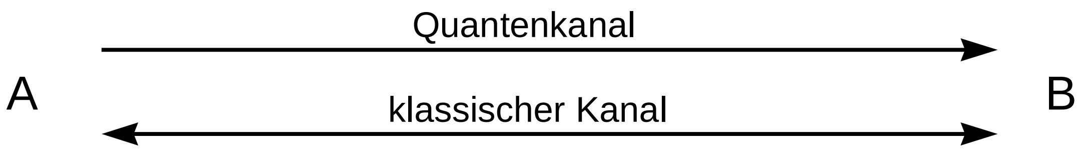

Abb. 5: Benötigte Kanäle in der Quantenkryptographie

3.1.1 4-states-2-observables Systeme[11]

Bei dieser Art der „one-time pad“-Erzeugung werden vier Quantenzustände in Form von Photonenpolarisationen und zwei Basen (Beobachter) eingesetzt. Die vier Photonenpolarisationen bestehen aus je zwei orthogonalen Quantenzuständen, zum einen linear polarisierte Photonen von 0 Grad und 90 Grad, zum anderen linear polarisierte Photonen von 45 Grad und 135 Grad [Rink97].

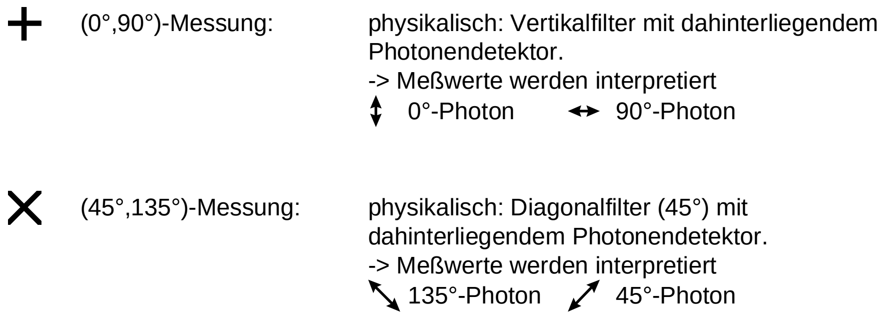

Abb. 6: Physikalische und logische Interpretation

Je Paar wird eine Kodierung der beiden Photonen mit 0 und 1 vorgenommen. Die beiden Basen sind jeweils einem Paar zugeordnet und können ihre Zustände messen. Dies geschieht physikalisch durch das Registrieren des Auftreffen eines Photons hinter dem jeweiligen Filter. Die Abstraktion von der physikalischen Interpretation in die logische ist der Abbildung 6 zu entnehmen. Den Partnern Alice und Bob stehen ein Quantenkanal und ein klassischer Kanal zur Verfügung. Zunächst sendet Alice über den Quantenkanal eine Sequenz von Photonen, die jeweils eine der vier Polarisationen haben, zu Bob. Alice merkt sich die Polarisationen ihrer versendeten Photonen. Bob wählt zufällig je Photon eine der beiden Basen und mißt die Polarisation. Bob merkt sich die Ergebnisse seiner Messungen. Im weiteren Schritt übermittelt Bob die Wahl seiner Basen über den klassischen Kanal an Alice. Dabei muß im Vorfeld eine Kodierung der Basen in 0 und 1 vorgenommen worden sein. Alice vergleicht die von Bob übermittelten Basen mit den von ihr generierten Photonen. Alice sendet nun an Bob über den

[10] engl. Lauscher, Horcher

[11] 4 Quantenzustände, 2 Basen (Beobachter)

klassischen Kanal die Information, welche Basenwahl zu richtigen Ergebnissen, und welche Basenwahl zu falschen Ergebnissen geführt hat. Hierbei ist ebenfalls eine Kodierung möglich, z.B. eine 1 für richtig und eine 0 für falsch. Nun wissen Bob und Alice, welche Informationen sie teilen. Dies werden im Durchschnitt 50 Prozent der ursprünglich von Alice versendeten Photonen sein. Die übrigen Photonenzustände bilden durch ihre Kodierung in 0 und 1 einen gemeinsamen Schlüssel für Alice und Bob, wenn Eve nicht in die Kommunikation eingegriffen hat, wovon nicht ausgegangen werden kann.

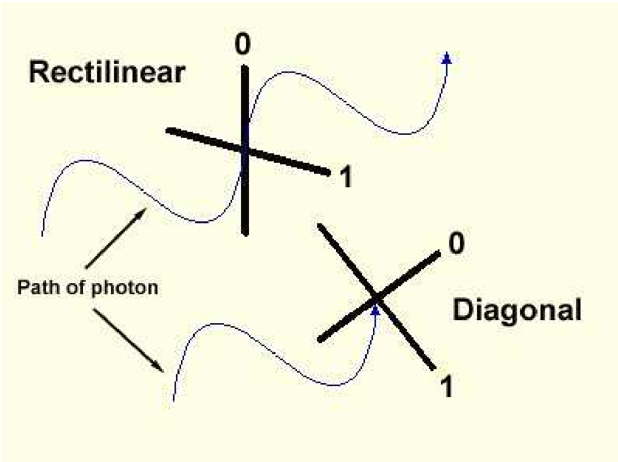

Abb. 7: Kodierung der Photonenzustände in zwei Basen

Daher prüfen Alice und Bob einen beliebig großen Anteil ihres vermeintlichen geheimen Schlüssels durch zufällige Auswahl einiger Bits. Sollte Eve Teile der Photonen gemessen haben, so hätte sie 50 Prozent der Photonenzustände unwiderbringlich zerstört, falls sie ein neues Photon an Bob gesendet hätte. Da Bob auch in diesem Fall nur in der Hälfte der Fälle eine „richtige“ Wahl getroffen hätte, würde sich in diesem (schlimmsten) Fall die Diskrepanz zu Alice auf 75 Prozent erhöhen. Gemessen an der Größe des vermeintlichen Schlüssels wäre dies eine Fehlerquote von 50 Prozent. Vergleichen Alice und Bob zufällig einige Bits des Schlüssels und geben damit auch die Benutzung dieses Teiles des Schlüssels zur Verschlüsselung auf, kann mit beliebiger Genauigkeit eine Aussage über die Existenz von Eve gemacht werden. Den Rest des vermeintlichen Schlüssels benutzen Alice und Bob einmalig zur Verschlüsselung einer Nachricht nicht viel größer als der Schlüssel, falls sie mit dem erfolgten Teilschlüsselvergleich zufrieden sind. Ansonsten wiederholt sich das ganze Verfahren, bis sie mit der Verbindung zufrieden sind.

In Tabelle 1 wird ein beispielhafter Verlauf der Schlüsselgenerierung dargestellt. Zunächst sendet Alice an Bob eine Sequenz von Photonen mit den dargestellten Polarisationen. Bob mißt diese Photonen jeweils mit einer zufällig gewählten Basis (Zeile 2). Die Ergebnisse der Messungen sind in Zeile 3 festgehalten. Nun sendet Bob die Art seiner Messung an Alice, wobei er die 0 für eine Diagonalbasis, die 1 für die andere Basis wählt. Diese Kodierung muß natürlich im Vorfeld vereinbart worden sein. Alice sendet nun ihrerseits die Richtigkeit der Basenwahl in intuitiver Kodierung zu Bob. Die letzte

A -> B	⤢	↕	↔	↔	⤡	↕	⤢	↕
B	×	×	+	×	+	+	×	×
B	⤢	⤢	↔	⤡	↕	↕	⤢	⤡
B -> A	0	0	1	0	1	1	0	0
A -> B	1	0	1	0	0	1	1	0
k e y	1	-	0	-	-	1	1	-

Tab. 1: Beispielhafte Schlüsselgenerierung im 4-states-2-observables-System

Zeile zeigt den erzeugten Schlüssel, der als Teilmenge aller übermittelten Photonen übrig bleibt. Diese Kodierung ist ebenfalls vereinbart worden. Sie ist der Tabelle aufgrund der ersten Zeile zu entnehmen.

3.1.2 2-states Systeme[12]

Eine alternative Möglichkeit zur Generierung eines geheimen Schlüssels mit Hilfe der Quantenkryptographie besteht darin, zwei nicht orthogonale Polarisationen zu benutzen, beispielsweise 0 Grad polarisierte Photonen und 45 Grad polarisierte Photonen, sowie die bekannten Basen (siehe Kapitel 3.1.1). Alice generiert eine zufällige Sequenz aus Photonen, die eine der beiden Polarisationen aufweisen, und sendet diese über den Quantenkanal zu Bob. Wieder merkt sich Alice die Wahl ihrer Photonenpolarisationen. Bob wählt je Photon eine zufällige Basis und mißt das Photon damit. Bei gegebener Kodierung der beiden Zustände in 0 und 1, kann Bob direkt bei der Messung feststellen, ob er einen der Zustände 0 oder 1 gemessen hat. Oder er weiß, daß er keine Aussage über den von Alice gewählten Polarisationszustand des Photons machen kann. Dies kommt daher, daß kein 45 Grad polarisiertes Photon eine Messung eines 135 Grad Photons ermöglicht, wohl aber ein 0 Grad polarisiertes Photon diese Messung ermöglicht, falls zur Messung die Diagonal-Basis benutzt wird. Im anderen Fall kann kein 0 Grad polarisiertes Photon eine Messung eines 90 Grad Photons ermöglichen, wohl aber ein 45 Grad Photon, bei Wahl der anderen Basis. Auf diese Weise kann ein Viertel der Photonen durch Rückschluß der möglichen Ausgangsphotonen richtig ermittelt werden. Bob sendet nun an Alice eine Sequenz, die Aufschluß über die Richtigkeit der von Bob gemachten Rückschlüsse enthält, wohl aber nicht die ermittelten Zustände. Alice und Bob haben zu diesem Zeitpunkt ihren vermeintlichen ge-

[12] 2 Quantenzustände

heimen Schlüssel generiert. Jedoch auch hier muß die Möglichkeit einer Manipulation durch Eve überprüft werden. Dazu werden wieder Teile des vermeintlichen Schlüssels verglichen. Sollte Eve eingegriffen haben, so können folgende Fälle auftreten: Eve hat einen Photonenzustand erkannt. Dann kann sie mit einer Wahrscheinlichkeit von ¼ erreichen, daß Bob diesen Photonenzustand ebenfalls erhält. Falls Eve den Photonenzustand nicht richtig ermitteln kann (3/4 der Fälle), kann sie nur einen beliebigen Zustand an Bob weiterleiten. Dieser ist mit 50-prozentiger Wahrscheinlichkeit der richtige, und mit einer Wahrscheinlichkeit von ¼ kann Bob ihn feststellen. Dies bedeutet aber auch, daß Eve mit einer Wahrscheinlichkeit von 3/32 durch ihren Eingriff in das System ein falsches Bit erzeugt, das bei der späteren Kontrolle durch Alice und Bob auftauchen könnte. Bei einer großen Menge Bits im vermeintlichen Schlüssel und relativ dazu großem Vergleich der Werte durch Alice und Bob, was natürlich erneut eine Reduzierung des brauchbaren Schlüssels bedeutet, wird Eve mit beliebig großer Wahrscheinlichkeit entdeckt. Daher ist auch dieses Verfahren zur Generierung eines „one-time pad“ geeignet. Verglichen mit der ersten Methode, bei der Eve gemessen am vermeintlichen Schlüssel 25 % Diskrepanz erzeugte, sind es hier 37,5 %. Wie im ersten Fall kennt Eve maximal 25 % des vermeintlichen Schlüssels.

A -> B	↕	↕	⤢	↕	⤢	↕	⤢	⤢	⤢	↕	↕	⤢
B	×	+	+	×	×	×	+	×	×	+	×	+
B	⤢	↕	↔	⤡	⤢	⤢	↔	⤢	⤢	↕	⤢	↕
B	?	?	1	0	?	?	1	?	?	?	?	?
B -> A	0	0	1	1	0	0	1	0	0	0	0	0

Tab. 2: Beispielhafte Schlüsselgenerierung im 2-states-System

In Tabelle 2 ist eine beispielhafte Schlüsselgenerierung im 2-states-System verdeutlicht. In einem ersten Schritt sendet Alice eine Sequenz Photonen mit dargestellter Polarisation an Bob. Bob entscheidet für jedes Photon über die Art seiner Messung: Enweder er wählt die Diagonalbasis oder die Rechtlinearbasis. Seine Ergebnisse sind in Zeile 3 festgehalten. Nun kann er direkt feststellen, welche gemessenen Werte in jedem Fall richtig sind: Nur das Ergebnis horizontaler Polarisation kann ihn beim Rechtlinear-Test auf ein 45-Grad Photon bringen, da ein 0-Grad Photon diese Messung niemals hätte initiieren können. Ebenso sieht es beim Messen eines 135-Grad Photons beim Messen mit der Diagonalbasis aus: Nur ein 0-Grad polarisiertes Photon kann dieses Meßergebnis hervorrufen. Alle weiteren Meßergebnisse lassen keinen Schluß auf die Polarisation des Ausgangsphotons zu. Die Kodierung der beiden Quantenzustände muß im Vorfeld erfolgt sein. Zuletzt sendet Bob an Alice die von ihm richtig erkannten Photonenpositionen, indem eine 0 für nicht erkannt und eine 1 für erkannt benutzt wird.

3.1.3 Verschränkte Zustände korrelierter Quanten

Eine dritte Möglichkeit, quantenmechanische Effekte in die Kryptographie einfließen zu lassen, ist die Verwendung sogenannter EPR-Photonen (nach einem Gedankenexperiment von Einstein, Rosen, Podolsky). Hierbei werden korrelierte Photonen durch Kristalle erzeugt, deren Polarisationen senkrecht zueinander stehen (nicht notwendigerweise, es gibt auch gleich-korrelierte Photonen-Paare). Der Trick besteht nun darin, diese gemeinsame Eigenschaft des Photonen-Paares auszunutzen. Denn eine Messung an dem einen Photon wirkt sich auf das andere Photon derart aus, daß es weiterhin seine Eigenschaft der orthogonalen Polarisation behält. Eine einfache Version könnte dadurch realisiert werden, daß Alice und Bob je ein Photon des korrelierten Paares erhalten. Jeder führt an seinem Photon eine Messung aus. Über einen klassischen Kanal tauschen Alice und Bob die Art ihrer Messung aus. Sollten ihre Messungen verträglich sein, teilen sie (bei gegebener Kodierung) ein Bit für einen vermeintlichen Schlüssel. Dieses Verfahren wird fortgesetzt, bis eine beliebig große Anzahl Bits für den Schlüssel generiert sind. Auch hier ist durch Teilauswahl einzelner beliebiger Bits des vermeintlichen Schlüssels und öffentlichem Vergleich einer beliebig großen Wahrscheinlichkeit die Existenz von Eve nachweisbar. Die Probleme dieses Verfahrens sind, daß es bisher nur über kurze Wegstrecken realisiert werden kann, da durch Wechselwirkungen die Korrelationen schnell zerstört werden können.

4 Zusammenfassung

Während Quantencomputer noch einige Zeit und technologische Entwicklung erfordern, um realisiert zu werden, ist im Bereich der Quantenkryptographie mit einigen Erfolgen bereits eine praktische Relevanz geschaffen worden. Quantenkanäle können bereits über eine Distanz von 50 Kilometern operieren. Dennoch ist eine größere Verbreitung im kommerziellen Sektor aufgrund der komplizierten und teuren Technologien noch eine Frage der Zeit. Auch sollten nicht die Probleme unbemerkt bleiben, die eine strikte Umsetzung der Theorie auch hier noch erschweren: Einzelne Photonen sind schwer herzustellen und Quantenkanäle können nicht absolut von Wechselwirkungen befreit werden (*noisy channels*[13]). Da jedoch, wie in der Einführung bereits erwähnt, die ständige Miniaturisierung heutiger Rechner zwangsläufig immer mehr in quantenphysikalisch relevante Bereiche vorstoßen läßt, wird die Quantenphysik in der Zukunft fester Bestandteil der Entwicklung von Computern sein, und damit auch Bestandteil von Algorithmen auf quantenphysikalischer Basis unter den Gesetzen, die die Quantenphysik vorgibt.

5 Literatur- und Quellenverzeichnis

5.1 Printmedien

[Orea79] Jay Orear. *Physik*. Carl Hanser Verlag, München 1979.

[Schr90] M.R. Schroeder. *Number Theory in Science and Communication*. Springer, 1990

[Rink97] Jürgen Rink. „Alice im Wunderland - Quantenrechner: Auf dem Sprung zur Realität?“. *c’t Magazin für Computertechnik*. Heise Verlag, Hannover, Nr.3, 1997.

[GrGr87] John Gribbin; Mary Gribbin. *Auf der Suche nach Schrödingers Katze*. Piper, München, 1987

[Fick88] Eugen Fick. *Einführung in die Grundlagen der Quantentheorie*. Aula-Verlag, Wiesbaden, 1988, 6. Auflage

[Hols92] Barry R. Holstein. *Topics in Advanced Quantum Mechanics*. Addison-Wesley, Redwood City (Kanada), 1992

5.2 Elektronische Dokumente

[URL-1] A Bibliography of Quantum Cryptography by Gilles Brassard (gesichtet 9.11.1997):
`http://www.iro.umontreal.ca/~crepeau/Biblio-QC.html`

[URL-2] Quantum Information at Los Alamos National Laboratory (gesichtet 10.9.1997):
`http://p23.lanl.gov/Quantum/quantum.html`

[URL-3] Quantum Cryptoanalysis - Introduction by Artur Ekert (gesichtet 11.1.1998):
`http://eve.physics.ox.ac.uk/Qcresearch/cryptoanalysis/qc.html`

[URL-4] Quantum Cryptography with Coherent States by T. Mor (gesichtet 5.1.1998):
`http://feynman.stanford.edu/`
`qcomp/huttner/bruno-27feb95-txt/bruno-27feb95-txt.html`

[URL-5] Quantum communication moves into the unknown (gesichtet 11.1.1998):
By David Deutsch and Artur Ekert
`http://eve.physics.ox.ac.uk/`
`NewWeb/Research/communication/communication.html`

[URL-6] OLIVER POZO‘S QUANTUM PAGE (gesichtet 11.1.1998):
`http://www.ae.utexas.edu/~pozo/quantum/quantum.html`

[URL-7] Secret Key Agreement by Public Discussion: Bibliography (gesichtet 18.12.1997):
`http://www.inf.ethz.ch/`
`department/TI/um/keydemo/Bibliography.html`

[13] engl. Rauschen in Kanälen

[URL-8] Index of reviews of papers in Quantum Computing and Cryptography (gesichtet 10.12.1997):
http://aerodec.anu.edu.au/~qc/reviewindex.html

[URL-9] Quantum Optics and Foundations of Physics Institut für Experimentalphysik Universität Innsbruck (gesichtet 20.12.1997):
http://www.uibk.ac.at/c/c7/c704/qo/index.html

Sicherheit auf der physikalischen Schicht

Michael Pohé

1 Allgemeines

Inhalt dieses Seminarbeitrags ist die Sicherheit auf der physikalischen Schicht. Aufbauend auf einem Grundverständnis der Begriffe soll gezeigt werden, wo bereits auf der physikalischen Schicht Datenschutz und Datensicherheit betrieben wird bzw. wo die Lücken sind, die durch eine „aufgesetze“ Software abgedichtet werden müssen. Allgemein läßt sich sagen, daß eine bereits in der Hardware implementierte Vorrichtung, sei es nun den Datenschutz oder die Datensicherheit betreffend, in der Entwicklung und Produktion teurer ist, ihre Aufgabe jedoch durch den Einsatz zusätzlicher Mikroprozessoren, welche speziell für diese Aufgaben entwickelt wurden, schneller erledigen kann. Gerade wenn es sich um zeitkritische Aufgabenstellungen handelt und der Schutzalgorithmus die CPU nicht zusätzliche belasten soll, ist eine Hardwareimplementierung überlegenswert.

Weiterhin kann eine Hardwarevorrichtung leichter gegen Manipulationen geschützt werden. Was zunächst nach ein extremer Vorteil zu sein scheint, hat auch seine Nachteile – nämlich genau dann, wenn die Schutzmaßnahmen nicht mehr ausreichen und aktualisiert, sprich erweitert, werden sollen. Nun ist eine Hardware nicht so flexibel in ihrer Änderbarkeit wie eine Software.

Auf Schutzmaßnahmen auf der physikalischen Schicht kann überhaupt nicht verzichtet werden, wenn mit physikalischen Mitteln angegriffen wird. Dann sind Sicherheitsmaßnahmen mit physikalischen Mitteln nicht verzichtbar, wie aus Kapitel 3.3 ersichtlich ist.

2 Datensicherheit auf der physikalischen Schicht

Datensicherheitsmaßnahmen können in zwei Kategorien aufgeteilt werden. Die einen dienen dazu, Fehler zu vermeiden, während die anderen bereits aufgetretene Fehler korrigieren um deren Auswirkungen zu lindern. Hierbei sind die Präventivmaßnahmen eindeutig vorzuziehen, da eine Korrektur unter Umständen zu Datenverlust, Zeitverlust oder Geldverlust führen kann, jedoch auf alle Fälle Nerven kosten wird.

Ein paar Maßnahmen zur Erhöhung der Datensicherheit, also zum Schutz gegen Datenverlust nicht gegen Datendiebstahl, die hier weiter nicht betrachtet werden sollen, seien nur kurz genannt und in ihrer Funktionsweise umrissen:

- **USV**

  USV steht für unterbrechungsfreie Stromversorgung. Wenn es im Stromnetz zu Spannungsschwankungen bzw. zum kompletten Stromausfall kommen sollte, reicht die Stromversorgung durch die USV aus, um auf allen angeschlossenen Rechner ein ordnungsgemäßen Beenden zu gewährleisten, alle in Bearbeitung befindlichen Daten auf Festplatte zu sichern und so einem unkontrollierten Systemabbruch mit der Gefahr des Datenverlustes zu entgehen.

Man unterscheidet die sogenannten Online- und Offline-Modelle. Das Offline Modell arbeitet wie ein Notstromaggregat und schaltet sich erst dann ein, wenn der Strom ausgefallen ist. Dadurch entstehen Schaltzeiten, die im Bereich von wenigen Millisekunden liegen. Bei einem IBM-kompatiblen Computer reicht diese Reaktionsgeschwindigkeit in der Regel aus, um die Gefahr eines Datenverlustes auszuschalten, bei anderen Systemen kann schon diese kurze Umschaltzeit zu Spannungsschwankungen führen, welche die Systemintegrität gefährden. Wenn diese latente Gefahrenquelle nicht akzeptabel ist, wird eine Netz-interkative USV, eine Weiterentwicklung der Offline-Technologie, verwendet. Diese USV ist parallel zur Spannungsversorgung geschaltet und gleicht Spannungsschwankungen auf für die Verbraucher tolerierbare Werte aus ohne direkt auf Batteriebetrieb umzuschalten. Aber auch hier kann es beim Umschaltvorgang zu Problemen kommen. So reagieren z.B. Telekommunikationsanlagen empfindlich auf die unvermeidbaren Phasenverschiebungen.

Hier kommt dann das Online-Modell zum Einsatz. Die erhöhte Datensicherheit wird durch aufwendigere Technik erkauft und damit ist die Online-Variante i.d.R. teurer als die Offline Variante. Die eingehende Spannung wird galvanisch getrennt und gleichgerichtet, anschließend geglättet, stabilisiert und wieder in Wechselspannung zurückgewandelt. Das Ergebnis ist eine ideale, sinusförmige Wechselspannung ohne Störimpulse, die empfindliche Computerbauteile in ihrer Funktion beeinträchtigen könnten. Dieses Verfahren gewährleistet für den Verbraucher weitestgehend Abschirmung vor Spannungsschwankungen, Unterbrechungen, Rauschen oder Spannungsspitzen. Die Batterie wird während des Normalbetriebs aufgeladen und übernimmt im Fall eines kompletten Stromausfalles die gesamte Spannungsversorgung, ohne das es zu bemerkbaren Spannungsschwankungen oder Phasenverschiebungen kommt. Da permanent eine eigene, ideal-sinusförmige Wechselspannung erzeugt wird, kommt es beim Umschalten zu keiner Gefährdung der Betriebssicherheit. Die Online-USV übernimmt so nebenbei auch noch die Funktion eines Spannungskonstanthalters und Netzfilters.

- **NVRAM**

  NVRAM-Bausteine (engl. non volatile RAM) sind nicht-flüchtige Speicherbausteine. Während des normalen Betriebs verhalten sie sich wie normales RAM, sind also relativ schnell im Vergleich zu EEPROM-Speicher [Mar96]. Kommt es zu einem Absinken der Betriebsspannung, wird ihr Inhalt in ein EEPROM kopiert. Bei Rückkehr der Spannung kann so der genaue Speicherinhalt wiederhergestellt werden. Da diese Speicherarchitektur sehr aufwendig und kostspielig ist werden in der Anwendung nur Festplattencaches mit dieser Speicherart versehen. So kann in Datenbanksystemen gewährleistet werden, daß auch bei Stromverlust die Daten konsistent bleiben.

- **Backup**

  Eigentlich sollte dieser Punkt nicht erwähnungsbedürftig sein, aber leider gibt es immer noch Betriebe, in denen die regelmäßige Datensicherung nicht selbstverständlich ist. Obwohl dies eine traurige aber bekannte Tatsache ist, wird die Gefahr die in einen Datenverlust steckt, leider verharmlost. Dabei melden über 70% der Firmen, die einen kompletten Datenverlust erlitten haben, innerhalb der nächsten zwei Jahre Konkurs an.

- **CO_2**

  In einem Computerraum darf selbstverständlich kein herkömmliches Feuerlöschsystem installiert werden. In kleineren Betrieben ist in der Regel sowieso keine Feuerschutzmaßnahme ergriffen worden. Wenigstens sollten die auf Band o.ä. gesicherten Daten nicht im gleichen Raum bzw. im gleichen Gebäude aufbewahrt werden. Wenn es sich um teure Hardware und wichtige Daten handelt, ist der Einbau einer CO_2-Löschanlage erwägenswert. So kann ein Feuer bekämpft werden, ohne daß die Daten und Computer unabwendbar zerstört werden.

Datensicherheitsmaßnahmen sind als Versicherung zu sehen. Es kann (wird hoffentlich) passieren, daß man alle Präventivmaßnahmen umsonst angeschafft hat. Falls es jedoch zu einem Zwischenfall kommen sollte, freut man sich im Nachhinein über jede investierte Mark.

2.1 RAID Systeme

Der Begriff RAID wurde 1987 von Patterson, Gibson und Katz an der Berkley Universität in Kalifornien geprägt. Sie veröffentlichten ein Fachzeitschriftsbeitrag mit dem Titel „A Case for Redundant Arrays of Inexpensive Disks (RAID)[14]“. In diesem Beitrag wurden verschiedene Typen eines Verbundes von Datenträgern, dargestellt. Dabei waren zwei Gesichtspunkt von besonderer Bedeutung: die Erhöhung des Datendurchsatzes und die Steigerung der Datensicherheit bei einer Verwendung möglichst preisgünstiger Datenträger.

Ein Problem beim Einsatz von RAID-Systemen mit vielen Festplatten ist, das die MTBF[15] eines Festplattenverbundes gleich der MTBF eines Einzelgerätes dividiert durch die Anzahl der Geräte im Array

[14] RAID: redundanter Verbund preiswerter Speichermedien

[15] Mean Time between Failure: Die heuristisch berechnete Zeit zwischen zwei Fehlverhalten eines Systems

ist. Da die Anzahl der Datenträger gleichzeitig ausschlaggebend für die Leistungsfähigkeit des Arrays ist, gilt es hier einen vernünftigen Kompromiß je nach Anwendungsfall zu wählen bzw.
Kommt es zum Ausfall eines Datenträgers hängt es vom verwendeten RAID-Typ ab, ob es zu einem Datenverlust kommt. Handelt es sich um einen Typ mit ECC (Error Correction Code[16]), kann in der Regel die Festplatte ohne Datenverlust bei laufendem System ausgewechselt werden.
Im Beitrag nennen die Autoren 5 RAID-Typen. Allen gemeinsam ist, daß sie i.d.R. dem System als ein logisches SCSI-Laufwerk erscheinen und wie eine handelsübliche Festplatte angesprochen werden können. Zum Einsatz kommen sowohl plattformunabhängige RAID-Kontroller (Abb. 1), welche als externes SCSI-Laufwerk agieren und interne, systemspezifisch angepasste Geräte. Die Verwaltung des Datenflusses, also die Aufteilung und Zusammenführung des Datenflusses sowie das Bilden der Fehlerkorrekturdaten, werden von einem speziellen RAID-Kontroller übernommen. Dieser restauriert im Falle eines Datenfehlers auf einem Speichermedium aus den Fehlerkorrekturdaten die ursprünglich gespeicherten Daten. Der Korrekturvorgang schlägt sich negativ auf die Lesegeschwingikeit auf.
Zwar gibt es auch softwarebasierte Lösungen, jedoch unterstützen diese nur Teile der RAID-Spezifikation. Einfache Typen, wie das Festplattenspiegeln (Kap. 2.1.2) oder das rolierende Schreiben von Datenfragmenten (Kap. 2.1.1) werden unterstützt, Typen mit rechenintensive Fehlerkorrekturalgorithmen werden nicht unterstützt, da diese Zusatzaufgaben durch den Hauptprozessor zusätzlich übernommen werden müßten und so eine zu hohe Systembelastung die Folge wäre.

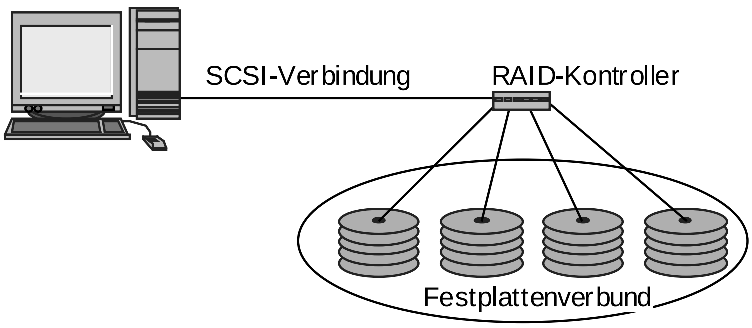

Abb. 1: Typische Anordnung eines plattformunabhängigen RAID-Festplattenverbundes

Handelsübliche RAID-Kontroller beschränken sich auf die Realisierung der Typen 0, 1 und 5. Wie aus den folgenden Kapiteln ersichtlich ist, werden dem Benutzer mit ihnen die für die Praxis interessanten Typen zugänglich gemacht. Während die Typen 0 und 1 die Basisdienste Performanzsteigerung und Datenredundanz bieten, vereint Typ 5 die Vorteile der anderen Typen und bietet einen vernünftige Kompromiß bezüglich der Datensicherheit und Datenflußgeschwindigkeit.

[16] Fehlerkorrekturcode; Verwendet wird derselbe Algorithmus wie bei fehlerkorrigiernden RAM-Bausteinen und modernen Festplattenkonzepten

2.1.1 Der RAID-Level 0 – Datensegmentierung

Der Typ 0 ist ein nicht-redundater Festplattenverbund ohne Paritätsprüfung. Mehrere, idealerweise gleich große Festplatten, werden in sogenannte *stripes*[17] unterteilt. Die Größe dieser Datensegmente variiert, je nach Anwendung, von der Sektorengröße der verwendeten Speichermedien bis zu mehreren Megabyte.

Der Schreibvorgang verläuft zyklisch auf alle Festplatten im Verbund gleichmäßig verteilt (Abb. 2). Dadurch wird eine Schreibgeschwindigkeit erreicht, welche weit über der Geschwindigkeit der verwendeten Einzelgeräte liegt, da bei der Datenübertragung zu den Festplatten nicht auf das Ende des langsamen, physikalischen Schreibvorganges gewartet werden muß.

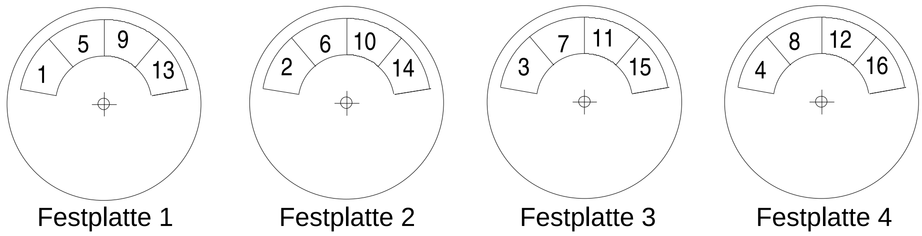

Abb. 2: Schreib-/Lesereihenfolge auf die Festplatten im Verbund beim RAID-Level 0

Damit der Lesevorgang gleichermaßen beschleunigt werden kann, müssen die Festplatten synchronisiert werden. Ansonsten kann es vorkommen, daß die Zugriffszeit sich dem Worst-Case-Fall einer einzelnen Festplatte nähert. Optimal ist es, wenn die Schreib-/Leseköpfe der Festplatten die gleiche relative Position zueinander haben wie beim Schreibvorgang. Erst wenn das letzte Datenpaket ankommt, ist der Lesevorgang beendet.

Ein weiterer Vorteil des rolierneden Schreibens von Datensegmenten ist, daß ein gleichmäßiger Füllstand aller Festplatten im Verbund gewährleistet ist.

Wenn man sich streng an die Definition des Begriffs RAID hält, ist der Typ null keine RAID-Klasse im eigentlichen Sinn, da er keine erhöhte Datensicherheit bietet. Daher kommt es beim Versagen eines Datenträgers im Festplattenverbund unweigerlich zum kompletten Systemversagen und Datenverlust. Eine Datenrekonstruktion ist nur durch das Wiedereinspielen einer unabhängig vom RAID-System durchgeführten Datensicherungskopie möglich. Trotzdem wird Typ 0 als RAID-Basistyp genannt. Als schnellste RAID-Variante wird er in vielen Computersystemen zur Datenflußoptimierung softwaremäßig implementiert.

2.1.2 RAID-Level 1 – Spiegelung

Level 1 ist der schnellste, datenredundante Raid-Typ. Dabei wird jedes Datensegment auf zwei Festplatten gespeichert. Anders als bei Level 0 ist der Inhalt der beiden Festplatten jedoch identisch. Die

[17] engl. für das Aufteilen der Dateien in kleine Datensegmente konstanter Länge.

zweite Platte ist eine Spiegelung der ersten Festplatte. Durch die doppelte Datenhaltung kann bei Ausfall einer Festplatte normal weitergearbeitet werden. Sogar eine Reparatur bei laufendem Server ist möglich. Während der Schreibvorgang nicht beschleunigt wird, kann beim Lesevorgang auf zwei Platten zugegriffen werden. Da sich die Leseoperationen überlappen können, hat der Typ 1, neben den Sicherheitsvorteilen, auch deutliche Lesegeschwindigkeitsvorteile gegenüber dem Ursprungssystem.
Wenn mehrere Festplattenverbunde des Typs 1 verbunden werden und dem System als eine logische Einheit erscheinen, spricht man vom *Dual-level array* oder RAID Typ 10. (Abb. 3)
Falls die Festplatte und ihre Spiegelung an unterschiedlichen RAID-Kontrollern angeschlossen sind, spricht man statt vom Spiegeln vom Verdoppeln[18]. Hier ist die Datensicherheit noch größer, da das System nun auch bei einem Kontrollerausfall weiter betriebsbereit ist.

Abb. 3: Schreib-/Lesereihenfolge beim RAID-Level 10

2.1.3 RAID-Level 2

Die Daten werden wie bei Typ 0 in Datensegmenten auf eine Gruppe von Festplatten verteilt. Der ECC wird jeweils auf einer anderen Festplatte als die Daten gespeichert (Abb. 4) Beim Einsatz mehrerer Korrekturcodefestplatten kommt es auch beim Versagen von zwei Festplatten nicht zum Datenverlust.
Bei kurzen Datentransfers wird die effektive Leseperformanz schlechter, da aufgrund der Datenaufteilung in kleine Segmente auch bei geringen Datenmengen alle Festplatten im Verbund Datenteile enthalten.

[18] in der Literatur wir auch der Begriff *duplexing* verwendet

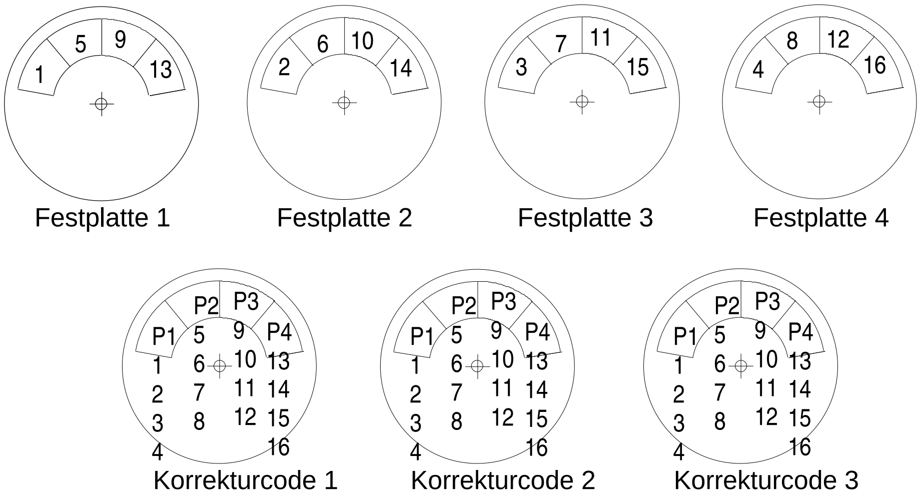

Abb. 4: Schreib-/ Lesereihenfolge mit ECC-Bildung bei RAID-Level 2 unter Einsatz mehrer ECC-Festplatten

2.1.4 RAID-Level 3

Ähnlich wie beim Typ 2 hat dieser Typ ebenfalls eine Festplatte zum Speichern der Paritätsinformationen (Abb. 5). Wenn eine Festplatte ausfällt, kann aus den Informationen der anderen Festplatten im Verbund durch XOR-Verknüpfungen der Inhalt des ausgefallenen Mediums rekonstruiert werden. Gegenüber RAID-1 wird also deutlich weniger Festplattenplatz zum Erreichen der gleichen Datensicherheit benötigt, jedoch wird dies durch ein aufwendigeres Verfahren erkauft.

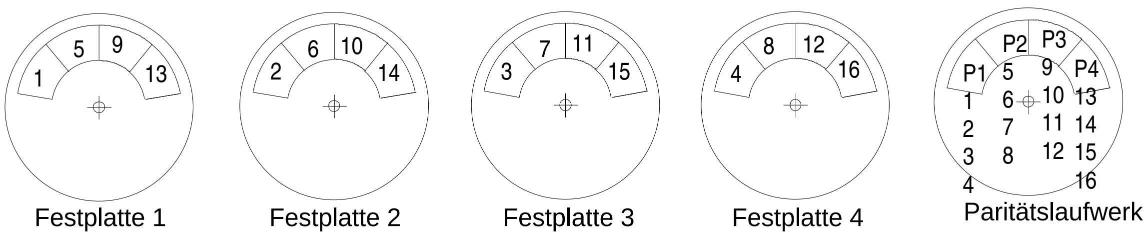

Abb. 5: Schreibvorgang bei RAID-Level 3 mit gesonderter Paritätsfestplatte

Kritisch für die effektive Schreibgeschwindigkeit ist die Paritätsbildung. Bei einem Schreibzugriff müssen zunächst alle Bereiche auf den Festplatten ausgelesen werden, in welche die neuen Daten geschrieben werden sollen. Aus dem ursprünglichen Inhalt und den neuen Daten wir die Parität gebildet. Anschließend erfolgt der eigentliche Schreibvorgang und das Abspeichern der Paritätswerte. Der Typ 3 hat wegen der kleinen verwendeten Datensegmente dieselben Probleme bei kurzen Datentransfers wie die Typen 0 und 2.

2.1.5 RAID-Level 4

RAID-4 ist RAID-3 sehr ähnlich, jedoch werden wesentlich größere Datensegmente verwendet, so daß Datensätze auch komplett auf einer Festplatte gespeichert werden können (Abb.6). Es wird die Lese-

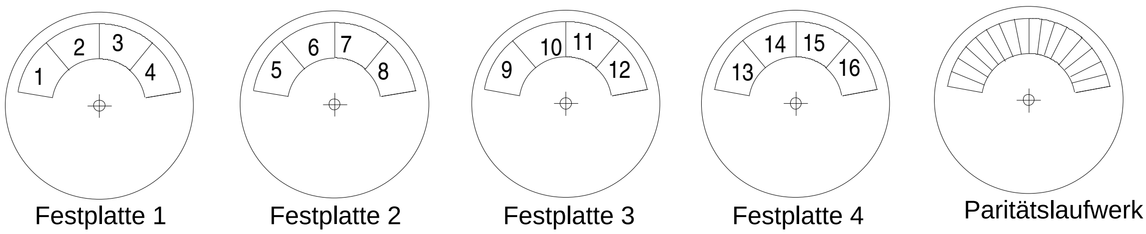

Abb. 6: Schreib- /Lesereihenfolge der Datensegmente bei RAID-4

performanz bei kleinen Datentransfers gegenüber den Typen null, eins und drei stark erhöht, da nicht alle Medien im Verbund an der Leseoperation beteiligt werden müssen. Es genügt zur Paritätsbildung der Lesevorgang von einer Festplatte aus dem Verbund. Die neue Parität wird durch Subtraktion der alten Daten und Addition der neuen Daten zum alten Paritätswert errechnet, ohne Informationen über den Inhalt der anderen Festplatten im Verbund zur Paritätsberechnung der Neuen Daten haben zu müssen. Auch hier können sich Schreiboperationen nicht gegenseitig überlappen, da sie alle die Paritätsplatte beschreiben. RAID-4 hat keine erkennbaren Vorteile gegenüber RAID-5

2.1.6 RAID-Level 5 – Datensegmentierung mit Paritätsbildung

Anders als bei den bisherigen Typen wird bei Typ fünf kein Laufwerk allein zum Speichern der Parität verwendet. Die Datensegmente werden zyklisch geschrieben, wobei die Paritätsinformation ebenfalls zyklisch verteilt wird (Abb. 7). Da sich die Daten und die dazugehörigen Paritätsinformationen auf unterschiedlichen Festplatten befinden, sind die Daten beim Versagen einer beliebigen Festplatte aus den Paritätsinformationen auf den anderen Festplatten im Verbund rekonstruierbar. Da es keine expliziete Paritätsfestplatte gibt, ist auch die damit verbundene Engstelle im Datenstrom vermieden. RAID-5 unterstützt mehrfache, parallele Schreibvorgänge, und ist somit die beste Wahl für Mehrbenutzersysteme.

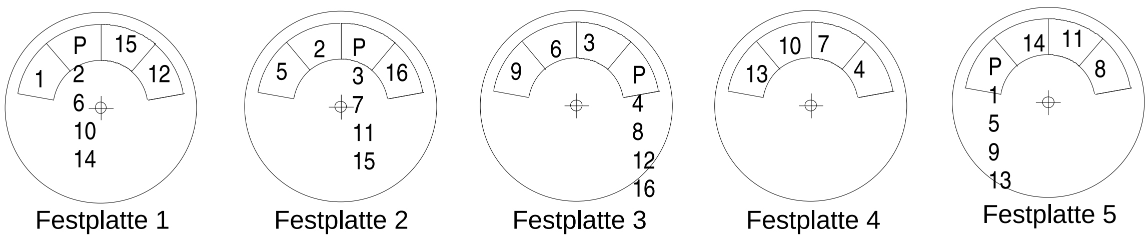

Abb. 7: Schreib-/Lesereihenfolge bei RAID-Typ 5

2.1.7 RAID: Zusammenfassung und Fazit

Der Einsatz von RAID-Systemen bringt mit Ausnahme von RAID-0 den Vorteil kontinuierlich durchgeführter Datensicherung. Anders als bei den klassischen Datensicherungsmethoden (Backup), die in regelmäßigen Abständen zum Teil nur inkrementell durchgeführt werden, ist immer der aktuelle Datenbestand gesichert. Nach dem Ausfall eines Speichermediums muß also nicht von einer je nach

letzten Backupzeitpunkt mehr oder weniger alten Version rekonstruiert werden. Ein weiterer Vorteil liegt in der Tatsache, daß man sogenannte *hotswaps* definieren kann. Dies sind Reserveplatten, die beim Ausfall eines Laufwerks im RAID-Verbund automatisch neu eingebunden werden.
Während die Geschwindigkeitsvorteile durch den Einsatz von RAID-Systemen vom eingesetzten Typ und Anwendungsfall abhängig sind, machen vor allem die Service- und Sicherheitsaspekte der RAID-Typen den Einsatz interessant. Tab. zeigt die Vor- und Nachteile in einer Übersicht.

RAID-Level	Plattenzahl	Hauptmerkmale
0	mind. 2 Stück	+: schnellste Variante **-** : keine Datensicherheit
1	gerade Anzahl	+: schnellster datenredundater Typ **-** : 50% Platzverlust durch Datenredundanz
2	mind. 3 Stück	+: Platzverlust auf 30% reduziert **-** : Extrafestplatte für ECC, schlechte Schreibperformanz
3	mind. 3 Stück	**+:** Platzbedarf für Redundanz auf 10-20% gesenkt **-:** schlechte Schreibperformanz durch aufwenige Paritätsbildung
4	mind. 3 Stück	**+:** größere Datensätze bleiben ungetrennt, daher einfachere Paritätsberechnung bei kurzen Datentransfers
5	mind. 3 Stück	**+:** zyklischer ECC-Schreibvorgang, dadurch Vermeidung des Flaschenhalses der ECC-Festplatte

Tab. 1: Die RAID-Typen in der Übersicht

Im Laufe der Zeit haben sich zahlreiche Varianten gebildet, so daß die genannten Typen nicht alle heute erhältlichen Systeme wiedergeben.

2.2 Die Spiegelung

Unter Spiegeln versteht man das synchrone Schreiben von identischen Datensätzen auf physikalisch getrennte Medien wie z.B. Festplatten. So wird bereits während des Schreibvorgangs eine Kopie des Originals erstellt, bzw. es werden gleichzeitig zwei Originale erstellt. Da kein Fehlerbehebungscode berechnet werden muß, ist das Verfahren relativ schnell. Der Schreibvorgang ist also ebenso schnell wie die Arbeit mit einem einzelnen Speichermedium, da der Datenstrom lediglich geteilt werden muß und keine weiteren Berechnungen stattfinden. Währenddessen wird der Lesevorgang stark beschleunigt, da beide Festplatten identischen Inhalt haben und so wechselweise auf beide Festplatten zugegriffen werden kann.

Hauptnachteil ist die Verdoppelung des Platzbedarfs gegenüber einfachen Speicherprinzipien. Eine Spiegelung der Festplatten ersetzt nicht eine regelmäßige dezentral gehaltene Datensicherung, da die Festplatten z.B. bei einem Brand beide zerstört werden können.

2.3 Redundante Systeme

Redundante, also in ihrer Funktion identische und daher für den eigentlichen Funktionsablauf überflüssige, Systeme sind eine weitere mögliche Schutzmaßnahme. Diese Redundanz kann sich im Inneren eines Computers abspielen, einen ganzen Rechner, eine Datenverbindung oder auch ganze Netzwerke umfassen.

Unabhängig von ihrem Einsatzort und ihrer Komplexität ist Redundanz immer dazu gedacht, ein System auch beim Versagen einzelner Komponenten vor einem Totalausfall zu bewahren. Um dieses zu gewährleisten, muß das redundante System bei Ausfall des Primärsystems dessen Aufgaben in vollem Umfang übernehmen können, ohne daß es zu Auswirkungen auf die Systemumgebung kommt. Die Garantie, daß auch in Ausnahmesituationen der Regelbetrieb unterbrechungsfrei fortgesetzt werden kann ist vor allem bei Systemen mit kritischen Aufgaben sehr wichtig. Als Beispiel sei die Regelung in einem Kernkraftwerk genannt. Aber nicht nur dieses viel zitierte Paradebeispiel für komplexe und sicherheitskritische Regeloperationen ist Anwendungsgebiet von redundanten Systemen. Auch in weitverbreiteten Anwendungsgebieten in der Industrie kann ein Systemausfall schwerwiegende Folgen haben. Wenn z.B. in einem Hochregallager das Rechnerleitsystem ausfällt, können wegen der allgemein verbreiteten chaotischen Lagerhaltung[19] weder Ein- noch Auslagerungen vorgenommen werden, wodurch der ganze Warenumschlag zum Erliegen kommt. Daher ist eine möglichst 100 prozentige Verfügbarkeit des Steuerrechners notwendig.

Es werden aber nicht nur komplette Rechner dupliziert, sondern auch einzelne Elemente. Dies ist z.B. bei einer Variante von RAID-10 (Kap. 2.1.2) der Fall. Bei ihr haben die Festplatten und ihre gespiegelten Kopien eigenen RAID-Kontroller. Damit kann sowohl bei Ausfall einer Festplatte als auch beim Ausfall eines Kontrollers der Rechenbetrieb mit z.T. eingeschränkter Leistungsfähigkeit fortgesetzt werden.

Auch das Internet ist ein Beispiel für Redundanz. In seiner ursprünglichen Bedeutung als atombombensicheres Computernetz machte es sich gerade diese Eigenschaft zunutze. Auch wenn Teile des Datennetzes ausfallen, erfüllt der Rest des weltweiten, engmaschigen Netzes die Aufgabe weiterhin ausreichend.

Man unterscheidet redundante Systeme auch nach ihrer Funktion während des Regelbetriebs.

[19] chaotische Lagerhaltung heißt, daß der optimale Lagerplatz vom Leitrechner bestimmt wird und auch nur diesem bekannt ist.

- Im Beispiel des RAID-Levels 10 hat das redundante System eine die Systemeigenschaft verändernde Funktion, nämlich die Steigerung der Leseperformanz. Es liegt demnach ein Grenzfall zur eigentlichen Redundanz vor.

- Eine weitere Variante ist ein System, welches parallel zum Primärsystem alle Berechnungen und Schaltungen durchführt. Sollte das Primärsystem ausfallen, kann es so alle Aufgaben störungsfrei weiterführen, da es sich im identischen Zustand wie das Primärsystem zum Zeitpunkt des Ausfalls befindet. Das Vorhandensein eines Ersatzsystems ändert jedoch nichts an der Ursache des Versagens des Primärsystems. Liegt die Fehlerursache nicht im Primärsystem sondern in der Systemumgebung, ist die korrekte Weiterführung des Betriebs durch das Ersatzsystem davon abhängig, ob es sich um eine einmalige oder eine andauernde Störquelle handelt.

- Die einfachste Variante ist ein passives System, welches nach einem Ausfall aktiviert wird und erst nach einer Verzögerung die Aufgaben des Primärsystems übernimmt. Bei dieser Variante kann es zu unkontrollierbaren Zuständen im Zeitraum ohne Rechnerkontrolle kommen.

Welche Art der Redundanz gewählt wird und welchen Umfang die gewählte Parallelität aufweist, ist der gegebenen Situation anzupassen.

2.4 Netzwerkfragmentierung

In einem Netzwerk kann es zu einer Reihe von Ausfällen und Problemen kommen. So kann es z.B. zum Ausfall einzelner Rechnerkomponenten, eines Servers oder einer Datenleitung im Netzwerk kommen. Jedem Problem kann durch entsprechende Maßnahmen begegnet werden. Ein dem ersten Anschein nach leicht realisierbares aber kostspieliges Verfahren ist die Redundanz von Komponenten bzw. ganzen Geräten. Wenn ein Gerät ausfällt, wird auf das andere zugegriffen. Dieser Ansatz wurde bereits beschrieben. Neben den genannten Vorteilen wie einer erhöhten Verfügbarkeit, können jedoch auch recht komplex zu behebende Probleme auftauchen.

Fällt ein Speichermedium oder ein Computer aus, kommt es durch den Einsatz redundanter Ersatzsysteme zu keinen schweren Störungen im Netzwerkbetrieb. Was ist jedoch, wenn eine Datenverbindung getrennt wird, und zwar derart, daß es zu einer Fragmentierung des Netzwerkes kommt. Mit diesem Begriff ist gemeint, daß es mehrere autark arbeitende Teilnetze gibt, die mit separaten Datenservern arbeitet (Abb. 8). Nachdem es zum Bruch des Netzes gekommen ist, wird mit getrennter Datenbasis gearbeitet, so daß es bei Mehrfachbearbeitung der verschiedenen Instanzen eines Dokumentes in den Teilnetzen zu Dateninkonsistenzen kommt.

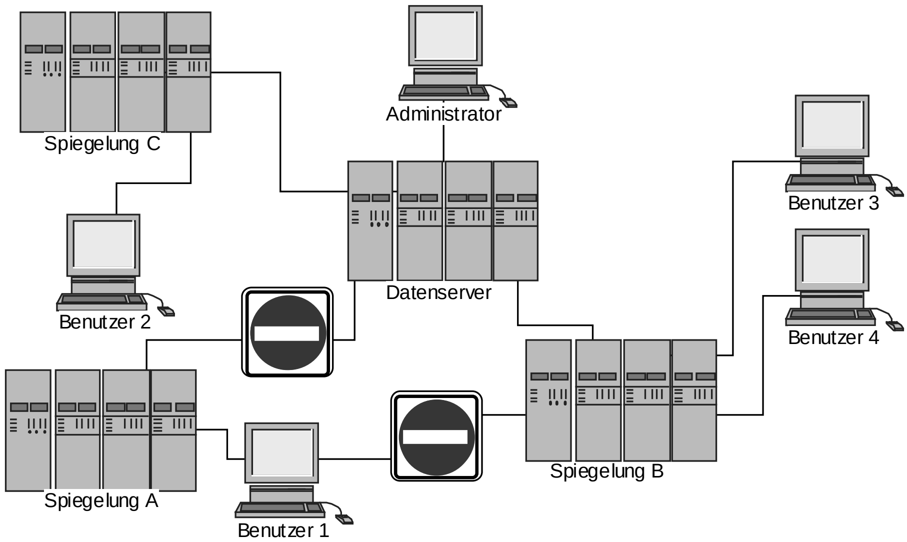

Abb. 8 : Netzwerkfragmentierung mit Abkapselung von Benutzer 1 und der Spiegelung A

Durch die redundante Auslegung der Datenspeicher kommt zu einem Spannungsfeld zwischen den Vor- und Nachteilen.

- Die erhöhte Datensicherheit und Verfügbarkeit der Daten durch Redundanz der Datenserver kann zu Problemen der Datenkonsistenz führen und erfordert demnach Regeln für Ausnahmefälle.
- Die Verringerung des Risikos einer Netzwerkfragmentierung führt zu einer komplexen Netzstruktur mit erhöhten Kosten in Anschaffung und Wartung.

2.4.1 Maßnahmen zur Erhaltung und Wiederherstellung der Datenkonsistenz

Wenn trotz Netzwerkfragmentierung das Entstehen einer Dateninkonsistenz vermieden werden soll, muß eine spezielle und zum Teil inflexible Dokumentnutzungsform gewahrt werden. Durch eine zu dem Dokumententyp passende Replikationsweise kann das Problem der Inkonsistenz weitgehend umgangen werden. Mögliche Realisierungsformen sind [Sen96]:

- **Reine Lesereplikation**

  Vom Originaldokument gibt es nur Lesekopien auf den verschiedenen Datenservern. Schreiboperationen sind nur durch Dokumentbesitzer möglich. Inhaltsänderungen werden nur durch ihn vorgenommen und er ist für die Datenkonsistenz aller Kopien zuständig. Es muß demnach nur gewährleistet werden das diese Änderungsmeldungen auch alle Datenserver erreichen. Da eine Änderung der replizierten Daten nicht möglich ist, ist dieses Replikationsprinzip nur für bestimmte Dokumenttypen sinnvoll, wie z.B. Vorlagen und Mitteilungen.

- **Bedarfsweise Replikation**

  Bei diesem Verfahren wird beim Zugriff auf die Datei das Vorhandensein einer lokalen Kopie überprüft und gegebenenfalls wir eine Replica angelegt. Diese Kopie dient zur Erhöhung der Leseperformanz, da Schreiboperationen aus Konsistenzgründen ein Verschieben der Originaldatei zu Benutzer erfordern. Nach dem Schreibvorgang wird die Datei ebenso automatisch wieder zurückkopiert. Anschließend wird eine systemweite Änderungsmeldung generiert, welche alle veralterten Replicas löscht. Ist eine Replica aktiv, wird es als ungültig markiert und nach Beendigung gelöscht.

  Im Falle einer Netzwerkfragmentierung kann es trotzdem zu Inkonsistenzen kommen, falls eine Replica ihre Löschanweisung nicht erhalten hat. Dieses Risiko wird zugunsten des einfachen Protokolls eingegangen. Solange man sich dieser Schwäche bewußt ist und die Risiken richtig einschätzt, sind die auftauchende Probleme behebbar.

- **Ändern aller Replikas**

  Anders als bei den bisher genannten Verfahren gibt es kein explizietes Original und dessen Kopien. Alle Dateien sind gleichwertig. Soll die Datei beschrieben werden, muß dieser Vorgang auf alle Datenversionen gleichzeitig durchgeführt werden. Es bedarf einer Steuerung der Schreibrechtvergabe an die Benutzer. Im Falle einer Netzwerkfragmentierung ist ein synchroner Schreibvorgang auf alle Dokumentinstanzen nicht möglich, womit eine Änderung des Dokumentes insgesamt unzulässig wird.

Die hier genannten Verfahren sind die einfachsten denkbaren Varianten und sollen lediglich die Problematik der Datenkonsistenz bei Netzwerkfragmentierung verdeutlichen.

3 Datenschutz auf der physikalischen Schicht

Während es bisher darum ging, Datenverluste durch Fehlverhalten von Technik und Personen zu bekämpfen, geht es jetzt um das Verhindern von bewußt herbeigeführter Datenmanipulationen und von Datendiebstahl. Dies impliziert, daß eine dritte Person bewußt in das System eindringen und nicht genehmigte Aktionen ausführen will. Zu schützende Angriffspunkte im System sind dabei unter anderem:

1. Der Einbruch über ein Datenterminal im Systemnetz
2. Der Zugang über das Computernetz von außerhalb
3. Das Abhören von Daten ohne in das Computernetz einzudringen.

Der Netzzugang wird i.d.R. durch Passwörter gesichert. Zusätzliche Legitimationsnachweise sind möglich. Als Beispiel wird in Kapitel 3.2 das sogenannte SMART CARD SYSTEM betrachtet.

Um das System vor Angriffen von außerhalb, also über das Internet oder per LAN/WAN zu schützen, werden Mechanismen wie Firewalls, das Routing bzw. Gateways eingesetzt. Dieses Thema wird durch andere Seminarbeiträge behandelt. Punkt drei ist Thema von Kapitel 3.3 Dieses Problemgebiet wird heute leider noch weitgehend außer acht gelassen – sehr zur Freude aller Datendiebe.

3.1 Verschlüsselung auf der Hardwareebene

Verschlüsselungsalgorithmen können entweder software- oder hardwaregestützt realisiert werden. Beide Varianten haben sowohl Vor- und Nachteile, auf welche später näher bezug genommen wird. In der Anwendung sind derzeit vor allem Softwarelösungen. Bei hardwarebasierten Realisierungen gibt es die Option, eine Verschlüsselung mit einer Zugangskontrolle durch den Einsatz von SmartCard-ähnlichen Trägermedien zu kombinieren (Kap. 3.2).

3.1.1 Softwarebasierte Verschlüsselung

Wird Verschlüsselung softwaremäßig betrieben, kann man ohne großen Aufwand ein bestehendes System nachrüsten. Ein Algorithmus kann jederzeit aktualisiert werden. Wenn sich herausstellen sollte, daß ein Verschlüsselungsalgorithmus nicht genügend Sicherheit gegen den unbefugten Zugang bietet, kann die verwendete Software gegen ein Konkurrenzprodukt mit aktuelleren Algorihmen ausgetauscht werden oder durch eine neuere Version des Programmes ersetzt werden.

3.1.2 Hardwarebasierte Verschlüsselung

Ein Hauptnachteil der softwarebasierten Algorithmen ist die zusätzliche Hauptprozessorbelastung durch die durchzuführenden Verschlüsselungen. Je sicherer ein Verschlüsselungsalgorithmus ist, desto komplexer und rechenintensiver ist er. Der Prozessor muß also alle Berechnungen vornehmen, sowohl den Dekodiervorgang beim Lesen als auch das Kodieren beim Schreibvorgang. Da hier die Sicherheit durch Geschwindkeitsverlust bezahlt wird, kann es sinnvoll sein, über die Einführung hardwarebasierter Verschlüsselung nachzudenken.

Sinnvolle Stellen im System, an denen der Einbau eines solchen Bauteils denkbar ist, sind z.B. der SCSI-Kontroller oder der RAID-Kontroller. Zum einen sind sie der Knotenpunkt aller Datenströme im Rechner zwischen Massen- und Hauptspeicher, zum anderen haben sie bereits einem eigenen Prozessor. Es wäre demnach möglich sowohl einen eigenen Kodierprozessor auf der Kontrollerplatine zu plazieren, als auch über eine hochintegrierte Lösung durch die Kombination aller Funktionen auf einem Prozessor nachzudenken – alles eine Frage der produzierten Stückzahlen.

3.2 Zugangangskontrollen

Bekanntestes Stichwort bei den Zugriffskontrollen sind die SMART CARDS [URL-4,URL-5]. Mit ihnen können verschieden Anwendungsbereiche abgedeckt werden. Anwendungsgebiete sind z.B.

- Identifikation und Authentisierung
- Speicherung sensibler Informationen

Die Identifikation geht hier nach dem Prinzip „Wissen und Besitz“ vor sich. Zusätzlich zu dem Wissen eines bestimmtes Passwortes kommt noch die Notwendigkeit des SmartCard-Besitzes. Dabei ist problematisch, daß die Sicherheit der SmartCards immer fragwürdiger wird. Immer häufiger ist zu lesen, daß der Inhalt einer SmartCard mit dem entsprechend Know-How und einem handelsüblichen PCMCIA- Lesegerät, auslesbar und veränderbar ist. Das Frankfurter Landgericht (WAZ) hat den Verkauf und Besitz einer durch die Firma S.A.D. vertriebene Software namens „Cards“ verboten. Eine Manipulation von Scheck und Kreditkarten ist mit dieser und ähnlichen Softwareprodukten möglich. Die Sicherheit der Daten muß kritisch betrachtet werden. Eine verlorene SmartCard ist in etwa so gefährlich wie ein Zettel mit dem Passwort an einer Pinwand.

Der Punkt „Speicherung sensibler Daten“ muß demzufolge ebenso kritisch betrachtet werden. Der Vorsatz, wichtige Daten für dritte unzugänglich aufzubewahren ist zweifelsfrei richtig, ob jedoch eine SmartCard sicherer ist als andere verschlüsselte Medien, sei dahingestellt. Der „normale“ Computerbenutzer im Betrieb kann mit einer SmartCard wohl wenig anfangen, wenn er nicht im Besitz des passenden Berechtigungscodes ist. Gegen Sicherheitsverstöße durch ihn würden aber auch andere Zugangskontrollen ausreichen.

Bei all der Kritik an der Sicherheit muß aber trotzdem anerkannt werden, daß die SmartCard in der Praxis eine immer bedeutender werdende Stellung einnimmt. Durch ihre Kompatibilität mit PCMCIA-Standard ist eine hohe Verbreitung von Lesegeräten gegeben. Alleine 1995 wurden über zehn Millionen PCMCIA-Lesegeräte verkauft. Bekanntermaßen ist PCMCIA ein Standard im Notebookbereich, wodurch externe Lesegeräte für die Zugangskontrolle nicht notwendig sind.

Trotz ihrer hohen Verbreitung gibt es Konkurrenzprodukte, welche z.B. als Stecker für die parallele Schnittstelle konzipiert sind. Als Beispiel diene ein System von AZ-Tech [URL-3] In diesen Stecker können bis zu zwei Schlüssel gesteckt werden. Diese beinhalten ein 64bit Passwort und werden durch eine 48bit Seriennummer Unikate. Es gibt Versionen mit zeitlich beschränkter Zugriffsgenehmigung zum ganzen System oder zu bestimmten Programmen. Außerdem kann diese Zeitbeschränkung der Schlüssel im Rahmen einer Online-Lizensierung verlängert werden.

Nachteil dieser Variante gegenüber der SmartCard ist der ungünstige Ort der parallelen Schnittstelle an Rechnerrückseite. Als Vorteil ist zu sehen, daß diese Format nicht so verbreitet ist, so daß Datenmanipulationen erschwert sind, da es noch keine fertig erwerbbare Software gibt.

3.3 Abhörsicherheit

Beim Thema Abhörsicherheit soll der physikalische Faktor im Vordergrund stehen. Es geht nicht darum, ob eine Datenpaket gut genug codiert wurde, um einen sicheren Transport von A nach B zu

gewährleisten. In diesem Kapitel soll es um das Abhören eines Computers bzw. dessen Datenströme mittels Auswertung von Streustrahlung – der sogenannten kompromittierenden Strahlung gehen.
Dabei handelt es sich um eine erstmals im April 1984 vom DDR-Geheimdienst durchgeführte elektronische Version des Lauschangriffs. Die Hauptgefahr liegt hierbei darin, daß es keinen physikalischen Kontakt mit dem Computer oder Datennetz gibt. Da es keine Datenmanipulationen gibt, kann auch kein illegaler Zugriff auf die Netzsicherheit festgestellt werden.
Trotz dieses alarmierenden Umstandes ist die Industrie wenig problembewußt. Längst wird diese Technologie nicht nur von Geheimdiensten in aller Welt genutzt, sondern insbesondere zur Wirtschaftsspionage eingesetzt. Da Rechner mit Inhalt, welche die „nationale Sicherheit" betreffen, längst tempest-konform[20], also abhörsicher sind, und auch die Rechnersystem der Banken größtenteils abgesichert sind, richtet sich das Interesse der Abhöraktionen heute gezielt auf Ziele in der Industrie und Wirtschaft.

> „Wir haben in einem Versuch einen einfachen tragbaren Schwarzweißfernseher für 200 Mark mit bescheidener Technik erweitert und zum Abhörgerät umfunktioniert. Dann kann man aus einiger Entfernung mitverfolgen, was auf einem PC-Bildschirm abläuft [...]" ([URL-1]).

In Europa gibt es nur zwei Hersteller tempest-konformer Computersysteme. Siemens stellt solche Systeme her, produziert jedoch nur geringe Stückzahlen, da „...auch für geringer abgeschirmte Systeme, abgesehen von Banken und Versicherungen, praktisch keine Nachfrage besteht", so Walter Kräutlein, verantwortlicher Produktmanager bei Siemens. Da die große Mehrheit der allgemein eingesetzten Rechnersysteme demzufolge keinerlei Schutz vor der Verbreitung von kompromittiernde Strahlung bieten, ist vor allem die Möglichkeit einer Nachrüstung von Interesse

3.3.1 Was kann abgehört werden

Prof. Dipl.-Ing. Erhard Möller bezieht sich in seiner Veröffentlichung nur sogenannte Computerstrahlung [URL-2]. Dies ist nicht die einzige vorkommende Art kompromittierder Strahlung, jedoch ist es vor allem diese Form, die uns im weiteren interessiert. An folgenden Stellen entstehen kompromittiernde Strahlung

1. Netzkabel
2. Hochfrequenz-Coax-Datenkabel
3. ein- und auslaufende Kühl- und Serviceleitungen der Zimmer.

Man muß demnach nicht nur die Computersysteme untersuchen, um die Entstehung und Verbreitung von kompromittierender Strahlung zu verhindern, sondern auch die Räumlichkeiten in denen die Rechneranlagen untergebracht werden.
Weiter kann die Art der kompromittiernden Strahlung unterschieden werden:

---

[20] engl. tempest: Sturm

1. elektromagnetische Strahlung in Form elektrischer und magnetischer Wellen
2. elektromagnetische Wellen auf der Oberfläche aller coaxialen metallenen Verbindungen
3. elektrische Interferenzerscheinungen in den Stromversorgungsleitungen des Systems.

Da jede dieser oben genannten Formen kompromittiernder Strahlungen in die jeweils anderen überführt werden kann, muß man sich auch mit allen drei Arten auseinandersetzen um einen ausreichenden Schutz vor der Verbreitung kompromittiernder Strahlung zu haben.
In einem Versuchsaufbau hat Prof. Möller die Stromkabel eines Computers mit Absorbern umgeben, so daß keine induzierten Wellen durch die Stromkabel abgegeben werden konnten. Trotzdem konnte er zehn Meter vom Videoterminal auf einem Fersehbildschirm mittels dessen Dipolantenne das decodierte Videosiganl des Computers anzeigen. Hieraus wurden folgende Schlüsse gezogen:

- Der Monitor sendet elektromagnetische Wellen aus, die
- über eine gewisse Distanz empfangen und dekodiert werden können, womit
- diese Strahlung kompromittierend ist.

Durch Abwandlungen im Versuchsaufbau können gezielt die einzelnen Strahlungsarten abgehört werden. Die allgemein bekannteste und oft genutzte Art kompromittiernder Strahlung ist das sogenannte baby-phone. Hierbei handelt es sich um eine bewußt in die Stromversorgung aufmodulierte Welle, die eine Sprachverbindung durch das Stromnetz ermöglicht. Auf die gleiche Weise kann eine nicht gewollte Datenaufmodulierung durch den Computer auf größere Distanz durch entsprechende Geräte lesbar gemacht werden.
Aus größerer Distanz ist wörtlich gemeint. Je nach Art der kompromittiernden Strahlung, ist diese in einer Entfernung von 25 bis 100 Meter empfang- und lesbar. Sogar in 150 Meter Entfernung sind, je nach Bauart des Bürogebäudes, die kompromittiernden Strahlungen noch klar empfangbar. In diesem Radius ist ein unauffälliges Abhören in einem angemieteten Büro, einem geparkten Auto o.ä. kein Problem.
Es gibt sogar fertige Geräte im Handel, um sich aus dieser Fülle von Daten zu bedienen. So bietet die Firma CCS aus New York ein Gerät namens „Computer Intercept System STG 4625“ an. Im Datenblatt verspricht das Gerät, daß es die Abstrahlungen „zu einer scharfen Reproduktion der abgefangenen Daten verarbeitet“ [URL-1]. Auch können Geräte, die eigentlich als Schutz gegen den widerrechtlichen Abhörvorgang entworfen wurden, zweckentfremdet werden. Ist mit Hilfe dieser Geräte das Vorhandensein kompromittierender Strahlungen nachgewiesen, ist es nur noch ein kleiner Schritt bis zum eigentlichen Abhören und Dekodieren.

3.3.2 Was dagegen getan werden kann

Wie bereits gesagt, kann jede Art kompromittiernder Strahlung in jede beliebige andere überführt werden, wonach eine Schutzmaßnahme nur dann sinnvoll sein kann, wenn auch jede Art bekämpft wird. Die Schutzmaßnahmen sind nach drei Gesichtspunkten unterteilbar. Jeder Maßnahmentyp hat dabei seine Stärken und Schwächen. Die Maßnahmenarten sind

- konstruktive Änderung des Objektes, welches kompromittiernde Strahlen aussendet
- Überlagerung der kompromittiernden Strahlung durch weißes Rauschen oder Störsignale
- Abschirmung und Filterung

3.3.2.1 Konstruktive Änderungen

Konstruktive Änderung bedeutet, daß die signalerzeugenden Prozesse geändert werden müßten. Dies hat so zu geschehen, daß das Spektrum und die Impulsamplitude derart geändert werden, daß ein Empfang durch einen (modifizierten) Fernseher nicht länger möglich ist. Wenn das Bildsignal geändert würde, also z.B. die Pixelfrequenz, Impulsamplitude o.ä., hat dies zwar zur Folge, daß der Abhörvorgang schwieriger wird, jedoch ergeben sich auch einige gravierende Nachteile.

So hat der Benutzer i.d.R. nicht die Fähigkeiten und technischen Möglichkeiten zur Modifizierung des Gerätes. Außerdem ist mit diesem Vorgang automatisch der Garantieanspruch erloschen. Obendrein geht die Betriebsgenehmigung verloren. Eine Änderung ab Werk schließt sich ebenfalls aus, da damit ein neuer Standard geschaffen würde, der auch den Lauschern schnell bekannt würde.

Die Zeitspanne, in welcher eine erhöhte Sicherheit gegeben wäre, ist also nicht abschätzbar.

3.3.2.2 Überlagerung des Signals

Ist die Verhinderung der Signalentstehung nicht praktikabel, kann der Inhalt der kompromittiernden Strahlung durch den Einsatz externer Geräte so verfälscht werden, daß eine Rekonstruktion des ursprünglichen Dateninhalts unmöglich wird. Durch ein gezielt überlagertes elektromagnetisches Signal mit höheren Impulsamplituden oder ein überlagerndes elektromagnetisches Rauschen, wird zwar nicht verhindert, daß die kompromittiernde Strahlung Daten in die Außenwelt getragen wird, jedoch ist deren Ausfilterung aus dem begleitenden Datenmüll durch die oben beschriebene Testanordnung mit einer „leicht modifiziertem TV-Ausrüstung“ nicht mehr möglich.

3.3.2.3 Abschirmung und Filterung

Schließlich ist noch die Möglichkeit der Abschirmung der kompromittiernden Strahlung gegeben. Es muß verhindert werden, daß eine der drei möglichen kompromittiernden Strahlungsarten das Gebäude bzw. das Zimmer verlassen kann.

Zwar ist eine Abschirmung relativ schnell nachträglich möglich, jedoch gibt es einige Punkte, die beachtet werden müssen.

- **Maßnahmen am Gerät**

  Vernachlässigbar ist in diesem Zusammenhang das hochwahrscheinliche stylistische Desaster. Abschirmung ist i.d.R. nicht von hohem designerischem Wert, sondern durch Funktionalität geprägt.

  Zu Beachten ist, daß eine vollständige Abschirmung des Gerätes aus betriebstechnischen Gründen oft nicht möglich ist. So müssen z.B. die Lüftung und Bedienelemente nach wie vor ihre Funktionalität beibehalten, so daß in der Abschirmung Lücken mit hohem Strahlungsausstoß entstehen können. Diese Lücken müssen gesondert abgesichert werden. So existieren z.B. spezielle Glasssorten, die durch ihren hohen Metallanteil eine ausreichend hohe Abschirmwirkung erzielen. Die Lüftungen können durch spezielle Metallgitter geschützt werden.

  Zusätzlich müssen alle Stromleitungen mit Filtern versehen werden, da, wie bereits erwähnt, auch die Stromkabel ein Medium zur Verbreitung von kompromittiernder Strahlung sind.

- **Maßnahmen am Raum oder Gebäude**

  Abschirmmaßnahmen am Gebäude haben mehrere große Vorteile. So liegt die Durchführung alleine bei der Firma. Dies ist von Vorteil, da der Gerätehersteller keine funktionsspezifischen Einschränkungen vorschreiben kann. Demzufolge sind die Abschirmungsmaßnahmen vom Gerätetyp unabhängig und können auch bei einem Systemwechsel weiterverwendet werden. Es entstehen höchstens minimale Folgekosten und die Abschirmmaßnahmen sind optisch besser ins Raumbild integrierbar als Maßnahmen man Gerät direkt.

  Lediglich wenn bauliche Änderungen an der Gebäudeaußenseite vorgenommen oder die statischen Eigenschaften des Bauobjekts geändert werden, bedarf es einer Genehmigung durch die entsprechenden Behörden.

Welche Maßnahmen im einzelnen zu treffen sind, um einen ausreichenden Schutz gegen Abhörversuche der kompromittiernden Strahlung zu erlangen, sei hier nicht im Detail diskutiert. Es sei lediglich gesagt, daß eine Wandverkleidung aus Stahl mit einer Stärke von drei bis fünf mm, Spezialglas in den Scheiben (ausreichende Gitter hätten eine zu geringe Lichtdurchlässigkeit) und Filtermaßnahmen an jeder Leitung, die den Raum verläßt, benötigt werden.

4 Fazit

Der Bereich der Datensicherheit und des Datenschutzes wird auch weiterhin eine hohe Beachtung erfordern. Es gibt immer ausgefeiltere Angriffsstrategien und immer leistungsfähigere Rechner bzw. Algorithmen, um bestehende Schutzmaßnahmen zu knacken oder zu umgehen. Schutzmaßnahmen müssen, wenn sie wirklich wirksam sein sollen, auf alle möglichen Angriffsarten vorbereitet sein – auch auf Abhörversuche der kompromittierenden Strahlungen

Schlecht geschützte Systeme sind durch die immer größer werdende, weltweite Verknüpfung via Internet, heute noch leichter angreifbar geworden.
Um diesen erhöhten Sicherheitsanforderungen genügen zu können, ist es durchaus sinnvoll, bereits mit Maßnahmen auf der physikalischen Schicht zu beginnen. Wie gezeigt, ist es sogar notwendig auf physikalische Angriffe mit physikalischen Mitteln zu reagieren, da kompromittiernde Strahlungen Daten übertragen die nicht verschlüsselbar sind (Ein- und Ausgabedaten). Anmerkung: Bevor ich diesen Beitrag verfaßt habe, war ich mir der Problematik der kompromittiernden Strahlung nicht bewußt und unterstelle, daß es einem großen Teil angehender Informatiker nicht anders geht. Deshalb finde ich es wichtig, auf diese Gefahr hinzuweisen.

5 Literatur- und Quellenverzeichnis

5.1 Printmedien

[Mar96] Peter Marwedel, *Rechnerarchitektur*, Vorlesungsskript an der Universität Dortmund im Fachbereich Informatik, 1996

[Sen96] Marco Sensken, *Konzepte für die Multikopienhaltung in verteilten Systemen*, Diplomarbeit an der Universität Dortmund am LS1 des Fachbereichs Informatik, 1996

[WAZ] WAZ vom 5.2.98, *Software ermöglicht Kreditkartenbetrug*, Aktenzeichen 3-12 0 207/97 am Frankfurter Landgericht

5.2 Elektronische Dokumente

[URL-1] `http://www.industriemagazin.co.at/heft9703/lauschangriff.html`

[URL-2] `http://www.fc.net/phrack/files/p44-10.html`

[URL-3] `http://www.az-tech.com/pr_ke.html`

[URL-4] `http://www.utimaco.de/utimacode.nsf`

[URL-5] `http://www.scmmicro.com.white.htm`

Sicherheit auf der Netzwerkschicht

Olaf Strozyk

1 Einleitung

Das Internet erlebt in den letzten Jahren einen großen Boom. Immer mehr Menschen nutzen – sowohl privat als auch geschäftlich – immer mehr Dienste. Unternehmen verbinden ihr firmeninternes Netz mit dem Internet, um selbst ihrer Klientel Dienste anzubieten oder Dienste anderer Firmen oder Organisationen in Anspruch nehmen zu können. Firmen mit mehreren Standorten verbinden ihre Intranets oft aus Kostengründen über das Internet miteinander.

Anders als in den Anfängen des globalen Netzwerks, als in erster Linie Universitäten das Internet zum Austausch von Informationen nutzten, handelt es sich in zunehmendem Maße um vertrauliche Daten, denen das Internet als Medium dient. Deshalb wird die Frage nach der *Sicherheit* der Daten während der Übertragung immer bedeutender.

Der Begriff Sicherheit meint hier *Authentizität* (Ist es wirklich der gewollte Kommunikationspartner, mit dem kommuniziert wird?), *Integrität* (Ist die Information unverändert übertragen worden?), *Vertraulichkeit* (Kann ein Dritter ungewollt in den Besitz der Information gelangen?) und *Nicht-Zurückweisbarkeit* (Kann der Empfänger später beweisen, daß der Sender eine bestimmte Information übermittelt hat?). Manchmal kann es für Kommunikationspartner bereits verhängnisvoll sein, wenn ein Dritter nur Kenntnis davon hat, ob bzw. wann miteinander kommuniziert wurde (*Verkehrsflußanalyse*).

Es gibt viele Bemühungen, in Programme auf der Anwendungsschicht Sicherheitsmechanismen zu integrieren, was zu Lösungen wie verschlüsselter *e-mail*[21] (siehe [Bohn98]) oder *S-HTTP*[22] (siehe [Mart98]) führt. Es ist jedoch immer noch üblich, telnet- oder ftp-Verbindungen unverschlüsselt aufzubauen, obwohl hierbei leicht ein Angreifer, der Zugang zu einem der Netzwerke zwischen Sender und Empfänger hat, das Passwort ausspähen und den Datenverkehr mitprotokollieren kann.

Diese Probleme kann man lösen, indem man auch bei diesen Diensten zu Client- und Server-Software wechselt, die den gesamten Datenverkehr verschlüsselt und so für Sicherheit sorgt. Eigentlich liegt es jedoch näher, nicht für jeden Dienst (neue) Sicherheitsprotokolle zu entwerfen, sondern bereits auf der diesen Diensten zugrundeliegenden Schicht, nämlich der Netzwerkschicht, Mechanismen zu integrieren, die Sicherheit im oben genannten Sinne bieten können.

---

[21] electronic mail, engl. für (durch ein Netzwerk versandte) elektronische Post.
[22] Akronym für secure hypertext transfer protocol, engl. für sicheres Hypertext-Übertragungsprotokoll.

Diese Seminararbeit stellt am Beispiel der Version 6 des Internet Protokolls (IPv6, manchmal auch als *IP next generation*[23], *IPng* bezeichnet) Maßnahmen zur Gewährleistung von Sicherheit auf der Netzwerkschicht vor.

Dazu werden zuerst die Neuerungen von IPv6 gegenüber der sich zur Zeit im Einsatz befindlichen Version 4 des Internet Protokoll (IPv4, siehe hierzu [Schr98]) erläutert. In Kapitel 3 werden dann die Sicherheitsmechanismen vorgestellt, die Authentizität, Integrität, Vertraulichkeit und Nicht-Zurückweisbarkeit bieten können. Die hier vorgestellten Mechanismen bieten keinen Schutz vor Verkehrsflußanalysen. Es gibt allerdings zahlreiche Techniken, die als Schutz vor solchen Angriffen verwendet werden können (siehe [VoKe83]), auf die hier jedoch nicht eingegangen wird.

Es sei angemerkt, daß die unten beschriebenen Sicherheitsmethoden auch in IPv4 implementiert werden können. Kaum ein sich im Einsatz befindlicher Protokoll-Stapel implementiert diese neuen Mechanismen jedoch bereits zum gegenwärtigen Zeitpunkt. Dagegen ist deren Implementierung (zumindest in Grundversionen, s.u.) in jeden IPv6-Protokoll-Stapel vorgeschrieben.

Zum Schluß wird in Kapitel 4 hinterfragt, ob und wie angreifbar diese Sicherheitsmethoden sind.

2 Internet Protocol Version 6

Das Internet Protokoll (IP) Version 6 wurde als Nachfolger für IP Version 4 entwickelt. Ein Ziel dabei war, in der Übergangszeit beide Protokolle nebeneinander benutzen zu können, da nicht zu einem bestimmten Zeitpunkt alle Rechner gleichzeitig umgestellt werden können. Deshalb kann die Umstellung schrittweise vorgenommen werden, ohne das ganze Netz in seiner Funktion zu beeinträchtigen.

2.1 Motivation

Das dringendste Problem der Version 4 des Internet Protokolls sind die langsam zur Neige gehenden freien Rechner-Adressen. IPv4 bietet zwar theoretisch mehr als 4 Milliarden Adressen; in der Praxis bleiben jedoch viele ungenutzt, da sie immer netzweise (Class C: 254, Class B: rund 65000, Class A: 16,6 Mio) vergeben werden (siehe [Schr98]). Außerdem reservieren sich viele Firmen in letzter Zeit mehr Adressen als sie eigentlich benötigen, da heute jedem bewußt ist, daß der Adreßpool begrenzt ist (siehe [Hose96]).

Ein weiterer Schwachpunkt bei IPv4 ist der relativ hohe Aufwand, der beim *Routing*[24] anfällt. Zum einen resultiert dieser direkt aus dem wenig hierarchischen Aufbau der Adressen, zum anderen werden durch Multimedia-Anwendungen heute ganz andere Anforderungen an das Netzwerk gestellt, als es bei der Entwicklung der Version 4 absehbar war.

[23] engl. für Internet-Protokoll der nächsten Generation.

[24] engl. für Finden eines (kurzen, bzw. performanten) Weges innerhalb eines Netzwerkes; oft bezogen auf einen einzelnes System (dann: „*router*“), das ein Paket weiterleiten muß.

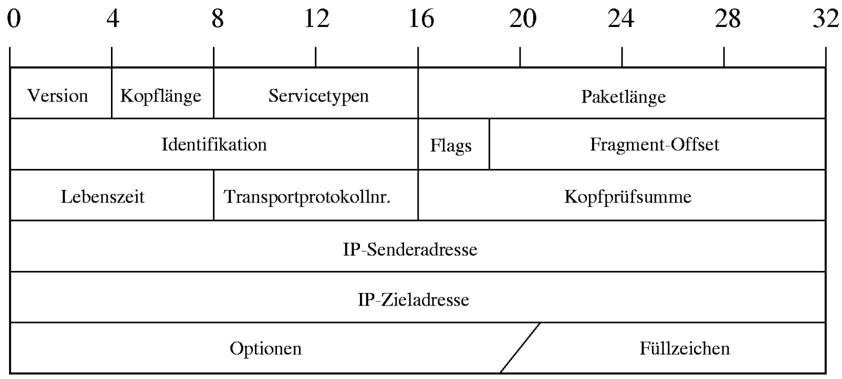

Abb. 1: Der IPv4 Basiskopf (nach [Hose96])

Multicast[25]-Techniken, die u.a. für Audio- oder Videoübertragungen sinnvoll sein können, waren in der Version 4 beispielsweise nicht vorgesehen. Es gibt zwar mittlerweile Auswege aus diesem Dilemma, um auch unter IPv4 *Multicast*-Techniken benutzen zu können, so z.B. den *Multicast Backbone*[26] (MBone), mit dem unter IPv4 erste Versuche laufen. Es ist jedoch sinnvoller, solche Erweiterungen direkt in die IP-Schicht zu integrieren.

In der Spezifikation des IPv4 waren außerdem keine Sicherheitsoptionen vorgesehen. Durch zwei neue *Header*[27]-Typen werden bei IPv6 nun bereits auf der Netzwerkschicht Sicherheitsmechanismen definiert. Während jede Implementierung von IPv6 diese unterstützen muß, können auch IPv4-Implementierungen entsprechend erweitert werden.

2.2 Unterschiede zwischen IPv6 und IPv4

Die Veränderungen sind in erster Linie (vgl. [DeHi95]):

- vergrößerte Adressierungsmöglichkeiten (siehe Kapitel 2.1)
  IPv6 erhöht den Adressraum von 32 auf 128 Bits, um eine Adressierungshierarchie mit mehreren Ebenen, eine viel größere Anzahl an Netzknoten und eine einfachere Selbstkonfiguration zu ermöglichen. Außerdem ermöglicht es neben der *Unicast*[28]- auch eine *Multicast*- und eine *Anycast*[29]-Adressierung.

---

[25] engl. für Senden des selben Datenpakets an eine Gruppe von Zielrechnern, wobei das ausgehenden Paket seitens des Senders nicht mehrfach übertragen werden muß.
[26] engl. für zur Übertragung mittels *Multicast*-Technik geeigneter Teil des Netzwerks (*backbone*: engl. für Rückgrat).
[27] engl. für Protokollkopf eines Datagramms; also eine Dateneinheit vor den Nutzdaten, die Informationen darüber enthält, was mit den Nutzdaten passieren soll.
[28] engl. für Senden eines Datenpakets an genau einen Zielrechner.
[29] engl. für Senden eines Datenpakets an genau einen (den schnellsten) Rechner, aus einer bestimmten Gruppe von Rechnern.

- Vereinfachung des Protokollkopfes (siehe Kapitel 2.2)
  Einige *IP-Header*-Felder wurden verworfen oder als optional erklärt, um die Bearbeitungskosten für gewöhnliche Pakete zu reduzieren und die benötigte Bandbreite für den Kopf einzuschränken.
- bessere Unterstützung für Erweiterungen und weitere Optionen
  Zu einem späteren Zeitpunkt können bei Bedarf weitere Optionen für den Options-Erweiterungskopf (siehe 2.2.2) oder ganz neue Erweiterungsköpfe (siehe 2.2.2) spezifiziert werden, ohne das Protokoll grundlegend verändern zu müssen.
- Einführung der Möglichkeit von Datenstromkennzeichen (engl.: „*Flow Label*")
  Dies erlaubt das Kennzeichnen von Paketen, die zu einem bestimmten Datenfluß gehören, für den der Sender besondere Bearbeitung anfordert, wie besondere Übertragungsqualitäten oder „Echtzeit"-Dienst.
- Authentifizierungs- und Vertraulichkeitsmerkmale (siehe Kapitel 3)
  In IPv6 sind Erweiterungen spezifiziert, die Authentifizierung, Integrität und Vertraulichkeit ermöglichen.

Die Sicherheitsoptionen, die das eigentliche Thema dieser Arbeit darstellen, sollen in Kapitel 3 vorgestellt werden. In den folgenden zwei Kapiteln werden weitere bedeutende Neuerungen genauer beschrieben.

2.2.1 Vergrößerte Adressierungsmöglichkeiten

Während bei Version 4 nur 32 Bit zur eindeutigen Identifikation eines Netzknotens benutzt werden, sind es bei IPv6 128 Bit. Es stehen also bei Vernachlässigung von Verwaltungsinformationen 2 hoch 128 Adressen zur Verfügung. Laut [Hose96] hat C. Huitema berechnet, daß selbst bei sehr uneffizienter Vergabe jedem Quadratmeter der Erdoberfläche mehr als 1500 Adressen zugewiesen werden könnten.

Bei IPv4 werden die IP-Adressen üblicherweise durch vier Dezimalzahlen dargestellt, die durch Punkte getrennt werden. IPv6-Adressen schreibt man als acht vierstellige Blöcke von Hexadezimalzahlen, die durch Doppelpunkte getrennt werden. Um die Schreibweise der langen Adressen zu vereinfachen, darf folgendermaßen gekürzt werden: Führende Nullen innerhalb eines Blocks dürfen weggelassen werden; falls ein ganzer Block aus Nullen besteht, bleiben also nur die benachbarten Doppelpunkte übrig. Ein Prefix aus mehreren Doppelpunkten darf nun noch durch zwei Doppelpunkte ersetzt werden.

Die 128 Bit einer *Provider*[30]-basierten *Unicast*-IPv6-Adresse läßt sich in 6 Bereiche unterteilen:

- Die ersten 3 Bits geben ihren Typ an und sind immer 010.

[30] engl. für Lieferant; hier: Institution, die anderen den Zugang zum Internet ermöglicht.

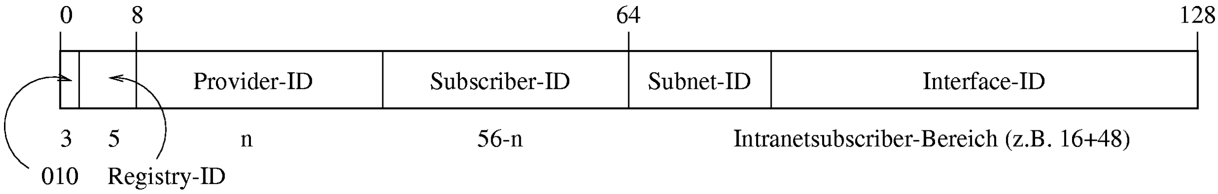

Abb. 2: *Unicast*-Adressen bei IPv6 (nach [Hose96])

- Die nächsten 5 Bits stellen die *Registry*[31]-ID dar und stehen für die Organisation, die die Adresse vergeben hat.
- Danach folgt eine variable Anzahl von (n) Bits, die den *Provider* kennzeichnen, der für alle Adressen mit diesem Präfix zuständig ist.
- Dem Provider-Feld folgt bis zum 64. Bit das *Subscriber*[32]-Feld, das den Kunden des *Providers* identifiziert. Es hat die Länge 56-n.
- Vom 65. Bit an steht die *Subnet*[33]-ID. Ihre Länge kann variieren. Es bieten sich jedoch hier 16 Bit an, damit für die
- *Interface*[34]-ID, die innerhalb des Subnetzes einen Rechner eindeutig identifiziert, die restlichen 48 Bit verwendet werden können.

Die angedeutete Aufteilung des Intrasubscriber-Bereichs ist deshalb empfehlenswert, weil alle Ethernetkarten eine weltweit eindeutige Nummer in ihrem Basic I/O System (BIOS) eingebrannt haben, die 48 Bit lang sind. Dies ermöglicht dann eine zustandslose Selbstkonfiguration: Zum Beispiel haben in einer Firma alle Rechner in einem *Subnetz* bis auf die letzten 48 Bits die gleiche Adresse, und eben diese lassen sich aus der Ethernetkarte auslesen.

Dieser modulare Aufbau trägt der Dynamik Rechnung, der das Internet und deren Teilnehmer unterworfen sind. Wechselt eine Firma beispielsweise den *Provider*, ändert sich nur der Präfix der Adressen.

Durch den hierarchischen Aufbau kann das *routing* anhand des Adress-Präfix erfolgen.

In der Übergangsphase sollen beide Protokolle gleichzeitig benutzt werden können. Deshalb integriert man die IPv4-Adressen in eine reservierte IPv6-Adresse (siehe Abb. 2). Die Notation hierfür ist eine Kombination aus bis zu sechs Hexadezimalzahlen und vier Dezimalzahlen, wie bei IPv4.

Es sind außer den Providerbasierten- noch verbindungslokale und ortslokale Adressen mit jeweils speziellem Präfix vorgesehen, auf die hier jedoch nicht weiter eingegangen werden soll.

Im Gegensatz zu *Unicast*-Adressen, sprechen *Multicast*-Adressen eine ganze Gruppe von Rechnern an. Sie dienen dazu, Bandbreite zu sparen, da sie bereits von der IP-Schicht ausgewertet werden und

[31] engl. für Register.
[32] engl. für Teilnehmer.
[33] engl. für „Unternetz“, also ein Netz innerhalb eines Netzes.
[34] engl. für Schnittstelle.

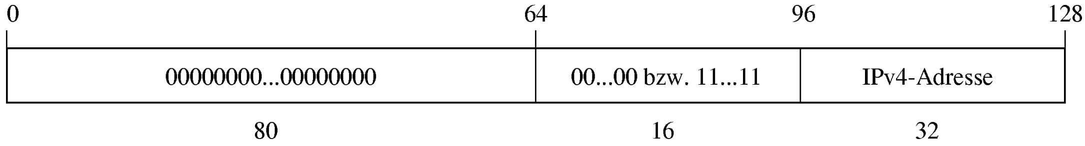

Abb. 3: IPv4-Adressen im IPv6-Format (nach [Hose96])

so eine Mehrfachübertragung identischer Datagramme vermieden wird. Dies ist insbesondere für Multimediaanwendungen wie Video- oder Audio-Übertragung an ein breites Publikum erforderlich.

Anycast-Adressen sprechen aus einer Gruppe von Rechnern den schnellsten an. Dies ist sinnvoll bei Anwendungen wie dem *domain name service*[35] (DNS) und bei dynamischen Netzwerkumgebungen. *Anycast*-Adressen sind wie *Unicast*-Adressen aufgebaut; es teilen sich jedoch alle Stationen einer *Anycast*-Gruppe dieselbe.

2.2.2 Vereinfachung des Protokollkopfes

Im Vergleich zu IPv4 hat sich der IPv6-Basiskopf (siehe Abb. 5) stark vereinfacht. Einige Felder sind ganz fallengelassen worden, da sich gezeigt hat, daß sie – außer das *routing* zu verlangsamen – keine Bedeutung haben, andere sind in sogenannte *Extension Header*[36] verlagert worden. Diese Erweiterungsköpfe dürfen, falls erforderlich, hintereinander in einer festgelegten Reihenfolge zwischen den IPv6-Basiskopf und den *Header* des höheren Protokolls (TCP[37]/UDP[38]/ICMP[39]) gereiht werden.

In jedem IPv6-(*Extension*-) *Header* wird der Typ des darauffolgenden *Header* anhand einer festgelegten 8-Bit-ID angegeben.

Extension Header dienen dazu, erweiterte Optionen oder Anforderungen aufzunehmen. Diese Aufspaltung erlaubt es u.a., nachträglich Protokoll-Erweiterungen durchzuführen. Im Normalfall müssen Router zwischen Sender und Empfänger nur den Basis-*Header* analysieren. Nur in Fällen, in denen eine besondere Behandlung des Pakets erwünscht ist (besonderes *Routing*, Angaben im *Hop-by-Hop Option Header*), sind weitere Köpfe zu untersuchen.

Die Köpfe sind (in der festgelegten Reihenfolge):

- *IPv6 header*

  Der IPv6 Basiskopf.

---

[35] engl. für Dienst, der Rechnernamen in Rechneradressen umsetzt.

[36] engl. Erweiterungs-(Protokoll-)kopf.

[37] engl. Akronym für „transmission control protocol“, verbindungsortientiertes Übermittlungssteuerungsprotokoll.

[38] engl. Akronym für „user datagram protocol“, verbindungsloses Protokol zur Übermittlung von kurzen Nachrichten.

[39] engl. Akronym für „Internet control message protocol“, Protokoll zum übermitteln von Kontrollnachrichten im Internet.

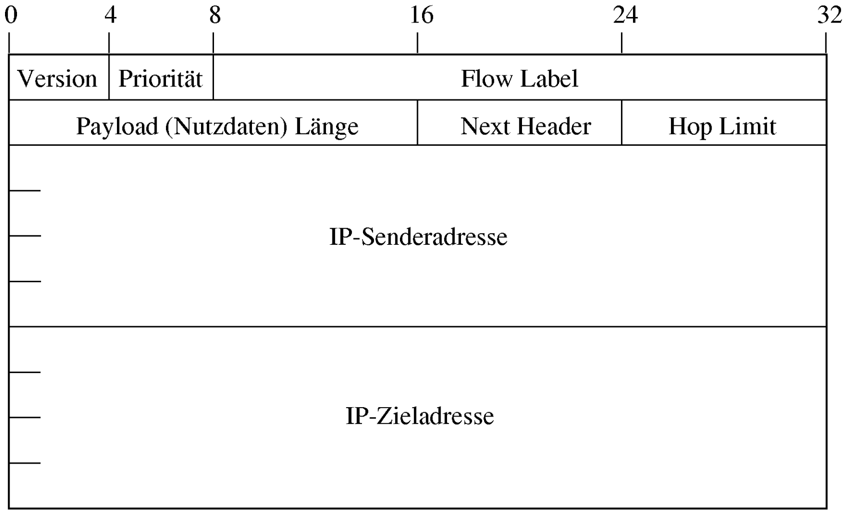

Abb. 4: Der IPv6 Basiskopf (nach [DeHi95])

- *Hop-by-Hop Options header*
  Dieser Erweiterungskopf wird benutzt, um optionale Informationen aufzunehmen, die von jedem Knoten auf dem Weg des Pakets analysiert werden müssen.
- *Destination Options header*
  Dieser Kopf wird an dieser Stelle nur bei existierendem *Routing header* verwendet; die in ihm enthaltenen Optionen werden dann von allen im *Routing header* angegebenen Stationen ausgewertet.
- *Routing header*
  Der Sender hat hier die Möglichkeit, Knotenpunkte aufzulisten, die auf dem Weg des Pakets zum Empfänger besucht werden sollen.
- *Fragment*[40] *header*
  Mit Hilfe dieses Kopfes hat der Sender die Möglichkeit Pakete zu fragmentieren, falls die *Maximum Transfer Unit* (MTU) auf dem Pfad kleiner ist als die Größe des Pakets. Erst beim Empfänger wird das Paket wieder zusammengesetzt.
- *Authentication*[41] *header* (AH)
  Dieser Kopf dient der Authentifizierung (Beglaubigung) und Sicherung von Integrität des gesamten

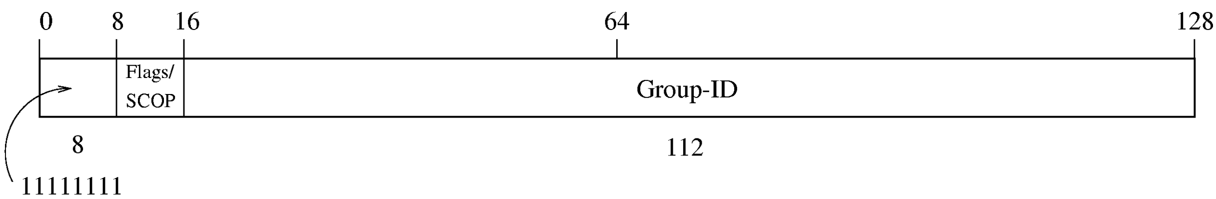

Abb. 5: Multicast-Adressen bei IPv6 (nach [Hose96])

[40] engl. für Bruchstück.
[41] engl. für Authentifizierung (Beglaubigung).

Datenpackets (siehe Kapitel 3). Bei Wahl geeigneter Algorithmen kann er auch als Schutz vor Zurückweisbarkeit dienen.

- *Encapsulating Security Payload*[42] *header* (ESP)
  Dieser Kopf enthält alle folgenden Header (samt Datagrammen) in verschlüsselter Form und bietet dadurch Vertraulichkeit und bei Wahl geeigneter Algorithmen auch darüberhinaus Authentizität und Integrität der Daten. Er kann dazu aber auch in Verbindung mit AH benutzt werden.
- *Destination Options header*
  Die Angaben in diesem Kopf werden nur vom letzten (eigentlichen) Empfänger ausgewertet.
- *High level header*
  Im Normalfall steht hier der Header des darüberliegenden Protokolls. Es darf hier jedoch auch ein komplettes IP-Diagramm stehen, das dann vom ersten gekapselt ist. Dies macht Sinn, falls ganze Pakete getunnelt oder mittels *Encapsulating Security Payload* geschützt werden sollen.

AH und ESP-Header werden in Kapitel 3 beschrieben. Eine genaue Definition der anderen Köpfe findet man in [DeHi95].

3 Sicherheitsmechanismen

Ein Hauptziel beim Erstellen der Sicherheitsarchitektur war, Benutzern die es wünschen, solide Sicherheitsmechanismen zu bieten, ohne daß den anderen dabei Nachteile entstehen. Die Architektur ist außerdem unabhängig von verwendeten Krypto-Algorithmen, so daß Algorithmen unabhängig vom Rest der Implementierung ausgetauscht werden können.

Um Zusammenarbeit im globalen Netz zu ermöglichen, sind Standard-Algorithmen vorgesehen, die jeder IPv6-Stapel implementieren muß; Sender und Empfänger können jedoch vereinbaren, daß andere (z.B. sicherere, siehe Kapitel 4) Algorithmen zur Verwendung kommen sollen.

Im heutigen IPv4-basierten Internet sind eine Anzahl von Angriffen bekannt. Bei einem „*IP-Spoofing*“ genannten Angriff täuscht der Angreifer durch Manipulation des Sender-Feldes an einem ausgehenden IP-Paket vor, das Paket käme von einem Rechner, dem der Empfänger vertraut.

Durch geschicktes Ausnutzen dieser Sicherheitslücke kann er beispielsweise durch „*DNS spoofing*“, einer Variante des IP-*Spoofings*, erreichen, daß in einem DNS-Server die Umsetzung von Rechnername zu Rechneradresse nicht mehr richtig vorgenommen wird und später andere Rechner, die beispielsweise nach der IP-Adresse einer Bank fragen, diejenige des Rechners des Angreifers erhalten.

Der *Authentication Header* (AH) dient der Authentifizierung und Integrität des übertragenen Datagramms. Der Empfänger hat dann die Möglichkeit, zu überprüfen, ob der Sender eines Pakets auch wirklich derjenige ist, wofür er sich ausgibt und ob Informationen während der Übertragung verändert

[42] engl. für „zur Sicherheit eingekapselte Nutzdaten“

wurden. Bei Wahl eines geeigneten Algorithmus kann er nachträglich sogar beweisen, eine Information von einer bestimmten Person erhalten zu haben (Nicht-Zurückweisbarkeit).
Eine weitere Schwachstelle bei IPv4 ist, daß alle Daten unverschlüsselt übermittelt werden, falls nicht auf einer höheren Netzwerkschicht Verfahren zur Verschlüsselung zum Einsatz kommen. Durch einen Angriff, der als „*Packet-Sniffing*“ bekannt geworden ist, kann ein Angreifer alle Pakete mitlesen, die durch sein (Ethernet-) Netzwerk geroutet werden. Da normalerweise niemand im Internet Einfluß darauf hat, welchen Weg die zu übertragende Information vom Sender zum Empfänger nimmt, müssen bei sicherheitsrelevantem Datentransfer Techniken zum Einsatz kommen, die das Mitlesen einer dritten Person verhindern.
Das *Encapsulating Security Payload* (ESP) Verfahren dient der Sicherheitskapselung, also der Verschlüsselung eines Datagramms oder eines Teils davon. Bei Wahl bestimmter Algorithmen kann es auch der Authentizität dienen.
Die Trennung der beiden Sicherheitsoptionen ist politisch motiviert. Dadurch, daß der AH keine Vertraulichkeit liefert, ist garantiert, daß Implementierungen davon weit verbreitet sein können, auch in Ländern, in denen Ein- oder Ausfuhr oder Benutzung von kryptographischen Methoden mit dem Ziel der Verschlüsselung verboten ist.

3.1 Typische Einsatzszenarien

Ein einfaches Szenario (vgl. Abb. 6) ist folgendes: Eine Firma hat Standorte in Dortmund und Frankfurt und möchte diese aus Kostengründen nicht direkt per Standleitung miteinander vernetzen, sondern über das Internet. Bei beiden Standorten ist vorgesehen, daß der Internetzugang außer zur Vernetzung der Standorte auch zum Einholen von Informationen usw. genutzt werden soll.
Es soll möglich sein, gefälschte Pakete zu identifizieren und da auch vertrauliche, nicht für die Konkurrenz bestimmte Daten zwischen den Standorten ausgetauscht werden sollen, soll der gesamte Datenverkehr zwischen beiden Standorten verschlüsselt werden.
Es wird davon ausgegangen, daß innerhalb der Standorte nur vertrauenswerte Mitarbeiter arbeiten, die sich weder an aktiven noch an passiven Angriffen beteiligen, und daß der dem IP zugrundeliegende Kanal innerhalb der Netzwerke (z.B. Ethernet) nicht angegriffen wird.
Die Firma baut deshalb ihre Gateways zum Internet hin zu Sicherheits-Gateways aus. Diese Sicherheits-Gateways werden so konfiguriert, daß Pakete, die vorgeben, vom anderen Standort zu kommen, nur dann in das lokale Netz weitergeleitet werden, wenn sie mittels AH korrekt authentifiziert sind. Dazu versehen sie selbst auch alle Pakete aus dem jeweils eigenen Netzwerk, die für den anderen Standort bestimmt sind, mit einem entsprechenden AH. Zusätzlich werden die eigentlichen Daten mittels ESP gekapselt.
In diesem Szenario muß AH und ESP nur im Sicherheits-Gateway konfiguriert werden. Die Rechner innerhalb der vertrauenswürdigen Sub-Netze bedienen sich beim Austausch von Informationen des

Sicherheits-Gateways. Nur zwischen den beiden Sicherheits-Gateways müssen so Sicherheitskombinationen (bestehend aus Algorithmus und Schlüssel, siehe nächstes Kapitel) ausgetauscht werden.

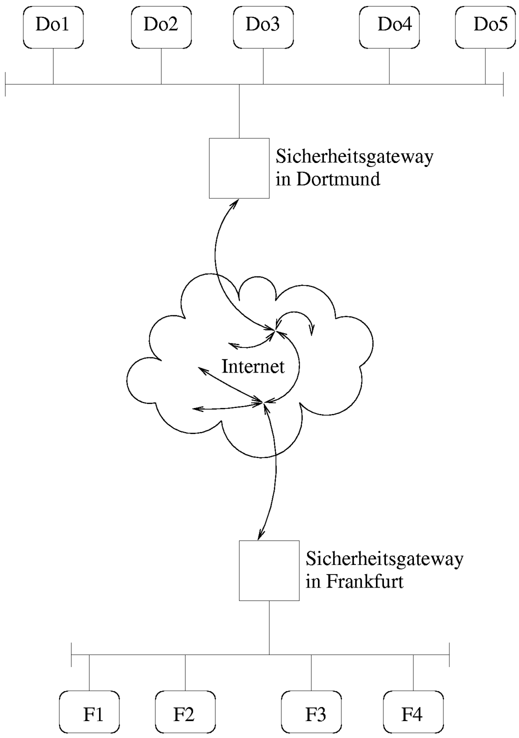

Abb. 6: Beispielszenario

Es ist jedoch auch möglich (und sicherer), für jedes Rechnerpaar oder bei Mehrbenutzersystemen sogar für jede Benutzer-Zielrechner-Kombination Sicherheitskombinationen auszutauschen. In diesem Fall wird ein Sicherheitsgateway nicht mehr benötigt.

3.2 Security Associations (Sicherheitskombinationen)

Sicherheitskombinationen sind sowohl für AH als auch für ESP grundlegend. Um AH und/oder ESP einsetzen zu können, ist es erforderlich, daß sich Sender und Empfänger auf kryptographische Verfahren geeinigt haben, und daß sie entsprechende Schlüssel miteinander ausgetauscht haben.

Die Kombination eines *Security Parameter Index*[43] (SPI) mit der Zieladresse bildet eine eindeutige „Sicherheitskombination". Eine Sicherheitskombination beinhaltet dann (mindestens) folgende Parameter (nach [Atki95a]):

- Authentifizierungsalgorithmus der bei AH verwendet werden soll,
- ein oder mehrere Schlüssel, der/die bei AH mit dem verwendeten Algorithmus benutzt werden soll(en),

[43] engl. für Sicherheitsparameter-Index

- Verschlüsselungsalgorithmus, Modus des Algorithmus und Transformation, die bei ESP verwendet werden sollen,
- ein oder mehrere Schlüssel, der/die bei ESP mit dem verwendeten Algorithmus benutzt werden soll(en),
- Information darüber, ob eine kryptographischen Synchronisation oder ein Initialisierungsvektor-Feld für den Verschlüsselungsalgorithmus notwendig ist (falls ja auch dessen Größe),
- Authentifizierungsalgorithmus und Modus, der bei der ESP-Transformation benutzt werden soll (empfohlen für ESP-Implementierungen),
- Authentifizierungsschlüssel, der/die (gegebenenfalls) mit dem Authentifizierungsalgorithmus verwendet werden soll(en), der Teil der ESP-Transformation ist (empfohlen für ESP-Implementierungen),
- Gültigkeitszeitraum des Schlüssels oder Zeitpunkt, wann ein Wechsel des Schlüssels erfolgen sollte (empfohlen für ESP-Implementierungen),
- Gültigkeitszeitraum dieser Sicherheitskombination (empfohlen für ESP-Implementierungen),
- Quelladresse(n) der Sicherheitskombination, kann eine Wildcard-Adresse sein, falls mehr als ein Sendesystem die gleiche Sicherheitskombination mit dem Zielsystem teilt (empfohlen für ESP-Implementierungen),
- Empfindlichkeitseinstufung (z.B. geheim oder vertraulich) der geschützten Daten (empfohlen für alle Implementierungen; zwingend erforderlich für Systeme, die vorgeben, mehrstufige Sicherheit zu unterstützen)

Das sendende System benutzt die Benuzter-ID und die Zieladresse, um die entsprechende Sicherheitskombination (und damit den SPI-Wert) auszuwählen.

Das empfangende System benutzt SPI-Wert in Verbindung mit der Zieladresse, um die korrekte Kombination zu ermitteln. So kann es für alle gültigen einlaufenden Pakete die korrekte Sicherheitskonfiguration ermitteln.

Falls eine vorher gültige Sicherheitskombination ungültig wird, sollte das Zielsystem (welches die Sicherheitskombination erzeugt), eine gewisse Zeit warten, bis es denselben SPI-Wert für eine andere Sicherheitskombination wiederverwendet.

Eine Sicherheitskombination ist für beide Richtungen notwendig. Eine authentifizierte Kommunikationssitzung benutzt deshalb zwei SPI, einen für jede Richtung.

Die Verbindung eines SPI und einer Zieladresse identifiziert eine Sicherheitskombination in eindeutiger Weise. Die Zieladresse kann dabei eine *Unicast*-Adresse oder die Adresse einer *Multicast*-Gruppe sein.

Die Empfänger-Orientierung der Sicherheitskombination impliziert, daß bei *Unicast*-Adressen normalerweise der Empfänger den SPI-Wert wählt. Bei *Multicast*-Datenverkehr gibt es mehrere Zielsysteme, aber nur eine *Multicast*-Gruppen-Adresse, weshalb ein System oder eine Person den SPI im

Auftrag der dieser Gruppe festlegen muß, um ihn dann allen legitimen Mitgliedern der Gruppe zugänglich zu machen.

3.3 Authentication Header

Der Authentication Header (AH) ermöglicht Authentifizierungs- und Integritätssicherung von IP Datagrammen mittels starker kryptographischer Verfahren. Die Benutzung von asymmetrischen Signaturalgorithmen wie RSA anstelle der Standardalgorithmen kann dem Empfänger darüber hinaus auch die Sicherheit bieten, daß der Sender im nachhinein nicht behaupten kann, er hätte die Information nie geliefert.

Vertraulichkeit und Schutz vor Verkehrsflußanalysen liefert dieses Verfahren nicht. Um Vertraulichkeit zu Erreichen kann IP *Encapsulating Security Payload* (siehe Kapitel 3.3) entweder anstelle von oder in Verbindung mit AH benutzt werden. Mittel zum Schutz vor Verkehrsflußanalysen findet man u.a. in [VoKe83].

Der *IP Authentication Header* versucht, Sicherheit zur Verfügung zu stellen, indem er dem IP Datagramm Authentifizierungsdaten hinzufügt. Diese Authentifizierungsdaten werden mit Hilfe eines Algorithmus (in Verbindung mit einem Schlüssel) berechnet, der auf alle Daten des IP Datagramms angewandt wird, die sich bei dem Transport nicht ändern. Felder innerhalb der Header, die sich laut Spezifikation während des Transport ändern (z.B. „*hop count*", „*time to live*", „*ident*", „*fragment offset*" oder „*routing pointer*"), gehen in diese Berechnung als Nullwerte ein.

Der AH ist ein *Extension Header* des IPv6 Protokolls, kann jedoch auch bei IPv4 benutzt werden. Er wird in beiden Fällen an eine Stelle zwischen dem IP-Header und dem Kopf des höheren Protokolls innerhalb des Datagramms, das authentifiziert werden soll, eingehängt.

IPv6Header	Hop-by-Hop/Routing	Auth Header	Others	Upper Protocol

Abb. 7: Der AH im IPv6-Datagramm (aus [Atki95b])

Bei IPv6 ist die Position an der der AH stehen darf, genau definiert (siehe Kapitel 2.2.2); bei IPv4 folgt der AH direkt dem Protokollkopf.

IPv4Header	Auth Header	Upper Protocol (e.g. TCP, UDP)

Abb. 8: Der AH im IPv4-Datagramm (aus [Atki95b])

Jeder IP-Stack, der vorgibt, AH zu implementieren, muß mindestens den Authentifizierungsalgorithmus „*keyed MD5*" [44] [MeSi95] unterstützen.

[44] engl. für Schlüssel benutzende Version des MD5-Algorithmus.

3.3.1 Aufbau des Authentication Header

Der Kopf selbst ist wie in Abb. 9 gezeigt aufgebaut.

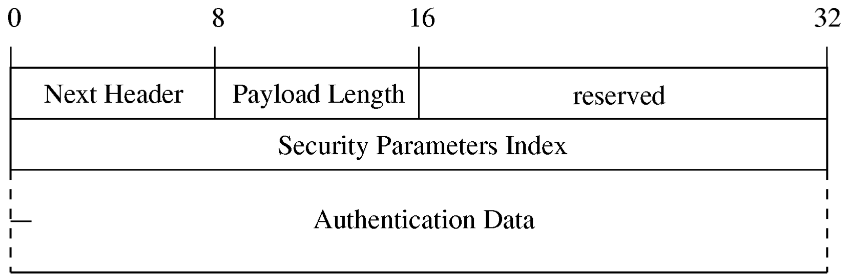

Abb. 9: *Authentication Header* (aus [Atki95b])

Die Felder haben folgenden Inhalt:

- *Next Header*

  Dieses Feld ist 8 Bits groß. Es beschreibt die Art des dem AH folgenden Kopfes. (Die Werte werden von der *Internet Assigned Numbers Authority*[45] (IANA) vergeben).

- *Payload Length*[46] (Authentifizierungsdaten-Länge)

  Dieses Feld ist ebenfalls 8 Bits groß. Es gibt die Anzahl der 32 Bit-Wörter an, die für die Authentifizierungsdaten gebraucht werden.

- *reserved*

  Dieses 16 Bits große Feld ist für späteren Gebrauch reserviert. Der Sender muß es mit Nullen füllen. Es wird in die Berechnung des Algorithmus einbezogen, aber der Empfänger muß es zum jetzigen Zeitpunkt ignorieren.

- *Security Parameters Index*

  Ein 32 Bit Pseudozufallswert, der die Sicherheitskombination für dieses Datagramm identifiziert. Der SPI-Wert 0 ist reserviert um anzuzeigen, daß keine Sicherheitskombination existiert.

  Die Menge der SPI-Werte im Bereich von 1 bis 255 sind von der IANA für spätere Benutzung reserviert.

- *Authentication Data*

  Die Länge dieses Feldes ist variabel, jedoch stets ein Vielfaches von 32 Bits. Falls die Authentifizierungsdaten nicht ein Vielfaches von 32 Bits groß sind und gefüllt werden muß, geschieht dies auf eine von der Implementation abhängige Art und Weise. Die Größe und Art der Benutzung dieses Feldes bleibt jedoch für alle Datagramms mit demselben Paar aus SPI und Zieladresse gleich.

---

[45] engl. für internetbezogene Nummernvergabestelle.
[46] engl. für Länge der Nutzdaten (hier: Länge der von *keyed MD5* erzeugten Ausgabe).

3.3.2 Berechnung der Authentifizierungsdaten

Die Authentifizierungsdaten, die im AH enthalten sind, werden üblicherweise mit einem „*Message Digest*[47]“-Algorithmus (z.B. MD5) berechnet, wobei danach entweder dessen Berechnung verschlüsselt wird oder der Algorithmus gleich mit Schlüsseln arbeitet (siehe [MeSi92]). Nur Algorithmen, von denen geglaubt wird, daß sie im kryptographischen Sinne starke Ein-Weg Funktionen sind, sollten zur Verwendung kommen.

Konventionelle Algorithmen zur Prüfsummenberechnung wie z.B. CRC[48]-16 sind nicht kryptographisch stark und dürfen deshalb nicht im AH zur Anwendung kommen.

Zur Verarbeitung eines ausgehenden Pakets wird nun folgendermaßen vorgegangen: Zuerst ermittelt das sendende System die Sicherheitskombination. Sicherheitskombinationen sind immer unidirektional.

Die Wahl der Sicherheitskombination ist mindestens abhängig von der Benutzer-ID des Senders und der IP-Adresse des Empfängers. Bei System-orientierter Schlüsselverwaltung teilen sich alle Benutzer dieselbe Sicherheitskombination zu einem Empfangssystem. Bei Benutzer-orientierter Schlüsselverwaltung können verschiedene Benutzer oder sogar verschiedene Anwendungen des gleichen Benutzers unterschiedliche Sicherheitskombinationen benutzen.

Die ermittelte Sicherheitskombination gibt an, welcher Algorithmus in welchem Modus mit welchem Schlüssel und anderen Eigenschaften auf das ausgehende Paket angewandt wird.

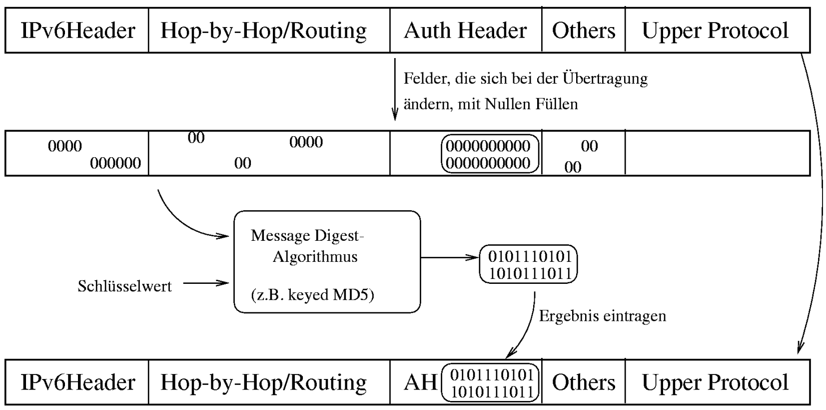

Abb. 10: Funktionsweise von AH

Wie bereits oben erwähnt werden Werte in Feldern, die sich während des Transports ändern müssen bei der Berechnung des Algorithmus als Nullwerte angesehen. Innerhalb einiger IPv6 Erweiterungsköpfe gibt es Bits innerhalb von Feldern, die anzeigen, ob das Feld in die Kalkulation mit seinem Wert

[47] engl. für Nachrichten-Auslese, meint hier das Bilden einer Wertes aus einer Nachricht mittels einer „schwer“ umkehrbaren Hash-Funktion.

[48] engl. Akronym für „cyclic redundancy check“, zyklischer Redundanzcheck, (leicht umkehrbarer) Algorithmus, mit dem ein Prüfsumme über Daten gebildet werden kann.

oder mit Nullwert einbezogen werden sollen. Auch das „*Authentication Data*“-Feld im AH, in das der berechnete Wert geschrieben wird, wird behandelt, als wäre es mit Nullen gefüllt.

Wenn ein Empfänger ein Paket erhält, das einen AH beinhaltet, ermittelt auch er mit Hilfe der SPI und der Zieladresse die Sicherheitskombination. Dann überprüft er, ob das „*Authentication Data*“-Feld und das Paket konsistent sind. Bei der Anwendung des Authentifizierungsalgorithmus gelten für ihn die selben Regeln bezüglich als Null anzusehender Felder wie für den Sender. Falls der Algorithmus anzeigt, daß Datagramm und AH zueinander passen, wird das Paket akzeptiert; ansonsten sollte er es als ungültig verwerfen und den Vorfall im Ereignisprotokoll erfaßt. Der Eintrag muß dann mindestens den SPI-Wert, den Zeitpunkt des Empfangs, die Adresse des Senders und Empfängers und (nur bei IPv6, falls existent) das Datenflußkennzeichen beinhalten.

3.4 Encapsulating Security Payload

Der „*Encapsulating Security Payload* (ESP)“-Mechanismus ermöglicht sichere Übertragung von IP Datagrammen im Sinne von Integrität und Vertraulichkeit – bei Wahl entsprechender Algorithmen kommt Authentifizierung hinzu.

Nicht-Zurückweisbarkeit und Schutz vor Verkehrsflußanalysen liefert dieses Verfahren nicht. Um Nicht-Zurückweisbarkeit zu erreichen kann IP Authentication Header (siehe Kapitel 3.2) entweder anstelle von oder in Verbindung mit ESP benutzt werden. Mittel zum Schutz vor Verkehrsflußanalysen findet man u.a. in [VoKe83].

ESP kann in zwei Modi zum Einsatz kommen:

- Im Tunnel-Modus wird das ganze originäre Datagramm verschlüsselt in einem ESP-Erweiterungskopf gekapselt. Diesem wird nun ein unverschlüsselter IP-Kopf vorangestellt, der beschreibt, wohin das verschlüsselte Datagramm geroutet werden soll. Es darf auch ein unverschlüsselter IP-Routing-Kopf zwischen IP-*Header* und ESP-*Header* eingefügt werden.
- Im Transport-Modus wird der ESP *Header* direkt vor den *Header* der Transportschicht gestellt und nur dieser verschlüsselt im ESP-Kopf geschützt. In diesem Modus wird Bandbreite gespart, da weniger Daten verschlüsselt werden.

Im Zusammenspiel mit ESP darf ein Authentication Header im unverschlüsselten Teil des Pakets – im ESP Tunnelmodus außerdem zusätzlich im verschlüsselten Teil des Pakets benutzt werden. Das Ziel im ersten Fall ist, vor allem die Klartextköpfe vor Veränderung zu schützen bzw. zu authentifizieren. In diesem Fall ist zunächst ESP und dann AH auf das ganze Paket anzuwenden. Im zweiten Fall werden nur die verschlüsselten Daten authentifiziert (was jedoch auch durch eine entsprechende Transformation realisiert werden kann). Dann wird erst AH und dann ESP auf das zu tunnelnde Datagramm angewandt.

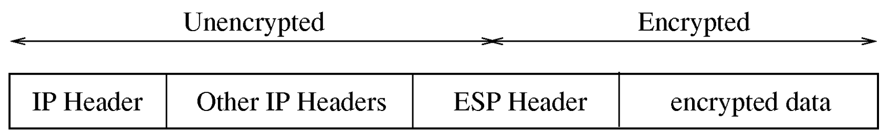

Abb. 11: Der ESP-Kopf im IP-Datagramm (aus [Atki95c])

Der ESP-Kopf ist wie der AH ein *Extension Header* des IPv6 Protokolls, kann jedoch auch bei IPv4 benutzt werden. Er kann an einer beliebigen Stelle zwischen Protokollkopf und Rahmen des höheren Protokolls stehen.

Jeder IP-Protokollstapel, der vorgibt, ESP zu implementieren, muß mindestens den „DES-CBC"-Verschlüsselungsalgorithmus unterstützen, dessen Anwendung in Zusammenhang mit ESP in [KMS95] genau beschrieben wird.

3.4.1 Aufbau des ESP Kopfes

ESP besteht aus einem unverschlüsselten Kopf gefolgt von verschlüsselten Daten. Die verschlüsselten Daten beinhalten sowohl die geschützten ESP-*Header*-Felder als auch die geschützten Benutzerdaten, welche entweder ein komplettes IP-Datagramm darstellen (im Tunnel-Modus) oder aus einem Rahmen des nächsthöheren Protokolls (z.B. TCP oder UDP) bestehen (im Transport-Modus).

Verschlüsselungs- und Authentifizierungsalgorithmus und das genaue Format der verschlüsselten Daten bei ihrer Anwendung werden als Transformation (engl.: „*transform*") bezeichnet. Das ESP-Format wurde so entwickelt, daß neue zusätzliche Transformationen entwickelt werden können und ihre Anwendung finden können. Die Pflicht-Transformation „DES-CBC", die jeder ESP-fähige IPv6-Protokollstack implementieren muß, wird in [KMS95] definiert.

Das einzige Pflicht-Feld im ESP-*Header* für alle Transformationen ist das SPI-Feld, ein 32 Bit-Wert, der die Sicherheitskombination für dieses Datagramm angibt.

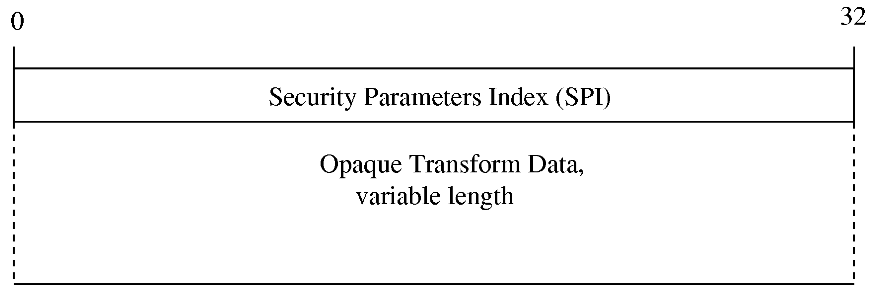

Abb. 12: *Encapsulating Security Payload Header* (aus [Atki95c])

Wie bei AH wird ein SPI von 0x00000000 als Zeichen dafür benutzt, daß keine Sicherheitskombination ermittelt werden konnte. Werte im Bereich von 0x00000000 bis 0x000000FF sind reserviert.

Transformationen können im verschlüsselten Bereich des ESP-Kopfes weitere Felder definieren, deren Art und Bedeutung in deren Spezifikation genau festgelegt sein muß.

3.4.2 Funktionsweise von ESP

Dieses Kapitel beschreibt die Schritte, die bei der Benutzung von ESP in den beiden Modi vorgenommen werden müssen. Die Anwendung der Transformation erfolgt sowohl im Tunnel- wie auch im Transportmodus vor einer möglichen Fragmentierung auf Senderseite und nach Zusammensetzung der Fragmente auf Empfängerseite.

Im ESP-Tunnelmodus steht der ESP Kopf hinter den *Headern*, die notwendigerweise im Klartext übertragen werden müssen und beinhaltet das komplette originäre zu tunnelnde IP Datagramm in verschlüsselter Form.

Ähnlich wie bei AH ermittelt der Sender zunächst anhand der Benutzer-ID und der Zieladresse die zu benutzende Sicherheitskombination. Danach wendet er die ermittelte Transformation auf das zu tunnelnde IP-Datagramm an und erstellt mit deren Ergebnis und dem SPI der Sicherheitskombination einen ESP Kopf. Dieser ESP-Kopf wird nun als letzter *Payload* hinter die Klartext-IP-Köpfe gestellt, die für die Übertragung notwendig sind. Die Information in den Klartext- (Erweiterungs-) Köpfen darf sich dabei von denen im gekapselten und verschlüsselten originalen IP-Datagramm unterscheiden.

Der Empfänger verwirft die Klartextköpfe (nachdem er ggf. deren Gültigkeit mit Hilfe eines Klartext-AH überprüft hat) und ermittelt danach anhand der Zieladresse des Pakets und des SPI im ESP-Kopf die Sicherheitskombination und damit die Transformation samt Schlüssel. Falls keine gültige Sicherheitskombination existiert, verwirft er den verschlüsselten ESP und erfaßt den Fehler im Ereignisprotokoll des Systems. Ein solcher Eintrag sollte den SPI-Wert, den Zeitpunkt, die Sender- und Empfängeradresse (aus dem Klartextbereich), und das Datenflußkennzeichen (ebenfalls aus dem Klartextbereich, falls vorhanden) enthalten.

Falls jedoch die Entschlüsselung gelingt, wird das originale IP Datagramm vom (nun entschlüsselten) ESP entfernt und gemäß der normalen IP Spezifikation weiterverarbeitet.

Im ESP-Transportmodus wird im ESP-Kopf im Gegensatz zum Tunnelmodus kein ganzes IP Datagramm gekapselt, sondern lediglich der Rahmen des höheren Protokolls (z.B. TCP, UDP, ICMP).

Die Schritte bei der Verschlüsselung auf Senderseite sind ansonsten die gleichen. Auf Empfängerseite wird nach der Entschlüsselung ähnlich wie bei normalem IP festgestellt, an welches Transportprotokoll der entschlüsselte Klartextrahmen weitergeleitet werden muß. Dort sind Mechanismen angesiedelt, die gegebenenfalls reagieren können, falls die Rücktransformation mißlang.

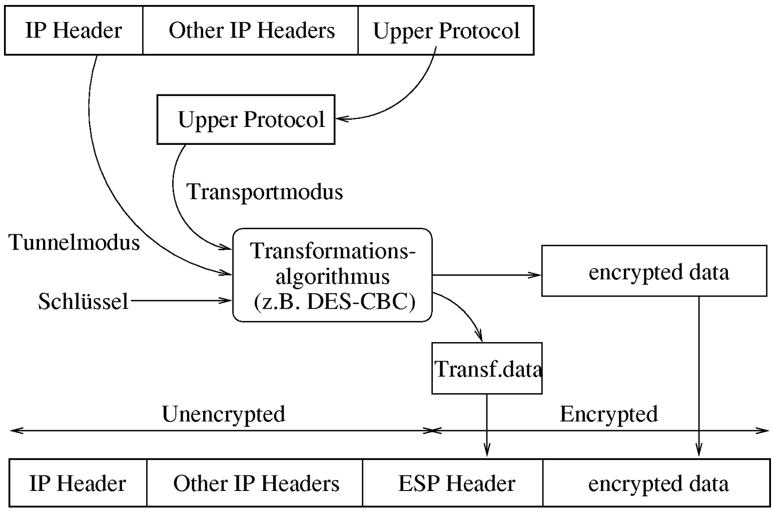

Abb. 13: Funktionsweise von ESP

3.5 Einfluß auf den Datendurchsatz

Beide Maßnahmen verlängern zum einen die IP Datagramme, was zu Lasten der Bandbreite geht; andererseits wird auch Rechenzeit für den Authentifizierungs- bzw. Transformationsalgorithmus sowohl auf Sender- als auch auf Empfängerseite benötigt, was zu einer Verzögerung des Datentransfers führen kann.

Die Rechenzeit (und damit die Verzögerung) ist dabei in erster Linie vom verwendeten Algorithmus abhängig, allgemeine Aussagen sind deshalb nicht machbar. In Routern zwischen Sender und Empfänger fällt nicht mehr Bearbeitungszeit an, wenn diese nicht auch in die Sicherheitsoptionen eingebunden sind. Hardwarelösungen (Spezialchips für Ver- und Entschlüsselung) werden für hohen Durchsatz empfohlen, sind dann jedoch i.A. auf einen oder wenige Algorithmen beschränkt.

Auswirkung auf die Größe der Datagramme hat vor allem ESP im Tunnelmodus, da u.U. dieselben Erweiterungsköpfe im Klartext und in verschlüsselter Form übertragen werden, aber auch AH durch den zusätzlichen Erweiterungskopf.

Der negative Einfluß auf den Datendurchsatz sollte jedoch zu vernachlässigen sein, wenn man den Gewinn an Sicherheit gegenüberstellt, den man durch die Optionen bei korrekter Handhabung erlangen kann.

3.6 Schlüsselverwaltung

Die Schlüsselverwaltung ist ein sehr wichtiger Punkt in der IP-Sicherheitsarchitektur. Ein bestimmtes Schlüsselverwaltungsprotokoll ist jedoch in der IP-Spezifikation nicht festgelegt worden. Ein bedeutender Grund dafür war, daß in der Vergangenheit bei veröffentlichten Schlüsselverwaltungsprotokollen immer wieder Schwächen aufgedeckt wurden. Dies hat die Entwickler dazu veranlaßt, die

Sicherheitsarchitektur so offen zu gestalten, daß die Schlüsselververwaltung ohne Einfluß auf Implementierungen der Sicherheitsmechanismen ausgetauscht werden kann.
Die einzige Verbindung zwischen den Sicherheitsoptionen und der Schlüsselverwaltung besteht darin, daß es für AH und ESP notwendig ist, die Parameter einer Sicherheitskombination für ein gegebenes Paar bestehend aus Empfängeradresse und SPI bzw. Empfängeradresse und Benutzer-ID ermitteln zu können.
Die Verbreitung der Sicherheitskombinationen kann man auf verschiedenen Wegen erreichen (nach [Atki95a]):

- manuelle Verteilung
  In kleinen, statischen Umgebungen ist es noch handhabbar, daß auf jedem System die Sicherheitskombinationen von anderen Systemen per Hand konfiguriert werden. Dies soll keine mittel- oder langfristige Lösung darstellen, ist aber kurzfristig in kleineren Netzwerken praktikabel. Auch im in Kapitel 3.1 vorgestellten Szenario einer Firma, die ihre Netzwerke über das Internet miteinander verbindet, ist es durchaus machbar, die Sicherheitsgateways an den Standorten per Hand zu konfigurieren.
- existierende Schlüsselmanagement-Techniken
  Es finden sich eine Anzahl von Schlüsselmanagement-Algorithmen in der Literatur. Needham und Schroeder haben beispielsweise in [NeSc78] und [NeSc81] einen solchen vorgeschlagen, der auf einem zentralen Schlüsselverteilungssystem beruht. Dieser Algorithmus wird im Authentifizierungssystem von Kerberos verwendet, welches am MIT entwickelt wurde [KoNe93].
  Ein Algorithmus von Diffie und Hellman [DiHe76], benötigt kein zentrales Verteilungssystem, ist jedoch unglücklicherweise mit aktivem „*man in-the-middle*[49]“-Angriff verwundbar. Diese Angriffsmöglichkeit kann gemildert werden, wenn beim Austausch unterzeichnete Schlüssel verwendet werden [Schn94].
- automatisierte Schlüsselverteilung
  Eine verbreitete Anwendung der IP Sicherheitstechniken verlangt ein auf Internet-Größe skalierbares Schlüsselverteilungsverfahren. Es sind Arbeiten in der IETF[50] im Gange, das *Domain Name System* so zu erweitern, daß es einem Sender-/Empfängerpaar ermöglicht, mit Hilfe von asymmetrischen Algorithmen eine authentifizierte Verbindung untereinander aufzubauen. Auf dieser können dann mittels dezentralen Schlüsselverteilungsverfahren Sitzungsschlüssel vereinbart werden.

---

[49] engl. für „Mann in der Mitte“, der Angreifer hört hierbei den Datenaustausch der legalen Kommunikationspartner mit (passiv) oder greift zusätzlich ein (aktiv).
[50] engl. Akronym für „Internet Engeneering Task Force“, Organisation, die sich aus zahlreichen technisch Interessierten zusammensetzt, die in Arbeitsgruppen Vorschläge und Spezifikationen für die Internet-Protokolle erarbeiten.

Bereits erwähnt wurden die unterschiedlichen Herangehensweisen bezüglich der Frage, zu wem ein Schlüssel gehört (system-, benutzer- oder sogar applikationsabhängige Schlüsselverwaltung). Empfohlen wird in diesem Zusammenhang, daß eine benutzerbezogene Schlüsselverwaltung von allen Systemen unterstützt werden sollte, damit auch die Sicherheit des Datenverkehrs von sich gegenseitig mißtrauenden Benutzern gewährleitet ist.

4 Angriffsmöglichkeiten

4.1 Angriffe auf das System selbst

Benutzern sollte bewußt sein, daß die IP-Sicherheitsarchitektur die Daten erst ab der IP-Protokoll-Ebene sichert. Auf darübergelegenen Ebenen (Transport-, Anwendungsebene; bzw nach ISO-OSI[51] Transport-, Kommunikations-, Darstellungs- und Anwendungsebene) werden die Daten weiterhin ungeschützt behandelt. Falls ein Angreifer also zu einer der oberen Schichten im Sende- oder Empfangssystem Zugang hat, können die IP-Sicherheitsmaßnahmen dagegen keinen Schutz bieten.

Aber auch auf der IP-Ebene ist die Sicherheit von mehreren Faktoren abhängig. Diese Faktoren beinhalten die kryptographische Güte der verwendeten Kryptoalgorithmen (dazu in Kapitel 4.2 mehr), die Güte der benutzten Schlüsselverteilungssysteme und die Fehlerfreiheit der Implementierungen der IP-Stapel und Sicherheitsheitsmaßnahmen in den beteiligten Systemen.

Die Sicherheit, die Implementierungen bieten können, ist dabei abhängig von der Sicherheit, die das verwendete Betriebssystem selbst bietet. Wenn das Betriebssystem keine Möglichkeiten bietet, die verwendeten Schlüssel (also alle symmetrischen und alle privaten asymmetrischen) vertraulich zu halten, kann Datenverkehr mit diesen Schlüsseln nicht vertraulich sein. Falls mehrere Benutzer auf einem System verschiedene Schlüsseln verwenden, darf es nicht einem Benutzer durch Schwächen im Betriebssystem möglich sein, die Schlüssel eines anderen Benutzers auszuspähen.

Jedem, der per *World Wide Web* Informationen abruft, sollte die Machbarkeit von Angriffen auf die Sicherheit des Betriebssystems durch sog. *Applets*[52], die Sicherheitslücken in WWW-Browsern ausnutzen könnten oder „Active-X-Controls“ speziell beim Microsoft Internet Explorer, die praktisch gar keinen Sicherheitsbeschränkungen unterliegen, bekannt sein. Auch „Trojanische Pferde“ (Programme, die eine bestimmte Funktionalität bieten oder nur vorgaukeln, in Wirklichkeit jedoch als Transporthülse für andere, vom Benutzer nicht gewünschte Programmteile dienen) können die Systemsicherheit und damit auch die vorgestellten Sicherheitsmaßnahmen untergraben. Die in diesem Absatz genannten Probleme sind jedoch keine besondere Schwäche von ESP, AH oder IPv6. Diese Verwundbarkeit tritt

---

[51] engl. Akronym für „International Standards Organization, Open Systems Interconnection), Referenzmodell der Organisation für internationale Standards bezüglich der Vernetzung von Systemen.

[52] engl. Kunstwort, Verniedlichungsform von Applikation; hier: kleines Java-Programm, das (normalerweise) in einer – „Sandkasten“ genannten – abgeschotteten Umgebung innerhalb eines WWW-Browsers beim Betrachten einer HTML-Seite gestartet wird.

notwendigerweise bei allen Sicherheitssystemen auf, sobald ein Benutzer gleichzeitig vertrauensunwürdige Programme ausführen und Sicherheitsoptionen manipulieren darf.

Falls mehrere Benutzer, mehrere Rechner oder gar alle Rechner in einem LAN dieselbe Sicherheitskombination verwenden, ist zu beachten, daß der Empfänger keine Aussage darüber machen kann, von welchem Rechner dieser Gruppe genau ein Paket geschickt wurde. Er kann nur authentifizieren, daß es von einem der Mitglieder dieser Gruppe kam.

4.2 Angriffe auf die Standardalgorithmen

Wie schon oben erwähnt, ist in jeder Implementierung von AH „*keyed MD5*" (siehe [MeSi95]) vorgesehen. Schlüssellängen bis 128 Bits müssen, längere sollten unterstützt werden. Der Algorithmus erzeugt eine 128 Bits große Ausgabe, die den Payload des AH-Erweiterungskopfes darstellt. Angriffe mit einem 10 Millionen Dollar teuren System auf den normale MD5 Algorithmus dauern laut [vOWi94] 24 Tage, was jedoch von der Größe der Ausgabe abhängig ist. Es ist zwar nicht klar, ob der Angriff auf „*keyed MD5*" übertragbar ist – bei Notwendigkeit hoher Sicherheitsstandards sollte jedoch – auch in Anbetracht der fortschreitenden Entwicklung – auf eine modifizierte Version des Algorithmus mit längeren Hashwerten zurückgegriffen werden.

Als Transformation für ESP ist in jeder Implementierung der Algorithmus „*data encryption standard*"[53] (DES) im „*cypher block chaining*"[54]-Modus (CBC) (siehe [KMS95]) vorgeschrieben. Dabei kommen 56-Bit-Schlüssel zu Einsatz. Der 64 Bits große Initialisierungsvektor wird in jedem Datagramm geändert und soll sich während der Lebensdauer der Sicherheitskombination nicht wiederholen.

Um einen „*cut and paste*"[55]-Angriff [Bell95] zu verhindern, sollte zusätzlich AH verwendet werden, um die Integrität des Datagramms zu gewährleisten.

Während die Spezifikation für diesen Standardalgorithmus erarbeitet wurde, haben mehrere Experten für Kryptologie Schwächen des DES-Verfahrens aufgedeckt. Unter anderem hat M. J. Wiener ein System vorgestellt [Wien94], das nur eine Million US-Dollar kostet und durchschnittlich alle dreieinhalb Stunden einen DES-Schlüssel per „*known plaintext attack*"[56] mit dem Wissen von nur ein bis zwei Datenblöcken (à 64 Bits) brechen kann. Da IP-Datagramme mit bekannten oder leicht zu erratenden Köpfen beginnen, ist dieser Angriff extrem praktikabel, auch wenn der Schlüssel oft geändert wird.

Daher empfehlen die Autoren selbst, besser Tripel-DES[57] oder andere Verfahren mit ESP zu verwenden.

---

[53] engl. für Datenverschlüsselungsstandard.
[54] engl. für Hintereinanderketten von verschlüsselten Blöcken.
[55] engl. für ausschneiden und kleben, hier: Einfügen von Daten in ein Paket, die vorher aus einem anderen Paket ausgeschnittenen wurden.
[56] engl. für Angriff auf den Schlüssel bei bekanntem Klartext und Chiffrat.
[57] eine Variante des DES, bei dem mit 168-Bit-Schlüsseln gearbeitet wird.

Sie schreiben, daß die Benutzung von einfachem DES immer noch besser wäre, als Datagramme unverschlüsselt zu übertragen, was jedoch nur dann richtig ist, wenn sich die Benutzer nicht uneingeschränkt auf den leicht zu brechenden Algorithmus verlassen.

5 Literatur- und Quellenverzeichnis

5.1 Printmedien

[Bohn98] T. Bohnenkamp. „Sicherheit auf der Anwendungsschicht: e-mail". *Seminarband Sicherheit im Internet*, Universität Dortmund, 1998.

[Bell95] S.M. Bellovon. Presentation at IP Security Working Group Meeting. Proceedings of the 32nd Internet Engeneering Task Force, March 1995, IETF, Danvers, MA.

[DiHe76] W. Diffie; M. Hellman. „New Directions in Cryptography". *IEEE Transactions on Information Theory*, Vol. IT-22, No. 6, November 1976.

[Hose96] F. Hosenfeld. „Next Generation". *magazin für computer technik*. Heinz Heise Verlag, Hannover, Nr. 11, 1996.

[Mart98] W. Martens. „Sicherheit auf der Anwendungsschicht: WWW". *Seminarband Sicherheit im Internet*, Universität Dortmund, 1998.

[NeSc78] R.M. Needham; M.D. Schroeder. „Using Encryption for Authentication in Large Networks of Computers". *Communications of the ACM*, Vol. 21, No. 12, December 1978.

[NeSc81] R.M. Needham; M.D. Schroeder. „Authentication Revisited". *ACM Operating Systems Review*, Vol. 21, No. 1, 1981.

[Schn94] B. Schneider. *Applied Cryptography*. John Wiley & Sons. New York. 1994

[Schr98] O. Schröder. „Grundlagen des Internets und des TCP/IP Stacks". *Seminarband Sicherheit im Internet*, Universität Dortmund, 1998.

[VoKe83] V.L. Voydock; S.T. Kent. „Security Mechanisms in High-level Networks". *ACM Computing Surveys*, Vol. 15, No. 2, June 1983.

[vOWi94] P.C. van Oorschot; M.J. Wiener. "Parallel Collision Search with Applications to Hash Functions and Discrete Logarithms", Proceedings of the 2nd ACM Conf. Computer and Communications Security, Fairfax, VA, November 1994.

[Wien94] Wiener, M.J., "Efficient DES Key Search", School of Computer Science, Carleton University, Ottawa, Canada, TR-244, May 1994.

5.2 Elektronische Dokumente

[Atki95a] R. Atkinson. „Security Architecture for the Internet Protocol". *Request for Comments: 1825*. 1995

[Atki95b] R. Atkinson. „IP Authentication Header“. *Request for Comments: 1826*. 1995

[Atki95c] R. Atkinson. „IP Encapsulating Security Payload (ESP)“. *Request for Comments: 1827*. 1995

[DeHi95] S. Deering; R. Hinden. „Internet Protocol, Version 6 (IPv6) Specification“. *Request for Comments: 1883*. 1995

[KMS95] P. Karn; S. Metzger; W. Simpson. „The ESP DES-CBC Transform“. *Request for Comments: 1829*. 1995

[KoNe93] J. Kohl; B. Neuman. „The Kerberos Network Authentication Service (V5)“. *Request for Comments: 1510*. 1993

[MeSi95] S. Metzger; W. Simpson. „IP Authentication with Keyed MD5“. *Request for Comments: 1828*. 1995

Sicherheit auf der Transport- und Sitzungsschicht

Tim Bahnes

1 Einleitung

Über Sicherheit auf der Transportschicht wird schon seit einigen Jahren nachgedacht und so haben sich einige Lösungen entwickelt, von denen jedoch bislang nur wenige zu einem Standard geworden sind. Das wohl bekannteste Protokoll ist das *Secure Socket Layer* (SSL) Protokoll von Netscape. Durch die Implementierung in den weit verbreiteten Netscape Produkten *Navigator* und *Enterprise Server* wurde es sehr schnell als Standard anerkannt. Microsoft erweiterte das Protokoll um einige Features und versuchte, das so entstandene *Private Communication Technology* (PCT) Protokoll durch Implementierung im Internet Explorer und den Web Server Produkten von Microsoft zu etablieren, bislang jedoch nur mit wenig Erfolg. Das Protokoll ist dabei so sehr mit SSL 2.0 verknüpft, daß bei der Verwendung von PCT Lizenzgebühren an Netscape zu entrichten sind. Weitere Protokolle sind das *Simple Key Exchange for Internet Protocol* (SKIP), das ausschließlich für UDP Verbindungen konzipiert wurde, sowie das *Photuris Protocol* und das *Internet Security Association and Key Management Protocol* (ISAKMP). Im folgenden werden das SSL und das darauf aufbauende *Transport Layer Security* (TLS) Protokoll der *Internet Engeneering Task Force* (IETF) und das Protokoll der *Secure Shell* (SSH) detailierter erläutert.

2 SSL - Secure Socket Layer

Das SSL Protokoll wurde erstmals 1994 von Netscape vorgestellt. Der erste Internet Draft für die Version 2.0 [Hick95] wurde 1995 von der IETF veröffentlicht. Da das Protokoll noch Schwächen hat, wird es kontinuierlich weiterentwickelt. Im März 1996 erschien die z.Zt. aktuelle Version 3.0 [Frei96]. Das Protokoll soll vor allem im Bereich des *Electronic Commerce*, also bei Zahlungsvorgängen im Internet, eingesetzt werden. Es wurde entwickelt, um erstens eine sichere Verbindung zwischen zwei Applikationen (Client und Server) zu gewährleisten, und zweitens, um den Server und optional auch den Client zu authentifizieren.

SSL wurde so konzipiert, daß es jedes Transportprotokoll um ein Konzept für einen sicheren Kanal erweitern kann. Abb. 1 zeigt, an welcher Stelle der Netzwerkschicht das SSL Protokoll aufsetzt, wenn es im Internet eingesetzt wird.

Da das SSL Protokoll unabhängig vom Protokoll der Applikation ist, können alle URL konformen Protokolle (z.B. HTTP, FTP, SMTP, NNTP oder Telnet) verschlüsselt werden, ohne daß diese Protokolle modifiziert werden müssen.

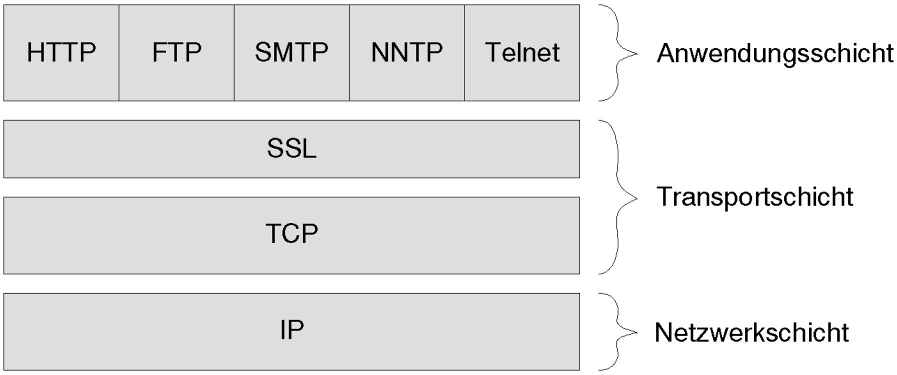

Abb. 1: Das SSL Protokoll im Schichtenmodell des TCP/IP Protokolls

2.1 Sitzungs- und Verbindungszustände

Eine SSL Sitzung kann mehrere sichere Verbindungen enthalten. Der Zustand einer Sitzung wird dabei durch eine Sitzungs Id, ein Zertifikat, eine Kompressionsmethode, einen Verschlüsselungsalgorithmus, einen Hauptschlüssel und ein *Flag*, das anzeigt, ob die Sitzung wiederhergestellt werden kann, beschrieben. Eine Verbindung wird auf der Client- und Serverseite jeweils durch eine zufällige Bytefolge, einen Schlüssel für den *Message Authentication Code* (MAC) und einen Schlüssel für die Verschlüsselung der gesendeten Daten, einen Initialisierungsvektor für Blockverschlüsselungsverfahren und eine fortlaufende Nummer beschrieben.

2.2 Record Layer

Bei dem *Record Layer* handelt es sich um die Schicht des SSL Protokolls, die von höheren Schichten Daten in Form von nicht-leeren Blöcken beliebiger Größe erhält. Diese Informationen werden in sog. *SSLPlaintext Records* von höchsten 2^{14} Bytes fragmentiert. Dabei hat ein *Record* die in Abb. 2 dargestellte Struktur.

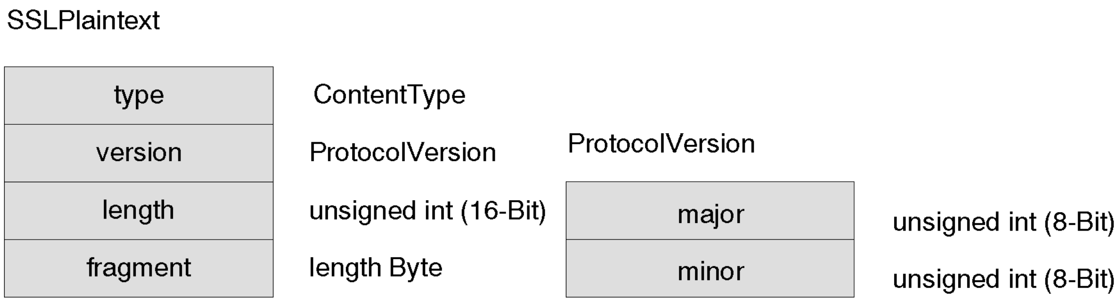

Abb. 2: Unverschlüsselte Records

type — Der Typ des Protokolls, das verwendet wird um das Fragment zu verarbeiten. Hier wird unterschieden zwischen der **change cipher spec** Nachricht[58], einer Fehlermeldung (**alert**), Nachrichten des *Handshake* Protokolls oder Applikationsdaten.

version — Die Versionsnummer der verwendeten SSL Protokolls.

length — Die Länge des Fragments in Byte.

fragment — Die Applikationsdaten. Die Daten werden vom Protokoll nicht interpretiert und erst von einem höheren Protokoll ausgewertet.

SSLCompressed

type	ContentType
version	ProtocolVersion
length	unsigned int (16-Bit)
fragment	length Byte

Abb. 3: Komprimierte Records

Komprimierte Records (Abb. 3) unterscheiden sich von unkomprimierten dadurch, daß die Länge eines Fragments höchstens 2^{14} + 1024 Bytes betragen darf. Der verwendete Kompressionsalgorithmus darf die Daten also auch im ungünstigsten Fall nicht zu sehr aufblähen. Es kann als Kompressionsmethode auch die Identitätsfunktion gewählt werden, die Daten werden dann unkomprimiert versendet.

Bei verschlüsselten *Records* (Abb. 4) darf die Länge eines Fragments 2^{14} + 2048 Bytes nicht überschreiten. Ein verschlüsselter *Record* wird immer aus einem komprimierten *Record* erzeugt.

SSLCiphertext

type	ContentType
version	ProtocolVersion
length	unsigned int (16-Bit)
fragment	GenericStreamCipher/ GenericBlockCipher

Abb. 4: Verschlüsselte Records

[58] Die **change cipher spec** Nachricht benachrichtigt den Empfänger über Änderungen bei der Verschlüsselungsmethode.

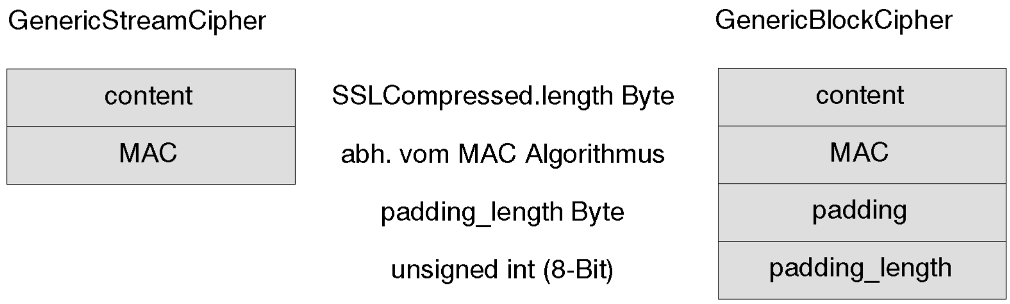

Abb. 5: Die Strukturen der Fragmente bei strom- bzw. blockorientierten Verschlüsselungsverfahren

Für die Verschlüsselung können sowohl block- als auch stromorientierte Verfahren zum Einsatz kommen. Bei beiden Verfahren wird vor der Verschlüsselung aus den Daten und einigen Parametern ein MAC gebildet, um die Integrität sicherzustellen. Dabei wird unter anderem eine fortlaufende Nummer in den MAC eincodiert, so daß fehlende, veränderte oder doppelte *Records* erkannt werden. Bei den blockorientieren Verfahren wird noch ein sog. *Padding* verwendet, das zum Auffüllen der Daten auf die für das Verschlüsselungsverfahren gültige Blockgröße dient.

2.3 Verbindungsaufbau

Bevor eine SSL Verbindung aufgebaut werden kann, müssen sich Client und Server auf die Protokollversion und das Verschlüsselungsverfahren einigen, optional kann auch noch eine Authentifizierung von Client und Server stattfinden. Dieser Vorgang wird durch das SSL *Handshake* Protokoll abgewickelt (Abb. 6).

Zu Beginn des *Handshake* Protokolls eröffnet der Client mit einer **client hello** Nachricht, auf die der Server mit der **server hello** Nachricht antworten muß, andernfalls schlägt der Verbindungsaufbau fehl. Die beiden Nachrichten haben die Aufgabe für die Festlegung der folgenden Attribute zu sorgen: Protokollversion, *Session ID*, Verschlüsselungsverfahren und Kompressionsmethode.

Wenn der Server authentifiziert werden muß, wird nun ein Zertifikat vom Server an den Client gesendet. Hat der Server kein Zertifikat, oder gilt dies nur als Signatur, so wird die Nachricht **server key exchange** gesendet. Ist der Server authentifiziert, kann er vom Client ebenfalls ein Zertifikat anfordern (**certificate request**), wenn das Verschlüsselungsverfahren dies zuläßt.

Nun sendet der Server die **server done** Nachricht und wartet auf die Antworten des Client.

Wurde vom Server ein Zertifikat vom Client angefordert, so sendet dieser das Zertifikat in der **certificate** Nachricht oder übermittelt die Fehlermeldung **no certificate**. Im Anschluß wird die Nachricht **client key exchange** gesendet, deren Inhalt vom vereinbarten Verschlüsselungsverfahren abhängig ist. Wenn der Client ein Zertifikat gesendet hat, das Signaturen unterstützt, so wird nun noch die digital signierte Nachricht **certificate verify** gesendet, um das Zertifikat explizit zu überprüfen.

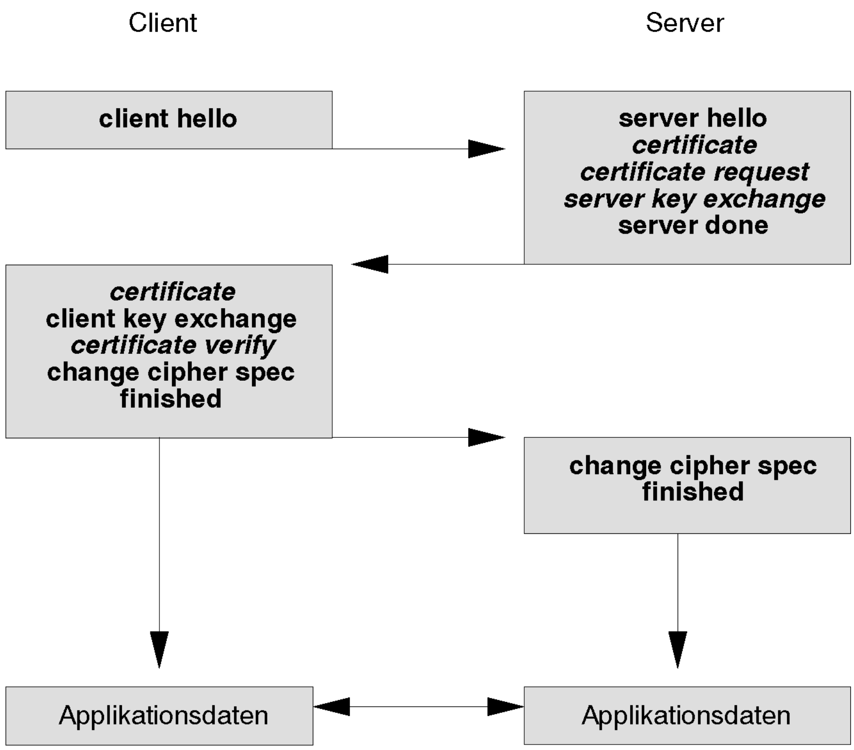

Abb. 6: Das SSL *Handshake* Protokoll

2.3.1 Das Handshake Protokoll

Das *Handshake* Protokoll verwendet die oben beschriebenen Nachrichten. Dabei ist darauf zu achten, daß die Nachrichten in einer genau festgelegten Reihenfolge gesendet werden müssen, ansonsten wird der Verbindungsaufbau abgebrochen. Die Nachrichten werden in einem *SSLPlaintext Record* versendet. Im folgenden werden die Nachrichten im Detail beschrieben. Die Reihenfolge entspricht der vom SSL Protokoll geforderten Reihenfolge.

Hello request

Die **hello request** Nachricht kann jederzeit vom Server gesendet werden, um den Client aufzufordern die **client hello** Nachricht zu senden. Hat der Client das Handshake Verfahren bereits begonnen, so ignoriert er diese Nachricht einfach.

Client hello

Die **client hello** Nachricht wird vom Client gesendet, wenn eine Verbindung aufgebaut werden soll, oder wenn der Client eine **hello request** Nachricht erhalten hat. Die Nachricht hat die in Abb. 7 dargestellte Struktur.

client_version Die Version des Protokolls, das der Client verwenden möchte. Nach Möglichkeit sollte dies die aktuellste Version (z.Zt. 3.0) sein.

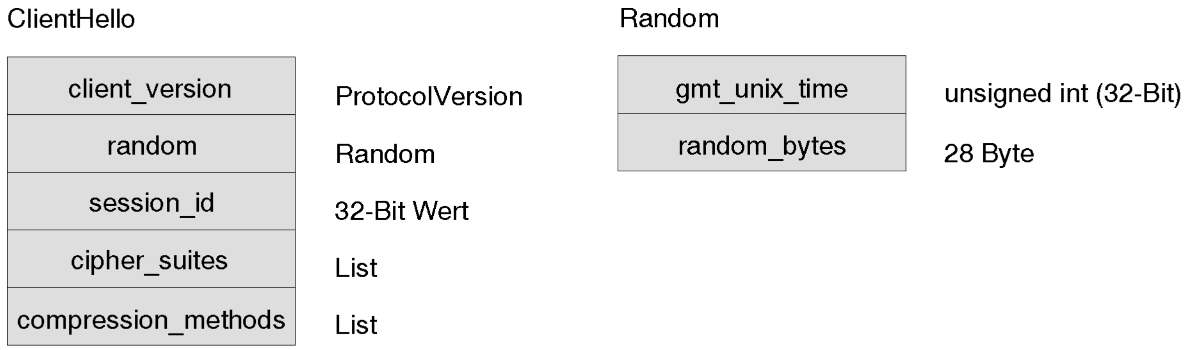

Abb. 7: Die Struktur der **client hello** Nachricht

random Eine Folge von 28 Bytes deren Werte durch einen sicheren Zufallszahlengenerator erzeugt wurden. Zusätzlich wird noch Zeit und Datum im Standard UNIX 32-bit Format übermittelt, die vom Protokoll jedoch nicht benötigt wird, aber evtl. von höherliegenden Schichten ausgewertet wird.

session_id Wird eine *Session ID* angegeben, so müssen die Sitzungsparameter nicht alle erneut angegeben werden, sondern werden wiederverwendet. Dabei kann es sich um Werte einer früheren Verbindung, der aktuellen Verbindung oder einer anderen, gerade aktiven Verbindung handeln. Der Ablauf des *Handshake* Protokolls wird dadurch erheblich verkürzt, wie in Abb. 8 zu sehen ist.

cipher_suites Verschlüsselung. Es werden ein Algorithmus für den Schlüsselaustausch und ein Verschlüsselungsverfahren vereinbart. Es sind auch mehrere Angaben, sortiert nach den Präferenzen des Clients, möglich.

compression_methods Es wird eine Liste von Kompressionsverfahren übermittelt, die der Client unterstützt.

Nach dem Absenden von **client hello** wartet der Client auf die **server hello** Nachricht des Servers.

Server hello

Der Server antwortet auf **client hello** mit einer **handshake failure** Fehlermeldung oder mit der Nachricht **server hello.** Die Struktur der Nachricht entspricht der der **client hello** Nachricht aus Abb. 7.

server_version Es wird die höchste vom Server unterstütze Version zurückgeliefert, die auch vom Client noch unterstützt wird.

random Wie beim Client eine Folge von 28 Zufallsbytes, die jedoch von denen des Clients unabhängig sein muß.

session_id Hat der Client eine *Session ID* angegeben, so überprüft der Server, ob im *Session Cache* diese Sitzung gespeichert ist. Wird die ID gefunden und der Server ist bereit, diese Sitzung zu akzeptieren, so wird dieselbe *Session ID* wie vom Client angefordert zurückgegeben. Beide Seiten müssen daraufhin

	mit der Nachricht **finished** fortfahren. Das verkürzte *Handshake* Verfahren ist in Abb. 8 zu sehen. Wird die Session ID nicht akzeptiert, oder die Session ID des Clients war leer, so wird eine neue ID für die aktuelle Sitzung übermittelt. Wird eine leere ID zurückgegeben, so bedeutet dies, daß die Sitzung nicht gecached wird und somit nicht wiederhergestellt werden kann.
cipher_suite	Das zu verwendende Verschlüsselungsverfahren.
compression_method	Die zu verwendende Kompressionsmethode.

Server certificate

Wenn sich der Server authentifizieren soll, was normalerweise der Fall ist, so sendet er sein Zertifikat direkt im Anschluß an die **server hello** Nachricht. Der Typ des Zertifikats muß für das vereinbarte Verschlüsselungsverfahren geeignet sein. Üblicherweise handelt es sich dabei um ein X.509.v3 oder ein modifiziertes X.509 Zertifikat, wenn *Fortezza*[59] verwendet wird.

Server key exchange

Abhängig vom verwendeten Verschlüsselungsverfahren übermittelt der Server mit dieser Nachricht die Parameter für den Schlüsselaustausch. Zusätzlich zu den Parametern wird noch ein Hashwert gebildet, und zwar aus den Zufallswerten von Client und Server, sowie den Parametern. Dieser Hashwert wird mittels MD5[60] oder SHA[61] Hashverfahren berechnet.

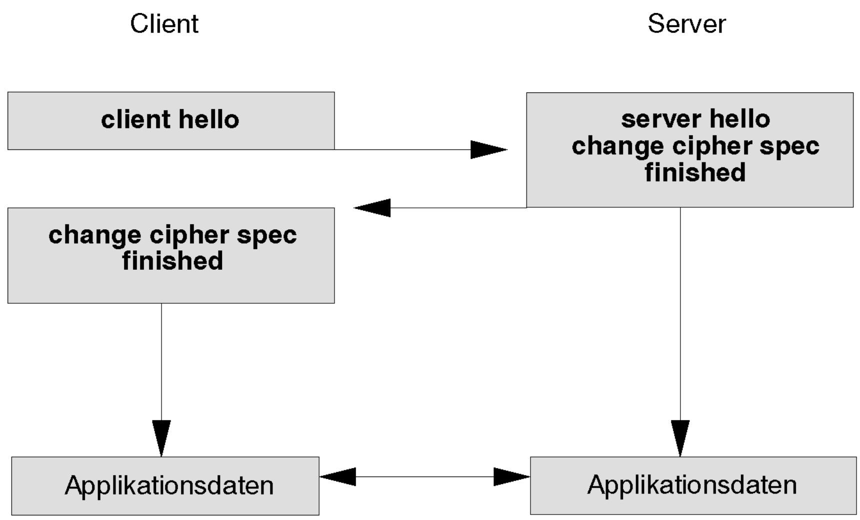

Abb. 8: SSL *Handshake* Protokoll bei Verwendung bereits benutzter Sitzungsparameter

[59] Eine PCMCIA Karte für Verschlüsselung und digitale Signaturen.
[60] Eine sichere Hash-Funktion, die einen Strom von Daten in einem Wert von festgelegter Größe zusammenfaßt.
[61] Ein sicherer Hash-Algorithmus der eine 20-Byte lange Ausgabe erzeugt.

Certificate request

Wenn es das Verschlüsselungsverfahren erlaubt, kann ein nicht anonymer Server ein Zertifikat vom Client anfordern. Dazu übermittelt der Server dem Client eine Liste von Zertifikatstypen und -autoritäten.

Server hello done

Der Server sendet diese Nachricht am Ende des **server hello** Teils des *Handshake* Protokolls. Danach wartet der Server auf die Antwort des Clients.

Client certificate

Dies ist die erste Nachricht die ein Client nach Erhalt von **server hello done** senden kann. Sie wird nur versendet, wenn der Server ein Zertifikat angefordert hat. Verfügt der Client über kein Zertifikat, so weist er den Server über die Meldung **no certificate** darauf hin, was der Server mit einem Abbruch des *Handshake* Protokolls beantworten kann. Die Struktur der Nachricht entspricht der beim Server (siehe Abschnitt Server key exchange).

Client key exchange

Diese Nachricht ist abhängig vom gewählten *Public-Key* Algorithmus.

Certificate verify

Mit dieser Nachricht verifiziert der Client sein Zertifikat, wenn dies Signaturen unterstützt.

Finished

Eine **finished** Nachricht wird immer direkt nach einer **change cipher spec** Nachricht gesendet, um sicherzustellen, daß der Schlüsselaustausch und der Authentifizierungsprozeß erfolgreich waren. Direkt nachdem diese Nachricht versendet wurde, kann mit dem Versenden von Applikationsdaten begonnen werden.
Die **finished** Nachricht besteht nur aus einem MD5 und einem SHA Hashwert, der aus allen Nachrichten, angefangen mit **client hello**, dem Absender und dem Hauptschlüssel berechnet wird. Durch diese Werte kann der Empfänger die Integrität des *Handshake* Protokolls verifizieren.

2.4 Software mit SSL Unterstützung

2.4.1 Netscape

Die große Verbreitung des *Navigators* hat stark dazu beigetragen, daß das SSL Protokoll so schnell zu einem Standard für sichere Verbindungen wurde. Der Navigator war der erste *Web-Browser*, der das SSL Protokoll zur Verschlüsselung von HTTP Sitzungen verwendete. Auch auf der Serverseite bietet Netscape mit dem *Enterprise Server* ein Produkt, daß das SSL Protokoll beherrscht.

Wer eigene Software mit SSL Unterstützung entwickeln möchte, dem bietet Netscape eine Referenzimplementierung des SSL Protokolls an (*SSLRef* 3.0). Dabei handelt es sich um eine Bibliothek, die alle Funktionen enthält, die für das Verwenden des Protokolls nötig sind. Die Bibliothek kann aus dem Internet geladen werden und darf in nicht-kommerziellen Anwendungen kostenlos verwendet werden. Soll SSLRef in kommerziellen Produkten eingesetzt werden, so sind dafür Lizenzgebühren an Netscape abzuführen. Diese belaufen sich einmalig auf 30.000 US Dollar.

2.4.2 Apache

Der frei verfügbare *Apache* Webserver wurde von C2NET unter anderem mit einer SSL Implementierung ausgestattet und wird nun unter dem Namen *Stronghold Web Server* vertrieben.
Der Server ist für fast alle UNIX Plattformen verfügbar und wird in Kürze auch für Windows NT erhältlich sein. Der Preis beläuft sich auf 995 US Dollar.

2.4.3 SSLeay und SSLapps

Bei *SSLeay*[62] handelt es sich um eine freie Implementierung des SSL Protokolls. Da die Implementierung nur mit Hilfe der öffentlich verfügbaren Dokumentation von Netscape erstellt wurde, fallen keine Lizenzgebühren an, die Verwendung in eignen Produkten, auch kommerziellen, ist kostenlos. Erhältlich ist die Bibliothek mit Dokumentation bei [URL-1].
Mit Hilfe der Bibliothek SSLeay wurden viele Applikationen angepaßt, so daß sie nun mit Hilfe des SSL Protokolls eine sichere Verbindung ermöglichen. Folgende Applikationen (die sog. *SSLapps*) sind z.Zt. verfügbar, evtl. aber noch im Alphastadium: telnet (Client und Server), ftp (Client und Server), die Webbrowser NSCA Mosaic und Lynx, die Web Server NSCA httpd, Apache und CERN (W3C) httpd, sowie die Datenbankschnittstelle mSQL. Die Bibliothek wurde an eine Vielzahl von Plattformen angepaßt, darunter sind diverse UNIX Plattformen, DOS, Windows (16- und 32-bit Versionen) und VMS.
SSLeay unterstützt die folgenden Verschlüsselungsalgorithmen: DES, RSA, RC4, IDEA und Blowfish. Bei der Verwendung von einigen bestimmten Verfahren, sollte jedoch vorher der lizenzrechtliche Status geprüft werden.

2.5 Schwachstellen

Prinzipiell ist das SSL Protokoll höchstens so sicher wie das verwendete Verschlüsselungsverfahren. So implementiert Netscape aufgrund der US amerikanischen Exportbestimmungen in seinen Produkten, die für den Export bestimmt sind, nur Verschlüsselungsverfahren, die mit 40-bit langen Schlüsseln arbeiten. Das führt dazu, daß diese Schlüssel mit Hilfe von *Brute Force* Verfahren entschlüsselt werden können. So wurde im Juli 1995 von Hal Finney eine SSL Sitzung veröffentlicht und

[62] Der Name von SSLeay entstand aus den Initialien des Programmierers Eric A. Young.

die Internetgemeinde aufgefordert, diese Sitzung zu entschlüsseln. Der Schlüssel wurde von Damien Doligez unter Zuhilfenahme von insgesamt 112 Computern geknackt. Eine weitere Gruppe hatte per Internet aufgefordert, Rechenzeit zu spenden, was dazu führte, daß der Schlüssel innerhalb von 32 Stunden gefunden wurde. Netscape betonte daraufhin jedoch, daß damit nur ein einzelnes Paket entschlüsselt wurde, jedes weitere würde dieselbe Rechenzeit erfordern. Die Kosten für diese Rechenleistung belaufen sich für normale Benutzer auf ca. 10.000 US Dollar (Quelle: [CT1095]). Fehler bei der Implementierung können dieses Problem jedoch noch verstärken. So wurde ebenfalls 1995 entdeckt, daß der Zufallszahlengenerator im Netscape Navigator 1.1 nicht sicher war. Die Generierung des Schlüssels war von der Systemzeit und der Prozeß ID abhängig. Da beide Werte bekannt, bzw. ungefähr geschätzt werden konnten, war es möglich, den Bereich der Möglichkeiten auf 2^{18} zu begrenzen und so einen Schlüssel innerhalb von 25 Sekunden zu berechnen.
Da das SSL Protokoll in der Version 2.0 noch einige Schwächen hatte, wäre ein Angriff denkbar, der einen Server und Client mit Protokoll 3.0 veranlaßt, nur die Version 2.0 zu verwenden. Dies wird jedoch verhindert, indem Clients, die sich im 2.0 Kompatibilitätsmodus befinden, spezielle Werte an die *Paddings* (siehe Kapitel 2.2) anhängen. Wenn der Server die *Paddings* auspackt und auf diese Werte stößt, meldet er einen Fehler. Beherrschen Client und Server also die neueste Protokollversion, so muß diese auch zwingend verwendet werden.
Weiterhin wäre es denkbar, daß ein Angreifer versucht, während der *Handshake* Phase Client und Server zu veranlassen, ein 40-bit Verschlüsselungsverfahren, oder sogar gar keine Verschlüsselung (falls dies implementiert wurde) zu verwenden. Dazu müßte der Angreifer Nachrichten des *Handshake* Protokolls verändern. Da jedoch für die **finished** Nachricht Hashwerte generiert werden, die alle Nachrichten des *Handshake* Protokolls beinhalten, würden veränderte Nachrichten bemerkt. Ein Angreifer müßte also nicht nur die Nachrichten, sondern auch den Hashwert verändern, was ohne Kenntnis des Hauptschlüssel nicht möglich ist.

3 TLS - Transport Layer Security

Zur Zeit versucht man bei der IETF, vorhandene Technologien, die als Internet Draft vorliegen und sichere Verbindungen auf der Transportschicht ermöglichen, zu verbinden. Das Ergebnis ist der Internet Draft *TLS Version 1.0* [DiAl97]. Dabei basiert TLS größtenteils auf dem SSL Protokoll, die notwendigen Modifikationen sind in [Dier97] beschrieben.
Es wird vorgeschlagen, daß alle Werte, die die Funktionalität des Protokolls betreffen, durch einen MAC vor Veränderungen geschützt werden. SSL überträgt beispielsweise die Protokollversion, ohne sie in den MAC der Nachricht einzubeziehen, eine Veränderung der Versionsnummer könnte jedoch Sicherheitslöcher öffnen.
Einige weitere Änderungen betreffen die Nachrichten. So soll die **no certificate** Nachricht entfallen und statt dessen eine leere **certificate** Nachricht übertragen werden. Weiterhin werden eine Vielzahl

zusätzlicher Nachrichten vorgeschlagen, die hauptsächlich die Ursache eines fatalen Fehlers näher spezifizieren.
Bis zum jetzigen Zeitpunkt liegen jedoch noch keine Implementierungen des TLS Protokolls vor.

4 SSH - Secure Shell

Die SSH ist ein Programm, mit dem man sich über ein Netzwerk auf einem anderen Rechner einwählen und dort Kommandos ausführen, oder Files von einem Rechner zu einem anderen übertragen kann. Damit stellt es eine sichere Alternative zu den von UNIX bekannten Programmen `rsh` (*Remote Shell*), `rlogin` (*Remote Login*) und `rcp` (*Remote Copy*) zur Verfügung. Das zugrundeliegende Protokoll wird in dem Internet Draft [Ylon96] beschrieben. Dabei handelt es sich um ein sicheres Transportschichtprotokoll, daß starke Verschlüsselung, kryptografische Rechnerauthentifikation und Integritätssicherung bietet. Das Protokoll ist unabhängig von der Implementierung der eigentlichen Applikation, also den Programmen `ssh`, `sshd` und `scp`.

4.1 SSH Transport Layer Protokoll

4.1.1 Verbindungsaufbau

Das SSH Protokoll setzt auf einem beliebigen Transportprotokoll auf. Der Client initiiert die Verbindung und erzeugt einen Transportkanal für Binärdaten. Bei der Verwendung des TCP/IP Protokolls lauscht der Server normalerweise am Port 22.
Wurde die Verbindung aufgebaut, senden beide Seiten einen Identifikationsstring der die Version des Protokolls spezifiziert. Direkt nach dem Senden des Identifikationsstrings, beginnt der Schlüsselaustausch, bereits auf der Basis des *Binary Packet* Protokolls.

4.1.2 Binary Packet Protokoll

Ein Binärpaket hat folgende Struktur:

Länge	Die Länge des Pakets. Die Länge enthält die Anzahl der Bytes, die diesem Wert folgen, inklusive eines optionalen MACs.
Länge des *Paddings*	Die Länge des *Paddings* in Byte.
Daten	Die Daten des Pakets.
Padding	Ein Padding füllt das *Paket* auf eine Länge auf, die ein Vielfaches von 8 Byte ist. Diese Regel ist auch bei der Verwendung von stromorientierten Verschlüsselungsverfahren einzuhalten. Es wird empfohlen, daß mindestens vier Bytes mit zufälligem Inhalt für das *Padding* verwendet werden.
MAC	Ein optionaler MAC.

Das Paket wird komplett verschlüsselt, auch die Längenangabe des Pakets. Bei der Implementierung sollte deshalb die Länge nach Empfang der ersten 8 Bytes bestimmt werden, da ein Paket immer mindestens 8 Zeichen lang sein muß. Die maximalen Länge eines Pakets darf 32768 nicht überschreiten, damit das Protokoll auch auf 16-Bit Plattformen implementiert werden kann.

4.1.3 Kompression

Der Kompressionsalgorithmus komprimiert die Daten eines Pakets, dabei darf die Größe des Pakets nicht über die maximal mögliche Größe hinaus anwachsen. Es wird entweder gar nicht oder mit Hilfe des ZIP Kompressionsalgorithmus komprimiert.

4.1.4 Verschlüsselung

Das Verschlüsselungsverfahren und der Schlüssel werden während des Schlüsselaustausches festgelegt. Es wird immer das komplette Paket verschlüsselt. Das Verschlüsselungsverfahren ist unabhängig von der Richtung, d.h., das normalerweise der Schlüssel für jede Richtung verschieden ist, aber auch in jeder Richtung ein anderes Verschlüsselungsverfahren verwendet werden kann.
Es werden die Verschlüsselungsverfahren IDEA, 3DES, DES im CBC Modus, ARCFOUR (RC4 kompatibel) oder die unverschlüsselte Übertragung unterstützt.

4.1.5 Datenintegrität

Um die Datenintegrität zu gewährleisten, wird ein MAC aus einem gemeinsamen Schlüssel, der Paketnummer und dem Inhalt des Pakets berechnet. Die Paketnummer wird dabei nie gesendet, sie dient nur dazu, sicherzustellen, daß keine Pakete verloren gegangen sind. Der MAC wird unverschlüsselt am Ende des Pakets versendet.
Auch hier gilt wieder, daß in jeder Richtung ein anderer Algorithmus zur Berechnung des MAC verwendet werden kann.
Die Erzeugung eines MAC ist optional, die zur Verfügung stehenden Algorithmen sind HMAC-MD5, HMAC-SHA und MD5.

4.1.6 Schlüsselaustausch

Der Schlüsselaustausch beginnt damit, daß beide Seiten eine Liste der unterstützten Algorithmen senden. Jede Seite hat dabei einen bevorzugten Algorithmus, und es wird angenommen, daß die meisten Implementierungen denselben Algorithmus bervorzugen. War diese Annahme falsch, so wird das erste Datenpaket ignoriert, ein allgemeingültiger Algorithmus wird ausgewählt und die Daten erneut gesendet.

Die Liste der Algorithmen wird in einem normalen Binärpaket versendet. Die Daten haben dabei folgende Struktur:

Identifier	SSH_MSG_KEXINIT
Cookie	*Cookies* sind Zufallswerte, die verwendet werden um Schlüssel abzuleiten, damit keine Seite den Hauptschlüssel der anderen Seite erraten kann.
Schlüsselaustausch-verfahren	Das Schlüsselaustauschverfahren wird wie oben beschrieben festgelegt. Zur Zeit wird ein Schlüsselaustausch nach dem RSA-SHA- oder nach dem Diffie-Hellmann-Verfahren unterstützt.
Host-Key Algorithmen	Liste aller *Host-Key* Algorithmen für die der Host gültige Schlüssel hat.
Public-Key Algorithmen	Liste der *Public-Key* Algorithmen die der Host unterstützt.
Verschlüsselung	Eine Liste von Verschlüsselungsverfahren. Es kann für jede Richtung ein anderes Verfahren festgelegt werden.
MAC Algorithmus	Liste der MAC Algorithmen.
Kompression	Liste der Kompressionsverfahren. Soll keine Kompression erfolgen, so muß dies explizit angegeben werden. Auch hier kann für jede Richtung ein anderes Verfahren festgelegt werden.
Hash Algorithmus	Liste mit unterstützten *Hash* Algorithmen.
Flag	Das *Flag* **first key exchange packet follows** zeigt an, daß auf dieses Paket das erste Datenpaket folgt. Wenn nun das SSH_MSG_KEXINIT Paket der anderen Seiten empfangen wird, und das *Flag* gesetzt ist, kann das folgende Datenpaket ignoriert werden, falls das Schlüselaustausch-Verfahren falsch geraten wurde. Auf diese Weise wird der Austausch in den meisten Fällen verkürzt.

4.1.7 Datenaustausch

Die Daten werden asynchron in einem kontinuierlichen Strom übertragen. Vor den eigentlichen Daten wird die ID SSH_MSG_DATA im Paket abgelegt.
Wird die ID SSH_MSG_CLOSE gesendet, so wird die Verbindung beendet und es werden keine Daten mehr akzeptiert.

5 Ausblick

Im Bereich der Sicherheit auf der Transportschicht hat sich das SSL Protokoll etabliert und wird auch in Zukunft einer der wichtigsten Standards bleiben, der von einer Vielzahl von Internet Produkten unterstützt wird. Unterstützt wird diese Entwicklung sicherlich durch die Spezifikation des TLS Protokolls durch die IETF. Da dieses Protokoll auf dem SSL Protokoll basiert, sind bestehende Implementierung leicht anzupassen und werden sicherlich nicht mehr lange auf sich warten lassen.

Das Protokoll der SSH besitzt ähnliche Merkmale wie das SSL Protokoll, wird bislang jedoch nur zur Verschlüsselung von Remote Sitzungen verwendet und hat in diesem Bereich bereits Konkurrenz durch Telnet Implementierungen, die auf das SSL Protokoll aufsetzen. Da die SSH im UNIX Administrationsbereich jedoch bereits eine sehr starke Verbreitung hat, wird sie auch auf lange Sicht der Standard für sichere Remote Sitzungen bleiben.

6 Literatur- und Quellenverzeichnis

6.1 Printmedien

[CT0995] Holger Reif. „Netz ohne Angst – Sicherheitsrisiken im Internet´´. *c´t*, Heise Verlag, Hannover, Nr. 9,1995.

[CT1095] Axel Kossel; Harald Bögeholz. „Netscapes SSL-Protokoll geknackt´´. *c´t*, Heise Verlag, Hannover, Nr. 10, 1995.

[DiAl97] Tim Dierks; Christopher Allen. „The TLS Protocol Version 1.0´´. *Internet Draft*, 1997.

[Dier97] Tim Dierks. „Modifications to the SSL protocol for TLS´´. *Internet Draft*, 1997.

[Frei96] Alan O. Freier; Philip Karlton; Paul C. Kocher. „The SSL Protocol Version 3.0´´. *Internet Draft*, 1996.

[Hick95] Kipp E.B. Hickman. „The SSL Protocol´´. *Internet Draft*, 1995.

[Ylon96] Tatu Ylonen. „SSH Transport Layer Protocol´´. *Internet Draft*, 1996.

6.2 Elektronische Dokumente

[URL-1] SSLeay (gesichtet 14.12.1997):
`http://www.psy.uq.edu.au/~ftp/Crypto/ssleay`

Sicherheit auf der Anwendungsschicht: WWW

Wolfgang Martens

1 Einleitung

Das Thema „Sicherheit auf der Anwendungsschicht“ ist ein Thema mit vielen Problemen in den verschiedensten Bereichen. Jede Anwendung oder Verbindung zwischen Computern ist für Daten eine Gefahr. Viele Dienste und Programme haben Sicherheitslücken, die die Sicherheit der Daten gefährden. Dieser Vortrag zeigt einige Punkte auf und bietet (Teil-) Lösungen an. Zuerst wird das Problem des Schutzes vor Abhören, Unverfälschbarkeit der Daten und der Authentizität des Kommunikationspartners beschrieben und wird das Secure Hypertext Transfer Protokoll (S-HTTP) vorgestellt. Dann werden Sicherheitslücken auf Seiten des Servers und der Clients aufgezeigt und schließlich ein kurzes Resümee gezogen.

2 Secure Hypertext Transfer Protokoll (S-HTTP)

S-HTTP ist eine Erweiterung des Hypertext Transfer Protokoll (HTTP). Es liegt in der aktuellen Version 1.2 vor und soll Daten durch Kryptographie sicher zwischen Client und Server austauschen. Hiermit sollen Angriffe auf Daten während des Transports verhindert werden („*Man in the Middle*“ - Angriffe).

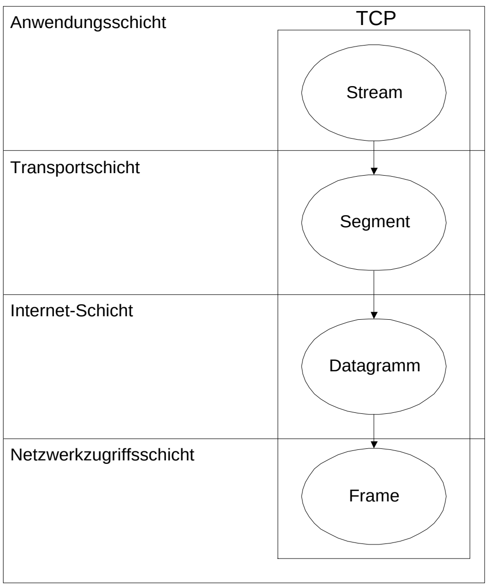

Abb. 1: S-HTTP

2.1 Geschichte

Um die Kommerzialisierung des Internets voran zubringen, wird eine gesicherte Übertragung gefordert um sichere Transaktionen über das WWW abzuwickeln (z.B. *E-Cash*). Hier ist eine wesentliche Voraussetzung , daß die Transaktionsdaten (Käufer - Verkäufer, Konten, Geheimnummern, Bestellungen) geheim bleiben. Weiter müssen die Personenkennungen der beteiligten Personen unverfälschbar sein. Enterprise Integration Technologies (EIT) entwickelte S-HTTP im Jahre 1994 um Sicherheitslücken des HTTP auszugleichen. EIT, RSA Data Security, Inc. und die National Center of Supercomputing Applications (NCSA) programmierten „Secure NCSA Mosaic“ (S-Mosaic) als Referenzimplementierung auf Clientseite. Die Firma Terisa System, Inc. erstellte unter dem Namen „SecureWeb“ einen Server.

2.2 Ablauf

Eine S-HTTP-Nachricht wird über den Klartext, die kryptographischen Präferenzen des Senders und die kryptographischen Präferenzen des Empfängers erstellt. Es wird ein für beide Seiten akzeptabler Algorithmus ausgehandelt und die Nachricht an den Empfänger gesandt, der sie dann entschlüsseln kann.

S-HTTP kennt drei voneinander unabhängige, beliebig kombinierbare Schutzmechanismen: Unterschrift, Authentizität und Verschlüsselung. Dazu werden verschiedene Arten des Schlüsselmanagements unterstützt, wie *Public Keys*, *Kerberos* oder *symmetrische Schlüssel*.

Der Austausch von symmetrischen Schlüsseln geschieht durch eine Nachricht, die durch asymmetrische Schlüssel geschützt sind. Die Authentizität wird über einen Hashwert über die Nachricht und einen vereinbarten Geheimwert geprüft.

2.3 Kryptographische Formate

Hier werden die zwei Formate des S-HTTP vorgestellt.

2.3.1 Public-key Cryptography Standart #7 (PKCS-7)

Das von der Firma RSA definierte Format PKCS-7 stellt eine Obermenge von Privat Enhanced Mail (PEM) dar, kann aber mehr als nur der für ACSII-Dateien definierte PEM - Standard. PKCS-7 ist eine allgemeine Syntax für kryptographisch behandelte Nachrichten und unterstützt 6 verschiedene Inhaltstypen (*Content Types*):

1. Data: normale, unverschlüsselte Daten,
2. EnsignedData: unterschriebene Daten,
3. EnvelopedData: eingepackte Daten,
4. DigestedData: geordnete Daten,
5. SignedAndEnvelopedData: unterschriebene, eingepackte Daten,

6. EncryptedData: verschlüsselte Daten.

2.3.2 MIME (Multipurpose Internet Mail Extensions) Object Security Standard (MOSS)

Im Vergleich zu PEM unterstützt MOSS eine beliebige Anzahl an Algorithmen. Das Prinzip ist, das jede MIME-Nachricht kryptographisch unterschiedlich behandelt werden kann. Daher kann es zwischen verschiedenen MOSS-Applikationen zu Unterschieden kommen, die zum Teil jedoch inkompatibel sind.

2.4 Nachrichtenformat

Wie HTTP beginnt S-HTTP mit einer *Request Line* oder einer *Status Line*, der beliebig viele Header und Inhalte folgen. Der Protokollbezeichner ist mit aktuellen Fall „Secure-HTTP/1.2“.

```
GET /secret HTTP/1.0                                        Request Line
Security-Scheme: S-HTTP/1.2                                 HTTP Header
User-Agent: Web-O-Vision 1.2beta
Accept: *.*
Key-Assign: Inband,1,reply,des-ecb;7878787878787878
```

2.4.1 Die *Request Line*

Sie enthält immer die Methode SECURE und ggf. eine URL, damit Proxy Server die Nachricht an den richtigen Server weiterleiten können. Des weiteren enthält sie die Protokollversion.

2.4.2 Die *Status Line*

Sie ist immer gleich und enthält „Secure-HTTP/1.2 200 OK“, weil keinerlei Informationen über gescheiterte Verbindungen nach außen sichtbar sein sollen.

2.4.3 Die *Secure HTTP Header Lines*

Sie bestehen jeweils aus einem Bezeichner, einem Doppelpunkt und einem Wert. Der Header wird durch zwei CR/LF - Zeichen (Zeilenumbruch) abgeschlossen (wie HTTP).

S-HTTP Headerlines sind

`Content-Privacy-Domain`

Bestimmt das verwendete Format des Austauschs: PKCS-7, MOSS

`Content-Transfer-Encoding`

legt das Übertragungsformat fest. Bei PKCS-7 sind dies BASE64, 8BIT oder BINARY. Bei MOSS sind alle definierten Werte erlaubt.

`Content-Type`

Beschreibt den MIME - Typ der entpackten Nachricht. Normalerweise application/http. Sollte die Nachricht mehrfach gesichert sein, steht hier application/s-http.

Bei MOSS wird hier die Art der kryptographischen Behandlung beschrieben (z.B. signed)

`Prearranged-Key-Info`

Enthält Informationen über die Schlüssel, die vor der Übertragung ausgetauscht wurden. Es gibt drei Möglichkeiten des Austausches: Outband, Inband, Kerberos.

Outband: Die Schlüssel sind aus einer Datenbank oder durch den Benutzer identifizierbar.

Inband und Kerberos: Schlüssel wurden vorher ausgetauscht.

Weiter enthält dieser Parameter die Informationen eines *Data Encrytion Key* (DEK) und eine *CoverKey-ID*.

`MAC-Info`

Stellt die Integrität der Nachricht sicher. In älteren Versionen von S-HTTP wird ein einfacher Hashwert aus Nachricht, optional Zeit und einer gemeinsamer Geheimzahl berechnet. Seit 1.2 wird HMAC verwandt:

HMAC = hash(hash(key) XOR pad_2 + hash(hash(key) XOR pad_1 + time + message

time ist optional, pad_2 und pad_1 sind Werte aus verschiedenen Eingabeblöcken. Key ist die SchlüsselId (CoveKey-ID).

In den Headern werden auch die kryptographischen Details ausgehandelt.

2.5 Neue HTTP-Header

S-HTTP fügt zwei weitere Header ein.

`Security Scheme`

Muß in allen HTTP - Nachrichten integriert werden, um sie als S-HTTP-fähig auszuweisen.

`Nonce-Echo`

Wenn einer der Partner NONCE anfordert erhält der als Antwort den String zurück, der dem Parameter folgt.

2.6 Fehlerfälle

Bei HTTP werden die Informationen bzgl. der Kommunikation in Klassen unterteilt. Im zwei Klassen müssen Änderungen gemacht werden, um S-HTTP sicher ablaufen lassen zu können.

2.6.1 Klasse 4 (Client Fehler)

Die „402“ (Zahlung erhalten) wird eingefügt. Das Zahlungsschema ist in S-HTTP jedoch nicht definiert.

Die „420“ (Sicherheit Neuversuch) zeigt an, daß die übermittelten Parameter vom Server nicht akzeptiert werden konnten oder ein S-HTTP-Request gesendet werden sollte anstatt eines HTTP-Requests. Als Antwort können andere Parameter gesendet werden, neue Schlüssel verhandelt werden oder ein NONCE angefordert werden.

Die „421“ (Header falsch) meldet eine fehlende oder falsche Headerzeile.

2.6.2 Klasse 3 (Umleitungen)

Hier sollten automatische Umleitungen oder automatische Versuche der Wiederholung im Browsern nicht implementiert werden, da sonst ungewollt Sicherheitslücken auftreten können. Zum Beispiel könnte ein Angreifer die Antwort eines Servers abfangen und durch eine mit dem öffentlichen Schlüssel des Clients verschlüsselte 301-Nachricht (Umgezogen) ersetzten, woraufhin der Browser anschließend die Daten des Benutzers an den neuen, aber falschen Server senden würde.

2.7 Servererweiterungen

Damit eine S-HTTP-Verbindung aufgebaut werden kann, wird ein neuer URL-Typ benötigt: shttp://. Des weiteren sollten Server fähig sein, Anforderungen nach Zertifikaten, Policies und *Certificate Revocation Lists* (CRL) zu beantworten bzw. abzufragen. Es gibt einen Request für das Zertifikat (GET/SERVER-CERTIFICATE), einen für die Policy (GET/POLICY) und eine für die CRL der Zertifizierungsstelle (GET/CRL). Alle drei sollen die Authentifizierung der Verbindung garantieren.

2.8 HTML Erweiterungen

Der Tag <A> wird um drei Optionen erweitert:

- DN gibt in der HTML-Seite die Signatur (*Distinguished Name*) des Empfängers bekannt,
- NONCE beinhaltet den String der bei Anfrage zurückgesendet wird (s.o.),
- CRYTOPTS enthält die Liste der Zertifikate, die der Empfänger benötigt um z.B. eine Signatur zu verifizieren.

2.9 Beispiel

Request mit RSA-Schlüsselaustausch und Inband-Schlüssel:
Mit einem S-HTTP-fähigen Webbrowser wird ein HTTP-Request auf eine Seite geschickt, der eine Link auf eine S-HTTP-Seite enthält. Der Browser erhält zusätzlich zu der Seite folgende Antwort :

```
200 OK HTTP/1.0                                                  | Status Line
Server-Name: Navaho-0.1.2.3alpha                                 | Server Name
Certificate-Info: PKCS7,MIAGCSqGSIb3DQEHAqCAMIACAQExADCABgkqh    | Schlüssel
        kiG9w0BBwEAAKCAM
        IIBrTCCAUkCAgC2MA0GCSqGSIb3DQEBAgUAME0xCzAJBgNVBAYTAlVTMSAwH
        gYDVQQKExdSU0EgRGF0YSBTZWN1cml0eSwgSW5jLjEcMBoGA1UECxMTUGVyc
        29uYSBDZXJ0aWZpY2F0ZTAeFw05NDA0MDkwMDUwMzdaFw05NDA4MDIxODM4N
        TdaMGcxCzAJBgNVBAYTAlVTMSAwHgYDVQQKExdSU0EgRGF0YSBTZWN1cml0e
        SwgSW5jLjEcMBoGA1UECxMTUGVyc29uYSBDZXJ0aWZpY2F0ZTEYMBYGA1UEA
        xMPU2V0ZWMgQXN0cm9ub215MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAMy8Q
        cW7RMrB4sTdQ8Nmb2DFmJmkWn+el+NdeamIDElX/qw9mIQu4xNj1FfepfJNx
        zPvA0OtMKhy6+bkrlyMEU8CAwEAATANBgkqhkiG9w0BAQIFAANPAAYn7jDgi
        rhiIL4wnP8nGzUisGSpsFsF4/7z2P2wqne6Qk8Cg/Dstu3RyaN78vAMGP8d8
        2H5+Ndfhi2mRp4YHiGHz0HlK6VbPfnyvS2wdjCCAccwggFRAgUCQAAAFDANB
        gkqhkiG9w0BAQIFADBfMQswCQYDVQQGEwJVUzEgMB4GA1UEChMXUlNBIERhd
```

```
        GEgU2VjdXJpdHksIEluYy4xLjAsBgNVBAsTJUxvdyBBc3N1cmFuY2UgQ2Vyd
        GlmaWNhdGlvbiBBdXRob3JpdHkwHhcNOTQwMTA3MDAwMDAwWhcNOTYwMTA3M
        jM1OTU5WjBNMQswCQYDVQQGEwJVUzEgMB4GA1UEChMXUlNBIERhdGEgU2Vjd
        XJpdHksIEluYy4xHDAaBgNVBAsTE1BlcnNvbmEgQ2VydGlmaWNhdGUwaTANB
        gkqhkiG9w0BAQEFAANYADBVAk4GqghQDa9Xi/2zAdYEqJVIcYhlLN1FpI9tX
        Q1m6zZ39PYXK8Uhoj0Es7kWRv8hC04vqkOKwndWbzVtvoHQOmP8nOkkuBi+A
        QvgFoRcgOUCAwEAATANBgkqhkiG9w0BAQIFAANhAD/5Uo7xDdp49oZm9GoNc
        PhZcW1e+nojLvHXWAU/CBkwfcR+FSf4hQ5eFu1AjYv6Wqf430Xe9Et5+jgnM
        Tiq4LnwgTdA8xQX4elJz9QzQobkE3XVOjVAtCFcmiin80RB8AAAMYAAAAAAA
        AAAAA==
Encryption-Identity: DN-1485, null, CN=Setec Astronomy, OU=Persona    Zertifizie=
Certificate,O=“RSA Data Security, Inc.“, C=US;                         rungsstelle
SHTTP-Privacy-Enhancements: recv-required=encrypt
<A name=tag1 HREF=“shttp://www.setec.com/secret“>                      URL
Don‘t read this. </A>
```

Um diese URL zu referenzieren, wird folgender HTTP-Request verwendet:

```
GET /secret HTTP/1.0                                               Request Line
Security-Scheme: S-HTTP/1.2                                        HTTP Header
User-Agent: Web-O-Vision 1.2beta
Accept: *.*
Key-Assign: Inband,1,reply,des-ecb;7878787878787878
```

Mit der Key-Assign-Headerzeile wird ein gemeinsamer Schlüssel vereinbart, der es dem Server erlaubt, die Antwort an den Client zu verschlüsseln, obwohl der Client selbst keinen öffentlichen Schlüssel besitzt. Nach der S-HTTP-Verarbeitung sieht der eingekapselte Request so aus:

```
Secure * Secure-HTTP/1.2                                          Request Line
Content-Transfer-Encoding: base64                                 Übertragung
Content-Type: application/http                                    Unversch.Format
Content-Privacy-Domain: PKCS-7                                    SchlüsselTyp
-----BEGIN PRIVACY-ENHANCED MESSAGE-----                          Nachricht
MIAGCSqGSIb3DQEHA6CAMIACAQAxgDCBqQIBADBTME0xCzAJBgNVBAYTAlVTMSAw
HgYDVQQKExdSU0EgRGF0YSBTZWN1cml0eSwgSW5jLjEcMBoGA1UECxMTUGVyc29u
YSBDZXJ0aWZpY2F0ZQICALYwDQYJKoZIhvcNAQEBBQAEQCU/R+YCJSUsV6XLilHG
cNVzwqKcWzmT/rZ+duOv8Ggb7oO/d8H3xUVGQ2LsX4kYGq2szwj8Q6eWhsmhf4oz
lvMAADCABgkqhkiG9w0BBwEwEQYFKw4DAgcECFif7BadXlw3oIAEgZBNcMexKe16
+mNxx8YQPukBCL0bWqS86lvws/AgRkKPELmysBi5lco8MBCsWK/fCyrnxIRHs1oK
BXBVlsAhKkkusk1kCf/GbXSAphdSgG+d6LxrNZwHbBFOX6A2hYS63Iczd5bOVDDW
Op2gcgUtMJq6k2LFrs4L7HHqRPPlqNJ6j5mFP4xkzOCNIQynpD1rV6EECMIk/T7k
1JLSAAAAAAAAAAAAAA==
-----END PRIVACY-ENHANCED MESSAGE-----
```

Zwischen den beiden Begrenzern steht eine PKCS-7-Nachricht, die mit RSA unter Benutzung des vorher im Certificate-Info übertragenen öffentlichen Schlüssels der Setec Astronomy verschlüsselt wurde. Auf Serverseite wird die Anfrage entschlüsselt und das Dokument bereitgestellt. Der HTTP-Response sieht so aus:

```
HTTP/1.0 200 OK
Security-Scheme: S-HTTP/1.2
Content-Type: text/html

Congratulations, you‘ve won.
<A href=“/prize.html“
 CRYPTOPTS=“Key-Assign: Inband,alice1,reply,des-ecb;020406080a0c0e0f;
 SHTTP-Privacy-Enhancements: recv-required=auth“>Click here to
claim your prize</A>
```

Eingeschlossen in eine S-HTTP-Nachricht wird dies zu:

```
Secure * Secure-HTTP/1.2
Content-Transfer-Encoding: base64
Content-Type: application/http
Prearranged-Key-Info: des-ecb,697fa820df8a6e53,inband:1
Content-Privacy-Domain: PKCS-7

-----BEGIN PRIVACY-ENHANCED MESSAGE-----
MIAGCSqGSIb3DQEHBqCAMIACAQAwgAYJKoZIhvcNAQcBMBEGBSsOAwIHBAifqtdy
.....
AAAAAAAAAAA=
-----END PRIVACY-ENHANCED MESSAGE-----
```

Die Daten zwischen den Begrenzern sind eine PKCS-7-Nachricht, die mit einem zufällig gewählten Sitzungsschlüssel (DEK) verschlüsselt wurden, der seinerseits mit dem vorher ausgetauschten Schlüssel 1 der Methode inband (inband:1) verschlüsselt ist.

[URL-1]

2.10 Vergleich mit Secure Socket Layer (SSL)

SSL benutzt die gleichen krytographischen Techniken um die Nachrichten zu verschlüsseln. Jedoch verschlüsselt SSL nicht die Nachricht selbst, sondern deren Übertragungskanal. SSL arbeitet auf der Transportschicht.

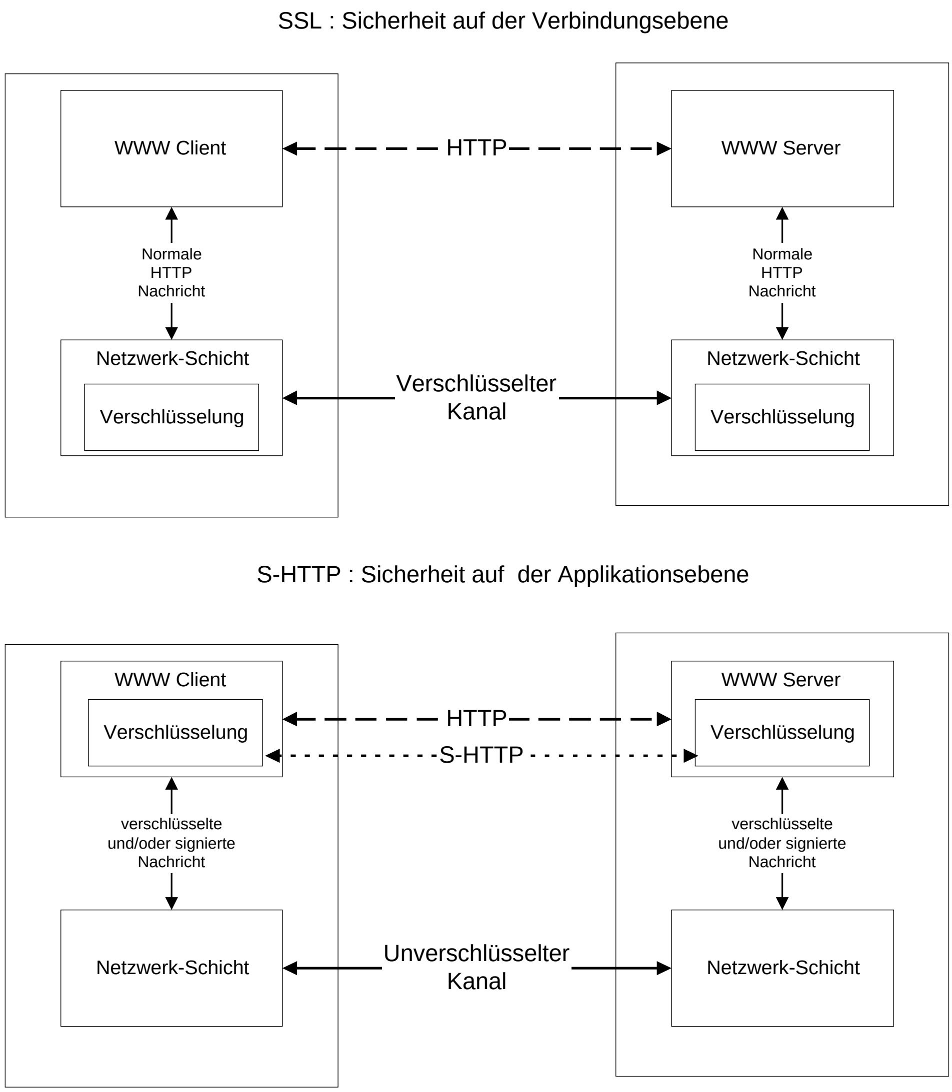

Abb. 2: Unterschied zwischen SSL und S-HTTP

SSL ist für einen allgemeinen Einsatz gedacht, findet jedoch zur Zeit nur Anwendung im WWW, während S-HTTP als Erweiterung des HTTP gedacht ist und daher nur für das WWW angewendet werden kann.

2.11 Zusammenfassung S-HTTP

S-HTTP ist eine Erweiterung des HTTP, die die Sicherheit bei der Übertragung im WWW gewährleisten soll. S-HTTP ist im Gegensatz zu SSL und anderen Verfahren auf der Anwendungsschicht angesiedelt. Jedoch wird S-HTTP nicht von Netscape und Microsoft unterstützt und findet daher kaum Verbreitung. Lediglich Mosaic hat von den großen Anbietern einen S-HTTP-fähigen Browser auf den Markt gebracht (S-MOSAIC).

3 Sicherheitslücken

3.1 Begriffserklärung Spoofing

„In network lingo, spoofing means pretending to be something you are not“[URL-3]
(In der Netzwerksprache heiß Spoofing, daß man etwas vorgibt, das man nicht ist.)

3.2 Arten

Es werden drei Arten von Spoofing unterschieden:

- Mail Spoofing: Hier wird vorgegeben, daß man der Empfänger der Mail von A ist, obwohl man selbst B ist. (z.B. könnten Mail an Helmuth.Kohl@cdu.bundestag.de tatsächlich an Gerd.Schröder@spd.hannover.de gehen)
- IP Spoofing: Hier gibt ein Rechner vor, eine andere Maschine zu sein als sie ist. (s. Vortrag Angriffsstrategien)
- Web Spoofing: Hier gibt ein Server vor, ein anderer zu sein, als er ist. (z.B. könnte java.Sun.com umgelenkt werden nach Microsoft.com....)

Das zuvor vorgestellte S-HTTP hat hier keine Schutzmöglichkeiten, da das Ziel nicht die transferierten Daten selbst sind, sondern der Angriff gegen die Struktur des WWW geht.

4 Attacken durch Anwendungen

Programme schaffen Sicherheitsprobleme, wenn sie über das Netz aufgerufen werden können. (z.B. über Telnet), da sie zumeist eigentlich nur für den lokalen Rechner konzipiert wurden.

4.1 Serverseite

Die Administratoren eines Systems haben mit einer unüberschaubaren Vielzahl dieser Sicherheitslücken der verwendeten Software zu kämpfen. Fast jede Woche kommt die Nachricht über eine Sicherheitslücke. Der Server der CERT (http://www.cert.dfn.de) versendet in Durchschnitt 15 - 20 Mails im Monat.

4.1.1 Einbuch in ein System

Der Einbruchsversuch hat mit Normalfall das Ziel des Erlangens von Rechten auf den Servern. Entweder werden Paßwörter entwendet oder ROOT-Rechte erschlichen. Solches ist mit einfachen Mitteln zu erreichen.
Zumeist wird über Telnet ein lokales Programm ausgeführt und dessen Schwächen mißbraucht(z.B. lpr, dbx).

Andere Wege gehen über einen angehängten Decoder an eine E-Mail. Hier lassen sich von Netscape und Internet Explorer die Mail - Paßworte ausspionieren.
Es gibt im Internet zahlreiche Anleitungen und fertige Programme, die einen Einbruch ermöglichen.

4.1.2 Entdecken des Einbruchs

Das Entdecken ist schwierig, da nicht alle Möglichkeiten eines Einbruchs überwacht werden können. Eine gewisse Sicherheit läßt sich nur dadurch herstellen, daß die wichtigen Dateien regelmäßig geprüft werden, so daß ungewollte Änderungen entdeckt werden können. Das Hauptproblem besteht jedoch nicht in den Einbrüchen aus dem Internet heraus, sondern durch eigene Mitarbeiter.

4.1.3 Schutz vor Einbrüchen

Die einzige Chance besteht darin es einem Eindringling so schwer wie möglich zu machen. Es müssen alle Advisatories (Nachrichten über Schutzfehler in Programmen - CERT) umgesetzt werden. Von außen gibt es technische Mittel wie Firewalls, oder das Verbot, Daten von innen direkt in das Internet zu versenden, indem alles über sogenannte Proxies läuft, die damit zu verhindern, daß lokale Programme Daten unkontrolliert in das WWW versendet.

4.1.4 Beispiele von Hacked Pages

Abb. 3: Grafik einer Hacked Page , welche auf 15 Servern am selben Tag erschien.

Abb. 4: Verändertes Logo von „Lost World“.

4.1.5 Fehler durch Ähnlichkeiten der URL

Da es eine Vielzahl von Top-Level-Domains (.de, .com, .org, .net....) gibt, kann es auch kommen, daß unter der gleichen Domain zwei völlig verschiedene Server zu finden sind. Zum Beispiel hat die NASA als staatliche Organisation die Domain nasa.ORG. Wie es jedoch für Ziele in den USA üblich ist wurde häufig nasa.com angewählt und damit ein existierender, aber nicht gewünschter Server erreicht.

4.1.6 Fehler in Software am Beispiel MS Internet Information Server

Bei diesem Programm, das die Anfragen auf Ausliefern von Seiten verwalten soll, gibt es zwei Sicherheitslücken:

- Über das Integrierte FTP-Programm läßt sich der ROOT-Datei-Baum für Zugriffe öffnen.
- Bei *Active Server Pages* (ASP) ermöglicht ein einziger zusätzlicher Punkt in der URL den vollen Schreib-/Lesezugriff auf diese Seite. Da ASP oft für Authentifizierung genutzt wird sind Paßwörter ausspähbar.

4.2 Clientseite

Auch auf Clientseite ist es möglich Schäden durch Lücken in Programmen zu erzeugen. Hier werden die Lücken genutzt um Daten abzuschöpfen oder zu löschen. Einige Beispiele hierfür sind:
MS Internet Explorer

- Da der Explorer stark in die Windows-Betriebssysteme integriert ist sind hier weitreichende Schwachpunkte bekannt geworden die häufig auf Active-X beruhen, da hier Sicherheitsaspekte nur sehr unzureichend berücksichtigt wurden.

- Ein großes Problem ist die Fähigkeit die *unified name convention* (UNC) - Namen aufzulösen. Bei UNC werden Name und Paßwort an den Server übertragen. Eine Seite im UNC Form ermöglicht das Abgreifen dieser Daten.
- Bei dem Internet Explorer 4 wird eine 265 Zeichen lange URL als Befehl interpretiert. Da bei Windows die Systemprogramme im Normalfall immer an derselben Stelle liegen, lassen sich alle Programme beliebig starten, ohne daß der Benutzer davon Kenntnis erlangt.

Netscape Navigator

- Hier liegen die Schwachpunkte in erster Linie bei den PlugIn's, die Löcher in das doch einigermaßen gute Sicherheitskonzept des Browsers und der Java Virtual Machine reißen.

[URL -4]

4.3 Zusammenfassung Attacken

Im Prinzip ist jeder Rechner, der am Internet hängt, potentiell gefährdet Opfer einer Attacke zu werden. Es ist nur begrenzt möglich, sich vor Schäden durch Einbrüche zu schützen, weil der Angreifer meist ein Schritt voraus ist. Eine gewisse Sicherheit kann nur durch hohen Aufwand der Benutzer gewährleistet werden, daß er sich regelmäßig informiert und Lücken aktiv schließt.

5 Fazit

Das Internet ist noch weit entfernt davon ein sicheres Kommunikationsmittel zu sein. Es gibt erste Ansätze dieses Problem in den Griff zu bekommen Ein wichtiger Schritt dahin ist es, daß sich die Benutzer bewußt sind, daß das was sie abschicken auch von Unbefugten gelesen werden kann. Des weiteren ist kein hundert prozentiger Verlaß darauf, daß das was er bekommt unschädlich ist. Es gibt Wege, die Sicherheit erhöhen, aber auch sie sind nicht absolut sicher vor Mißbrauch und Lücken. Es werden neue Standards für eine sichere Übertragung benötigt (z.B. Internet Protokoll Version 6 - IPv6). Es zwingend notwendig, daß die Betriebssysteme in Hinblick auf die Sicherheit neu überarbeitet werden, um Schwächen von Anwendungen auszugleichen und nur den autorisierten Nutzern Zugriff zu gewähren.

6 Literatur- und Quellenverzeichnis

6.1 Elektronische Dokumente allgemein

[URL-1] Universität Heilbronn, Diplomarbeit Tobias Häcker Stand 31.03.1997 (gesichtet 17.01.1997):
`http://people.swl.fh-heilbronn.de/~tobi/Diplomarbeit/`

[URL-2] University of Illinois, Stand nicht feststellbar (gesichtet 17.01.1998):
http://www.ncsa.uiuc.edu/InformationServers/WebSecurity/iw3_tut/NETSCAP2.HTM

[URL-3] SystemExperts Corporation Stand 06.01.1998 (gesichtet 17.01.1997)
http://www.sys-exp.com/docwebsp.htm

[URL-4] Netzwerk Sicherheits Competence Center, Stand nicht feststellbar (gesichtet 17.01.1998)
http://www.cert.dfn.de/

6.2 Elektronische Dokumente zum Thema S-HTTP (ohne explizite Nennung)

[URL-a] Department of Computer and Systems Sciences, Stand nicht feststellbar (gesichtet 07.01.1998)
http://www.dsv.su.se/~lennarts/shttp.html

[URL-b] Mersch Online, Stand 04.01.1998, (gesichtet 17.01.1997)
http://www.mersch.com/research/xchange/shttp.htm

[URL-c] Duke University, Druham, North Carolina, Stand 15.01.1998 (gesichtet 17.01.1998)
http://www.duke.edu/~wgrobin/ethics/netshop/s-http.htm

[URL-d] Universität Hannover, Stand 13.07.1995 (gesichtet 17.01.1997)
http://www.rvs.uni-hannover.de/arbeiten/studien/sa-lschuette-html/node106.html

6.3 Elektronische Dokumente zum Thema Spoofing

[URL-e] Ohio State University, Stand nicht feststellbar (gesichtet 17.01.1998)
http://www.cis.ohio-state.edu/~drapcho/694qpres-final/sld022.htm

[URL-f] Cisco Connection Online, Stand 1996 (gesichtet 17.01.1997)
http://www.cisco.com/warp/cpropub/59/PPPConfig/BRI_to_IPX_Router.html

[URL-g] CIAC, Stand nicht feststellbar (gesichtet 17.01.1998)
http://ciac.llnl.gov/ciac/bulletins/f-08.shtml

6.4 Elektronische Dokumente zum Thema Sicherheit

[URL-h] W3Org, Stand 03.09.1997 (gesichtet 17.01.1997):
http://www.w3.org/Security/

[URL-i] Concord, Stand nicht feststellbar (gesichtet 17.01.1997):
http://concord.cscdc.be/conference/presentation/3_1200_1_1/index.htm

Sicherheit auf der Anwendungsschicht: e-mail

Torsten Bohnenkamp

1 Einordnung des Vortrags / Begriffsklärung

Dieser Vortrag befaßt sich mit dem Thema Sicherheit bei der Kommunikation im Internet. Dabei werden anhand von PGP und S/MIME die wichtigsten Programme/Verfahren der verschlüsselten Versendung von e-mails vorgestellt. Zunächst soll aber die Motivation für den Aufwand der Anwendung derartiger Software dargestellt werden. Abschließend wird das Online-Echtzeit-Übertragungsverfahren ICQ (Online-Konferenzen / Chatrooms) vorgestellt.

1.1 e-mail

Was ist eine e-mail ?
Häufig wird eine e-mail als elektronische Postkarte verstanden. Diese Betrachtungsweise trägt der tatsächlichen Verwendung aber nicht Rechnung. Betrachtet man den Inhalt, so stellt man fest, daß oftmals sensible Informationen übertragen werden, die nicht ohne weiteres für jeden zugänglich sein sollten (z.B. Firmeninterna, Kreditkartennummern, private Kommentare usw.). Informationen also, die man bestenfalls per Brief, besser noch per Einschreiben, versenden würde. Zudem können ungeschützt übertragene e-mails systematisch und effektiv erfaßt und verändert werden. Schließlich handelt es sich um elektronische Dokumente und nicht um handschriftliche Originale.
Der Kreis der Nutzer von e-mail Systemen hat sich in der jüngeren Vergangenheit schnell erweitert. Von der Universität über den privaten Anwender bis zur zunehmenden Nutzung in der Industrie und in Dienstleistungsunternehmen ist die e-mail ein schneller, komfortabler und vermeindlich sicherer Weg Informationen zu übertragen. Neue Anwendungsmöglichkeiten, wie z.B. ‚online banking' und ‚electronic commerce', führen dazu , daß zunehmend sensible Daten übertragen werden. Daher ist es wichtig den Benutzerkreis über mögliche Manipulation und Mißbrauch zu informieren, sodaß Maßnahmen getroffen werden können, damit sensible Informationen auf sicherem Weg vom Absender zum Adressaten geschickt werden können.

1.1.1 Der Weg einer e-mail

Ein Mail-System kann in die folgenden drei Stufen unterteilt werden [ChZw95]:

- der Mail-Server erhält e-mails von externen hosts bzw. schickt sie an externe hosts
- das Mail-Programm des Providers oder der Domain speichert die e-mails auf lokalen hosts
- das Anwenderprogramm erlaubt das Lesen bzw. Schreiben von e-mails

Die e-mail wird zunächst mit dem Mail-Clienten des Browsers oder der Provider-Software erstellt. Diese Nachricht wird vom Mail-Server des Providers an den Mail-Server des Adressaten geschickt und dort solange in einer Mailbox gespeichert bis der Adressat online geht und die e-mail empfängt. Auf diesem Weg, der über unterschiedliche Netze und Server führt, ist die Nachricht nicht geschützt. Die verwendeten Übertragungsprotokolle wie z.B. SMTP (Simple Mail Transfer Protocol) gewährleisten Datensicherheit, d.h. sie ermöglichen einen verlustfreien Datentransport innerhalb der unterschiedlichen Netzwerke. Aspekte die den Datenschutz betreffen, spielen bei der Datenübertragung eine nur untergeordnete Rolle [Dama97].

1.1.2 Formen der Manipulation

Die Ansatzpunkte für Manipulationen können sehr unterschiedlich sein. Vorstellbar sind Versuche die e-mail Übertragungen abzuhören, indem man sich Zugang zu den Mail-Servern verschafft. Dies kann mit den Zugriffsrechten eines Mail-Administrators oder durch Umgehung von Sicherheits-vorrichtungen, wie z.B. Firewalls, gelingen. Damit hat man Zugang zum kompletten e-mail Verkehr der über den betreffenden Server weitergeleitet wird. Mithilfe einer Adressenmanipulation ist es möglich sich zwischen Sender und Empfänger zu schalten, sodaß man Zugriff auf die e-mails einer bestimmten Person erhält. Eine andere Möglichkeit stellen die sogenannten ‚data driven attacks‘[63] dar. Dabei werden e-mails benutzt um Zugang zur Benutzerkonsole zu erhalten, oder interne Netzwerke mittels Viren zu stören. Grundsätzlich kann man die folgenden Formen der Manipulation unterscheiden:

- Nachricht mitlesen und speichern:
  Die Information wird systematisch erfaßt
- Nachricht verfälschen:
  Der Inhalt wird verändert oder die e-mail erreicht einen anderen Adressaten oder der Adressat wird durch eine falsche Absenderadresse getäuscht.
- Nachricht selektiv löschen / abfangen:
  Nur ausgewählte Nachrichten werden übertragen. Es findet eine sender- oder empfängerseitige Filterung statt.
- Nachricht als nicht autorisierte Zugangsmöglichkeit nutzen:
  Benutzung der e-mail als ‚trojanisches Pferd‘ um Kontrolle über die Konsole zu erhalten, oder ein Virus (Stichwort Makroviren) wird aktiviert, der ein eventuell vorhandenes internes Netzwerk stört.

[63] siehe dazu 'Building Internet Firewalls' [ChZw95]

1.1.3 Motive für den Mißbrauch

An Motiven für die mißbräuchliche Verwendung illegal erworbener Information hat es schon in der Vergangenheit nicht gemangelt. Das Postgeheimnis, sowie die aktuelle Diskussion über den sogenannten großen Lauschangriff zeigen die Sensibilität dieses Themas auf. Die zunehmende Verwendung von Mail-Systemen für den Informationsaustausch, online-shopping und online-banking, bieten neue Angriffsflächen. Durch die elektronische Informationsübertragung ist es möglich, effektiv und nahezu spurlos zu manipulieren. Die folgende Auflistung ordnet die Motive für den Mißbrauch in verschiedene Kategorien:

- kriminelle Gründe:

  Motivation durch die zunehmende Bedeutung des ‚electronic commerce' und die Abwicklung privater Bankgeschäfte. Dabei sind die unrechtmäßige Nutzung von Kreditkartennummern und manipulierte Überweisungen nur eine Möglichkeit. Denkbar ist auch, die georderte Ware an eine andere Adresse schicken zu lassen oder über ein ‚trojanisches Pferd' den Benutzer (bzw. das Anwenderprogramm), zu eigentlich nicht beabsichtigten Überweisungen zu veranlassen.
- Mißbrauch der Privatsphäre:

  Das Mitlesen, die Verfälschung und das Abfangen kann persönliche Gründe haben. Das Speichern von privaten Informationen und die statistische Auswertung kann aber auch kommerziellen Zwecken dienen (man glaubt kaum was man bei einer Auskunftei über sich erfahren kann).
- Industriespionage:

  Naheliegend ist das professionelle ‚Abhören' zum Zweck des Informationsvorteils, um z.B. die Höhe des Angebots des Konkurrenten, Ideen und Inhalte von Werbekampagnen, geplante strategische Unternehmensentscheidungen usw. zu erfahren.
- Kontrolle und Überwachung:

  Die USA ist das beste Beispiel für die Absicht der totalen Überwachung und Kontrolle beim elektronischen Informationsaustausch. Der ‚Clipper' Chip soll es der NSA, einer Kontrollbehörde, ermöglichen, die über Telekommunikationsgeräte (Telefon, Fax usw.) ausgetauschten Informationen abzuhören. Auch in anderen Ländern mit politisch brisanter Lage ist das Mißbrauchspotential direkt erkennbar.
- Zugang zu internen Netzwerken:

  Zugriff auf interne Netzwerke und damit nicht nur auf Informationen, die per e-mail übertragen werden, sondern auf alle gespeicherten Daten. Dadurch besteht zudem die Möglichkeit, Daten zu verfälschen und zu löschen, bzw. das Netzwerk durch einen Virus lahmzulegen.

1.2 Problematik der Informationstechnologie

Die schnelle Entwicklung im Bereich Multimedia und die Nutzung des Internets für Kommunikation und Informationsaustausch führt vielfach zu einer Realisierung des technisch Möglichen, ohne die

Risiken in Bezug auf den Datenschutz und das Mißbrauchspotential zu berücksichtigen. Dieselbe Informationstechnologie, die es ermöglicht sich weltweit, schnell und kostengünstig zu informieren und zu kommunizieren kann umgekehrt auch dazu eingesetzt werden, um die übertragenen Informationen zu filtern, zu seinem eigenen Vorteil einzusetzen. Die Folgen davon sind mögliche Manipulation und Mißbrauch von persönlichen oder wirtschaftlich sensiblen Daten.

1.2.1 Problematik der Mail-Administratoren

Mail-Administratoren stehen vor einer schweren Aufgabe, wenn es um die sichere Übertragung von e-mails geht. Die Ursache dafür ist die Konzeption der Datenübertragung im Internet. Damit es überhaupt möglich war, Daten kreuz und quer durch unterschiedliche Netze zu schicken, wurden Aspekte wie Datensicherheit (Abhör- und Diebstahlsicherheit) hintenangestellt. Im Vordergrund stand der effektive Datentransport und der individuelle Zugang für alle Teilnehmer. Die Struktur dieses weltumspannenden Netzes mit der dementsprechenden Systemvielfalt macht es schwer einen Standard zu definieren und durchzusetzen, der eine sichere Übertragung der Daten gewährleistet [Dama97].

> „[...] So ist zum Beispiel das zentrale Mail-Protokoll SMTP (Simple Mail Transfer Protocol), über das nach wie vor fast alle e-mails im Netz verschickt werden, so offen wie das sprichwörtliche Scheunentor. Eine Mail unter falschem Namen und mit falschem Absender (Mailspoofing) oder eine kostspielige Massenmail über einen fremden Mailserver und damit über ein fremdes Konto laufen zu lassen (Spamming) ist fast so leicht wie der Mausklick auf einen Hyperlink. [...]“ [Dama97]

Mit den ‚secure sockets layern‘ von Netscape gibt es zwar Möglichkeiten, um beispielweise das online-shopping hinreichend sicherer zu gestalten. Aber dieses geänderte Protokoll, erkennbar an der Adresse HTTPS an stelle von HTTP, wird längst nicht von allen Anbietern in Internet benutzt [Dama97]. Aufgrund dieser Strukturproblematik bleibt dem Mail-Administrator nur die Anwendung von Firewalls, die eine systematische Kontrolle der eingehenden Mails erlauben, um das interne Netzwerk zu schützen. Jeder Mail-Administrator bzw. jeder der dessen Zugangsberechtigung hat (ob zurecht oder nicht), hat Einsicht in den e-mail Verkehr und damit alle oben beschriebenen Möglichkeiten des Mißbrauchs. Aufgrund der globalen Struktur des Internets ist es kaum denkbar eine Institution zu etablieren, die wie ehemals die ‚gelbe Post‘, eine Art Postgeheimnis garantieren kann.

1.2.2 Problematik mit Mail-Servern

Mail-Server stellen den Kontakt mit der externen Welt her. Sie empfangen Daten von anderen Servern und Befehle, wie mit diesen Daten zu verfahren ist. Damit sind sie der erste Angriffspunkt auf dem Weg zum Adressaten einer e-mail. Ansatzpunkte für Attacken stellen die empfangenen Befehle dar. Durch den Zugang zu einem Mail-Server erhält man die Kontrolle über alle Funktionen und Zugriffsrechte dieses Servers.

> „The server directly accepts commands (related to delivering mail) from external hosts, for this reason, if the server isn’t secure, it may end up immediatley giving an attacker all the access it has itself. [...] Because it talks to the external world, the server is vulnerable to attacks in

> commands it receives from the outside world; these are called ‚command channel attacks' " [ChZw95].

Um den Server vor solchen Attacken zu schützen, werden Firewalls eingesetzt, die die Zugriffsrechte und erlaubte Kommandos kontrollieren. Zudem sollen in Übertragungsprotokolle integrierte Sicherheitsfunktionen vor Manipulationen schützen. Desweiteren werden neue Protokolle (IPv6) entwickelt, die nicht nur die Datensicherheit, sondern auch Datenschutz durch Verschlüsselung garantieren sollen. Ob und inwiefern die Server derart geschützt sind, die auf dem Weg einer ankommenden e-mail gelegen haben, läßt sich für den Adressaten respektive den Absender der mail nicht nachvollziehen. Damit ist nicht sichergestellt, daß die Daten nicht doch manipuliert bzw. eingesehen werden können. Mithilfe von Verschlüsselungsverfahren, wie PGP oder S/MIME kann aber erreicht werden, das die Daten nur für den Adressaten verwertbar sind.

1.3 PGP

Was ist PGP ?

PGP - ‚Pretty Good Privacy' - ist ein Softwareprodukt zur verschlüsselten Übertragung von e-mails. Wie der Name schon sagt, soll die Privatsphäre (privacy) bei der Übertragung von e-mails gewahrt bleiben. Dabei ist nicht nur eine geschickte Codierung der Information selbst nötig, sondern auch eine Authentifikation des Absenders, um sicherzugehen, daß die Nachricht auch von demjenigen stammt, der vorgibt sie abgesendet zu haben. Die beiden Begriffe ‚Privacy' und ‚Authentifikation' bilden daher den Kern des Konzepts von PGP.

1.3.1 Historischer Abriß / Entwicklung

Das Programm ist 1978 von Phil Zimmermann entwickelt worden und hat sich zum bekanntesten Verschlüsselungsprogramm für e-mails entwickelt. Das hat folgende Gründe: Einerseits ist die Software als Freeware erhältlich und damit kostenfrei für jedermann zugänglich, andererseits hat sich aufgrund jahrelanger bizarrer Rechtsstreitigkeiten um die Verwendung von patentrechtlich geschützten Algorithmen (RSA) und den Export aus den USA (Export starker Kryptographie ist verboten: Stichwort RC2/40) eine regelrechte ‚Fangemeinde' gebildet [URL-1].

1.3.2 Arbeitsweise von PGP

1.3.2.1 Das Prinzip der asymmetrischen Verschlüsselung [64]

PGP arbeitet nach dem Prinzip der asymmetrischen Verschlüsselung. In diesem 1976 von Diffie und Hellmann entwickelten Public-Key Verfahren besitzt jeder Anwender zwei unterschiedliche, aber zusammengehörige Schlüssel: einen öffentlichen (public key) und einen privaten Schlüssel (private

[64] siehe dazu Vortrag „Kryptographie: Grundlagen und Algoritmen"

key). Diese Schlüssel sind so beschaffen, daß eine Nachricht, die mit einem öffentlichen Schlüssel kodiert wurde, nur mit dem zugehörigen privaten Schlüssel dekodiert werden kann [GrWe97].
Die Nachrichten werden dabei mit dem öffentlichen Schlüssel des Empfängers kodiert. Die Entschlüsselung funktioniert nur mit dem privaten Schlüssel des Empfängers. Der öffentliche Schlüssel wird notwendigerweise an die e-mail Partner versandt. Da bei dem Prinzip der asymmetrischen Verschlüsselung nur der private Schlüssel die Dekodierung ermöglicht, kann für den Austausch der öffentlichen Schlüssel ein potentiell unsicherer Übertragungsweg benutzt werden. Die Authentizität wird durch eine Unterschrift mit dem privaten Schlüssel sichergestellt.

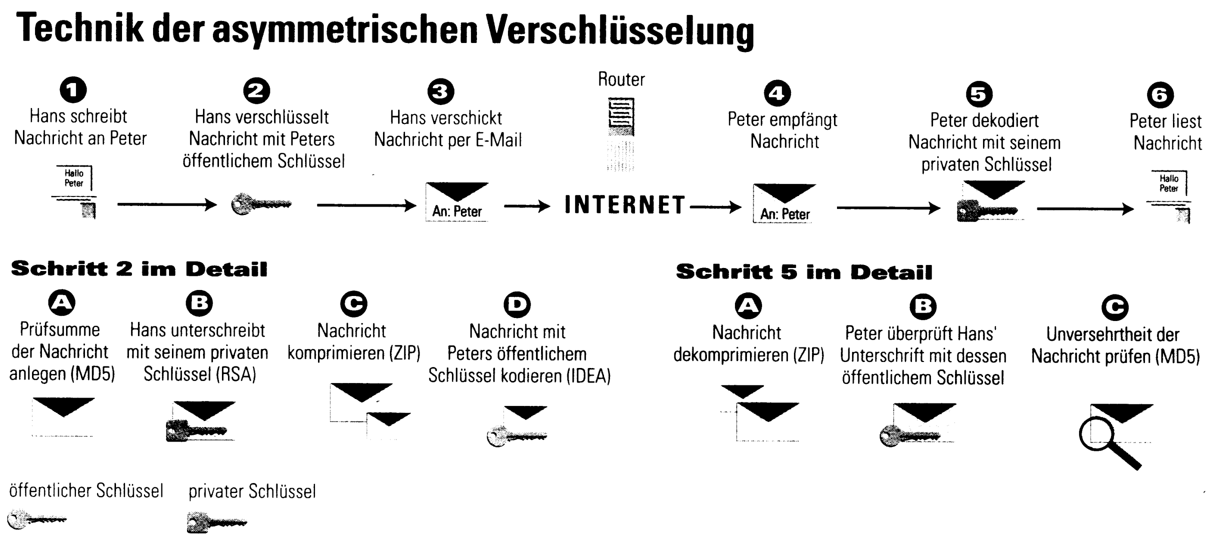

Abb.1: Technik der asymmetrischen Verschlüsselung [GrWe97].

1.3.2.2 RSA Algorithmus

Für die Erzeugung des Schlüsselpaars wendet PGP den 1977 von Rivest, Shamir und Adleman entwickelten RSA Algorithmus an. Er beruht auf der Multiplikation von Primzahlen und der Tatsache, daß die Zerlegung in Primfaktoren bei entsprechender Wahl der Schlüsselgröße praktisch unmöglich ist. Damit soll gewährleistet sein, daß der private Schlüssel nicht aus dem öffentlichen Schlüssel berechnet werden kann. PGP bietet bei der Schlüsselgenerierung die Wahl zwischen 384 und 2048 Bit. Default-Werte sind 512 Bit für ‚einfach komerziell' 768 Bit für ‚hochgradig komerziell' und 1024 Bit für ‚militärische Sicherheit'. Die Bearbeitungszeit für die Verschlüsselung mit dem RSA Algorithmus ist für das Verschlüsseln ganzer Nachrichten unkomfortabel lang. Daher wird RSA nur für die Schlüsselgenerierung und die Kodierung der Schlüssel verwendet, die die Nachricht und die Unterschrift kodieren.

1.3.2.3 IDEA Verfahren

Das IDEA Verfahren ist ein symmetrisches Blockverschlüsselungsverfahren. Blöcke von je 64 Bit Länge werden dabei mit einem 128 Bit Schlüssel kodiert. PGP setzt dieses Verfahren für die Verschlüsselung der Nachricht ein. Der IDEA Schlüssel wird dabei nur ein einziges mal verwendet. Vor der Versendung der Nachricht wird er mit dem privaten Schlüssel kodiert und dann zusammen mit der Nachricht übertragen.

1.3.2.4 Message Digest 5

Das Message Digest 5 Verfahren erzeugt eine Prüfsumme der zu übertragenden Nachricht. Die so erzeugte 128 Bit Zahl wird wiederum mit dem privaten Schlüssel des Absenders kodiert. Damit wird die Authentizität und die Unversertheit der Nachricht sichergestellt.

1.3.3 Angriffsmöglichkeiten

Prinzipiell kann es kein absolut sicheres System geben. Selbst wenn die technischen Anforderungen an ein sicheres System erfüllt werden, bleibt der Faktor Mensch und die Tatsache der Implementierung in einer potentiell unsicheren Umgebung bestehen.

1.3.3.1 Kryptoanalytische Attacke

Eine kryptoanalytische Attacke kann sich auf IDEA oder das RSA Schlüsselpaar beziehen. In beiden Fällen sind jedoch keine erfolgreichen Attacken bekannt. Selbst durch eine massive sogenannte ‚brute force‘ Attacke ist nicht zu erwarten, daß diese Schlüssel geknackt werden könnten. Auf der anderen Seite kann eine bahnbrechende Entwicklung im Bereich der systematischen Attacke (z.B. Algorithmus zur Primfaktorzerlegung) in Zukunft zum Erfolg führen. Es gilt: in der praktischen Kryptographie gibt es keine Garantie für absolute Sicherheit [URL-1].

1.3.3.2 Fälschung öffentlicher Schlüssel

Die Achillesferse bei Systemen, die mit öffentlichen Schlüsseln arbeiten ist in der Schlüsselverwaltung und Gültigkeitsüberprüfung der öffentlichen Schlüssel begründet. Vor der Akzeptierung eines öffentlichen Schlüssels ist unbedingt die Authenzität zu prüfen. Auch die Verwaltung des öffentlichen Schlüsselbundes stellt ein Sicherheitsrisiko dar. Im Gegensatz zum privaten Schlüssel ist diese Datei in der Regel nicht durch ein Mantra geschützt.

1.3.3.3 Virus bzw. ‚trojanisches Pferd‘

Eine andere Angriffsmöglichkeit könnte ein speziell entwickelter Virus oder Wurm sein, der PGP oder das Betriebssystem infiziert. Dieser hypothetische Virus könnte so entworfen sein, daß er das Mantra, den geheimen Schlüssel oder den entschlüsselten Klartext "mithört" und unbemerkt in eine Datei schreibt oder über ein Netzwerk zum Besitzer des Virus schickt. Er könnte auch das Verhalten von

PGP so ändern, daß Unterschriften nicht richtig geprüft werden. Ein ähnlicher Angriff könnte eine geschickte Imitation von PGP sein, die sich im Wesentlichen wie PGP verhält, aber nicht so arbeitet, wie anzunehmen wäre. Beispielsweise könnte diese Imitation absichtlich dahingehend verstümmelt sein, daß Unterschriften nicht mehr korrekt geprüft werden, so daß gefälschte Schlüssel nicht mehr erkannt werden können. Eine solche Version von PGP - ähnlich einem ‚trojanischen Pferd' - ist von einem Angreifer verhältnismäßig einfach erstellt, weil der Quellcode von PGP weit verbreitet ist [URL-1].

1.3.3.4 Zugriff auf die Benutzerkonsole

Ein weiteres potentielles Sicherheitsproblem entsteht durch die Art und Weise, wie bei den meisten Betriebssystemen Dateien gelöscht werden. Wenn man eine Klartext-Datei verschlüsselt und danach löscht, löscht das Betriebssystem die Daten nicht physikalisch. Es markiert nur diejenigen Datenblöcke der Festplatte oder Diskette als "gelöscht", die den Inhalt der "gelöschten" Datei enthalten, so daß sie für die Speicherung anderer Daten freigegeben werden. Das ist das gleiche, als würde man vertrauliche Papiere einfach zum Altpapier legen. Die Blöcke auf der Festplatte enthalten nach wie vor die originalen vertraulichen Daten, die ohne großen Aufwand wiederhergestellt werden können [URL-1].

1.3.3.5 Kompromittierende Strahlung

Eine andere Angriffsmöglichkeit für einen gut ausgerüsteten Gegner ist die Auswertung der elektromagnetischen Strahlung, die ein Computer aussendet. Mithilfe der entsprechenden technischen Ausrüstung ist es möglich, bis zu Entfernungen von 100m, diese Strahlung auszuwerten und damit jeden Tastendruck und jeden Bildschirminhalt aufzuzeichnen. Das würde alle Passworte, Nachrichten usw. offenlegen. Abwehren läßt sich dieser Angriff durch eine geeignete Abschirmung des Computers, des Zubehörs (Drucker usw.) und gegebenenfalls der Netzwerk-Verkabelung. Eine solche Abschirmung ist unter dem Begriff "sturmsicher" bekannt.

1.4 S/MIME

S/MIME ist die Abkürzung für ***S**ecure **M**ultimedia **I**nternet **M**ail **E**xtensions*. Dieses Protokoll geht auf MIME zurück, ein Übertragungsprotokoll, daß die enthaltenen Formate der zu übertragenden Datei beschreibt. MIME definiert dabei einen Standard für Mail-Programme. Dieser ermöglicht es, Dateien, die RTF-Texte (Rich Text Format: formatierter Text mit z.B. unterschiedlichen Fonts), Grafiken sowie Audio-Daten und andere Multimedia Informationen enthalten, direkt mit dem geeigneten Programm (soweit vorhanden), darzustellen [ChZw95]. S/MIME ist eine Weiterentwicklung dieses Standards, der nun auch Funktionen für die Ver- und Entschlüsselung von Dateien enthält.

1.4.1 Entwicklung

Grundgedanke bei der Entwicklung des MIME Standards war die Erweiterbarkeit um zusätzliche Funktionalität. Daher war es naheliegend diesen Standard, der auch für den Datentransfer der WWW-Server untereinander benutzt wird, um die Funktionalität der sicheren Übertragung von e-mails zu erweitern. Hauptziele bei der Implementierung waren Sicherheit und Interoperabilität unter Berücksichtigung einer globalen Anwendbarkeit. Das bedeutet:

- Sicherheit und Anwendbarkeit für viele Anwender:
  S/MIME basiert auf den Public Cryptography Standards (PKCS) insbesondere dem PKCS #7 Standard, in dem Verfahren der digitalen Signatur und des digitalen Umschlags festgelegt sind [URL-3]. Die Einhaltung dieser Standards und das Verfahren von ‚Beglaubigungsinstitutionen' für digitale Unterschriften erschließt einen großen Anwenderkreis.
- Interoperabilität:
  Dieses Prinzip setzt auf die Integration der S/MIME Funktionalität in führende Applikationen (Netscape Communicator, MS Internet Explorer usw.). Ziel ist es, ein Label, vergleichbar mit ‚IBM-Kompatibel', zu etablieren, sodaß jede Software, die dieses Label aufweist, mit den verschlüsselten Dateien der anderen Software-Produkte umgehen kann [URL-3].
- Globale Anwendbarkeit:
  Um eine unproblematische, weltweite Anwendbarkeit zu garantieren, ist es wichtig, die US-Exportbeschränkungen für Kryptographie einzuhalten. Innerhalb der USA kann starke Kryptographie angewendet werden. Für den Übersee-Kontakt werden kleinere Schlüsselgrößen eingesetzt (RC2/40) [URL-4].

1.4.2 Arbeitsweise

Zunächst wird nach dem Prinzip der asymmetrischen Verschlüsselung ein privater und ein öffentlicher Schlüssel nach dem RSA Algorithmus generiert. Dann wird die Nachricht mit einer digitalen Signatur versehen. Dazu durchläuft das Dokument einen Hashing-Algorithmus (Message Digest), sodaß ein Fingerabdruck des Dokuments erzeugt wird. Mit dem privaten Schlüssel wird es nun unterschrieben. Jetzt ist sowohl der Inhalt vor Veränderung geschützt, als auch die Authentifikation des Absenders gegeben. Dann wird das Dokument mit DES oder RC2 (symmetrische Verfahren) kodiert und der symmetrische Schlüssel mit dem öffentlichen Schlüssel des Empfängers verschlüsselt [URL-4]. Das DES Verfahren (Data Encryption Standard) ist in den 70er Jahren von IBM entwickelt worden und kodiert die Nachricht blockweise (64 Bit Blockgröße) mit 56 Bit. RC2 arbeitet mit einstellbarer Schlüsselgröße. Um beispielsweise die US-Exportvorschriften einzuhalten, wird eine Schlüsselgröße von 40 Bit verwendet (RC2/40). Der ganze Vorgang läuft zugunsten der einfachen Anwendbarkeit automatisch ab, d.h. er ist durch die Integration in den Mail-Clienten vorgegeben. Ebenso verhält es sich auf der Empfängerseite [URL-4].

1.4.3 Angriffsmöglichkeiten

Prinzipiell kann es kein absolut sicheres System geben. Selbst wenn die technischen Anforderungen an ein sicheres System erfüllt werden, bleibt der Faktor Mensch und die Tatsache der Implementierung in einer potentiell unsicheren Umgebung bestehen [65].

1.4.3.1 Kryptoanalytische Attacke

Eine kryptoanalytische Attacke kann sich auf den symmetrischen Schlüssel oder das RSA Schlüsselpaar beziehen. Für die RSA Schlüssel gilt: es sind keine erfolgreichen Attacken bekannt. Die verwendeten symmetrischen Verfahren, DES mit 56 Bit und RC2 mit 40 Bit (für internationalen Informationsaustausch), sind ein Schwachpunkt. Der mit DES erzeugte Schlüssel ergibt $2^{56} = 7.2 \times 10^{16}$ Möglichkeiten und kann mithilfe schneller Rechner bereits in einigen Stunden ‚geknackt' werden. Besonders markant wird dieses Defizit bei RC2/40 deutlich. Bei dieser Schlüssellänge existieren $2^{40}=1.1 \times 10^{12}$ verschiedene Kombinationen. Dazu folgendes Beispiel:

‚Screensaver als Schlüsselknacker' [CT97_3] aus c't 12/97

> „Krypto-Autor und Sicherheitsberater Bruce Schneier bietet auf seinem WWW-Server einen Windows-95 Screensaver zum Download an, der im Hintergrund 40 Bit lange RC2-Schlüssel knackt.... Hintergrund ist die Selbstbeschränkung von S/MIME –Anwendungen auf RC2/40".

Damit wird der Schwachpunkt von S/MIME deutlich. Für die Zukunft ist allerdings eine Änderung in den US-Exportvorschriften für starke Kryptographie zu erwarten. Damit können dann längere Schlüssel benutzt werden, z.B ‚triple DES'.

1.4.3.2 Fälschung öffentlicher Schlüssel

Die Achillesferse bei Systemen, die mit öffentlichen Schlüsseln arbeiten ist in der Schlüsselverwaltung und Gültigkeitsüberprüfung der öffentlichen Schlüssel begründet. Bei S/MIME geschieht die Schlüsselüberprufung mithilfe von Zertifizierungsstellen. Ansatzpunkt für einen Angriff ist damit nicht mehr der einzelne Anwender, sondern eben diese Institutionen. Gelingt es Zugang zu erlangen, so sind alle von dieser Stelle beglaubigten Schlüssel potentiell unsicher. Andererseits ist diese Art der Schlüsselverwaltung notwendig um große Benutzergruppen zu erschließen.

1.4.3.3 Zugriff auf die Benutzerkonsole

Eine weitere Möglichkeit erschließt sich über physikalischen oder elektronischen Zugang zur Benutzerkonsole. Der Zugang wird innerhalb interner Netzwerke zumeist durch eine einfache Passwortabfrage kontrolliert. Wird dieses Passwort dem Angreifer bekannt, so hat er Zugriff auf die Datei die Schlüsseldateien. Zudem ist auch hier die Möglichkeit der Wiederherstellung von nicht physikalisch gelöschten Dateien gegeben.

1.4.3.4 Kompromittierende Strahlung

Eine andere Angriffsmöglichkeit für einen gut ausgerüsteten Gegner ist die Auswertung der elektromagnetischen Strahlung, die ein Computer aussendet. Mithilfe der entsprechenden technischen Ausrüstung ist es möglich, bis zu Entfernungen von 100m, diese Strahlung auszuwerten und damit jeden Tastendruck und jeden Bildschirminhalt aufzuzeichnen. Das würde alle Passworte, Nachrichten usw. offenlegen. Abwehren läßt sich dieser Angriff durch eine geeignete Abschirmung des Computers, des Zubehörs (Drucker usw.) und gegebenenfalls der Netzwerk-Verkabelung. Eine solche Abschirmung ist unter dem Begriff "sturmsicher" bekannt [66].

1.5 Diskussion PGP versus S/MIME

Vergleicht man PGP und S/MIME so stellt man fest, daß die verwendeten Verschlüsselungsverfahren auf demselben Konzept beruhen (Verschlüsselung der Information mit symmetrischen Verfahren – asymmetrisches Verfahren für die öffentlichen und privaten Schlüssel). Aufgrund der Einhaltung exportpolitischer US-Vorschriften steht S/MIME auf den ersten Blick schlechter da. Für den professionellen Einsatz und eine große Verbreitung im Hinblick auf eine Standartisierung durch die IETF (Internet Engeneering Task Force) ist es jedoch kaum denkbar Software einzusetzen, gegen die patentrechtliche Bedenken bestehen. Zudem hat die Vergangenheit gezeigt, daß sich die Software am Markt durchsetzt, die interoperabel mit den führenden Softwareprodukten ist (Stichwort ‚Bill Microsoft Gates‘) . Da PGP diese beiden Aspekte nicht erfüllt, bietet es sich für Anwender an, für die Sicherheit, Individualität und Unabhängigkeit im Vordergrund stehen. Für S/MIME gilt: Nicht nur die Reduzierung der Schlüsselkomplexität, um den US Exportvorschriften zu genügen, sind an das heute technisch Erforderliche anzupassen, auch der Punkt der Interoperabilität ist bei konkurrierenden Software-Produkten längst nicht automatisch sichergestellt.

Zur Marktsituation:

Nachdem es zunächst so aussah als würde S/MIME die besten Karten für die Normung durch die IETF haben, gab es dann noch Schwierigkeiten mit Lizenzen in Bezug auf den RSA Algorithmus [URL-2].

Daraufhin zeigte PGP mit dem neuen Open-PGP Kompromißbereitschaft. Mittlerweile bietet PGP inc. komplette Programmpakete an (z.B. PGP Business Security Suite) [URL-7] [URL-8], die aber auch Funktionen enthalten, die dem ursprünglichen Grundkonzept widersprechen. ‚Corporate Message Recovery‘ fordert für alle verschlüsselten e-mails von und an Mitarbeiter zusätzlich die Kodierung mit dem Firmenschlüssel. Dadurch wird eine quantitative Kontrolle durch die Firmenleitung möglich. Dies ist ein Bruch im Konzept von PGP. Die Ausdehnung der Anwendung dieser Funktionalität auf Provider oder staatliche Institutionen wäre bedenklich, technisch aber nur ein kleiner Schritt [CT97_1]. Andererseits werden Probleme bei der Authentifizierung der öffentlichen Schlüssel durch

[65] siehe dazu Vortrag „Angriffsstrategien“
[66] siehe dazu auch Vortrag „Sicherheit auf der physikalischen Schicht“

sogenannte Cross-Zertifizierung von PGP Zertifizierungsstellen gelöst. Damit werden größere Anwendergruppen erschlossen. Die so erhaltenen öffentlichen Schlüssel sind damit beglaubigt [CT97_2].
Durch die Integration des S/MIME Standards in neue Browser-Generationen (IE4 und Netscape Communicator) wird S/MIME für praktisch jedermann zugänglich. Allein die Tatsache, daß diese Funktionalität praktisch per Mausklick verfügbar ist und man sich keine Gedanken um Kompatibilität und Version x.y machen muß, läßt erwarten, daß sich in Zukunft die meisten Anwender für S/MIME entscheiden werden.

1.6 ICQ

ICQ ist ein Internet Tool. Es dient dazu im online Betrieb eine Echtzeit Verbindung herzustellen - sozusagen einen Kanal offenzuhalten. Auf diesem Kanal können Gespräche stattfinden (Online chat), Nachrichten ausgetauscht werden, Dateien versand werden oder man hat ein ‚offenes Ohr' für die Internet Umwelt („[...] hang out with your fellow netters while still surfing the Net") [URL-10].
Im Vergleich zur e-mail bietet ICQ den Vorteil der direkten Erreichbarkeit und des interaktiven Austauschs von Informationen. Durch den multi-user Modus sind auch online Konferenzen möglich. Der Austausch von Nachrichten und Informationen ermöglicht die gemeinsame Arbeit an einem Projekt [URL-10]. Desweiteren haben sich themenbezogene Gesprächsforen und Anwendergruppen gebildet [URL-12]. Zusätzlich dazu wird die konventionelle Versendung von e-mails und Dateien angeboten. Um diese Möglichkeiten zu eröffnen ist ein WWW-weites Servernetz installiert worden, an das man sich automatisch zu Beginn einer online-Sitzung anmeldet [URL-9]. Die Zahl der Anwender liegt bei 7.4 Millionen (Stand Ende Januar 98). Etwa 2 Millionen nutzen dieses Tool täglich [URL-12]. Bei der Installation der Software wird eine UIN - Universal Internet Number - vergeben, unter der man bei einem ICQ-Server als registrierter Anwender geführt wird. Zusätzlich füllt man einen Personalbogen mit Name, Adresse aber auch persönlichen Daten aus [URL-11]. Korrektheit und Vollständigkeit der Angaben ist nicht erforderlich. Die Daten dienen den anderen ICQ-Nutzern zur Auswahl des Chat-Partners. Man kann diesen Personalbogen als erweiterten Telefonbucheintrag verstehen. Dementsprechend sollte man berücksichtigen wozu man ICQ benutzt und wie man sich darstellen möchte. Grundsätzlich sind diese Angaben nicht geschützt. Der ICQ Client ist kompatibel mit Netscape Communicator und MS Internet Explorer sowie den meisten e-mail Applikationen [URL-9].

1.6.1 Arbeitsweise

Beim Start des Betriebssystems wird das Programm im ‚detect' Modus geladen. Sobald eine Internet-Verbindung hergestellt wird, erkennt ICQ diese Verbindung und geht in den online Status über. Dadurch ist man im ICQ Servernetz angemeldet und erreichbar für alle anderen Anwender [URL-11]. Um selbst aktiv werden zu können, erstellt man sich eine persönliche Adressenliste. Die gewünschten Anwender wählt man über eine Suchmaschine anhand ihrer UIN oder den Daten aus dem Personalbogen aus. Das Serversystem ermittelt automatisch ob die betreffenden Anwender off- oder on-line sind und in welchem Status. Mögliche Stati sind: ONLINE für ereichbar, AWAY bei kurzer Abwesenheit (durch Bildschirmschoner aktiviert), DO NOT DISTURB in Erwartung wichtiger Mitteilungen und PRIVACY läßt es erscheinen als wäre der Anwender offline [URL-11]. Nun ist es möglich, Anfragen für Gesprächsbereitschaft (chat requests), Nachrichten usw. zu senden. Dabei ist es nicht zwingend nötig, daß der gewünschte Gesprächspartner online ist. Wie bei der normalen e-mail erhält der betreffende Anwender die Nachricht sobald er online geht. Zusätzlich ist es noch möglich über die Funktionalität IGNORE Anfragen und Nachrichten von bestimmten Anwendern zu unterbinden. Die ungewollte Aufnahme in ‚fremde' Anwenderlisten kann mittels AUTHORIZATION verhindert werden [URL-11].

1.6.2 Problematik im Hinblick auf Datenschutz und Datensicherheit

Die Übertragung von e-mails und das Versenden von Dateien geschieht über die Standard Internet-Protokolle und ist entkoppelt vom ICQ Servernetz. Damit entspricht die Datensicherheit dem Status des Internets. Konzepte die den Datenschutz verbessern, z.B. durch Verschlüsselung, werden nicht angeboten. Problematischer ist der Informationsaustausch innerhalb des ICQ Servernetzes zu beurteilen. Durch das schnelle Wachstum der Anwenderzahlen kommt es zwangsläufig zu Überlastungen des Servernetzes. In solchen Fällen wird die Vermaschung des Netzes reduziert, d.h. konkret: die Verbindung zwischen Servern wird unterbrochen. Damit werden bestehende Verbindungen gekappt. Für den Anwender bedeutet das: bestehende Verbindungen werden aus unbekannten Gründen unterbrochen (keine Meldung über Serverüberlastung) und nicht jeder chat-Partner kann erreicht werden. Die Datensicherheit ist daher nicht gewährleistet. Was den Datenschutz angeht, so ist zu berücksichtigen das das Konzept dieses Kommunikationssystems eine gewisse Offenheit erfordert. Es ist gewünscht das jeder Anwender Zugang zu den Gesprächsforen hat und es ist notwendig (im Sinne des Konzepts) sich die Gesprächspartner anhand ihrer persönlichen Daten aus dem oben angesprochenen Personalbogen auszuwählen. Bedenklich ist allerdings, daß dem Anwender eine Anonymität zugesichert wird (im Personalbogen werden in der Regel Spitznamen angegeben), die nicht geschützt ist. Selbst der Vorgang der online-Registrierung und die Vergabe der UIN Nummer geschieht unverschlüsselt. Dadurch besteht kein Schutz vor Manipulation. Dadurch, daß Funktionen wie AUTHORIZATION und IGNORE speziell zu aktivieren sind, kann es ungewollt passieren, daß sich Mithörer einschleichen oder man mit Personen ‚chattet', die nicht die sind, die sie vorgeben zu sein. Dadurch sind die Aspekte PRIVACY und AUTHENTIFIKATION nicht erfüllt.

System Administratoren sind im Bereich des Server-Netzes für Einhaltung des normalen Betriebs zuständig. Bei Bedarf können Zugangsverbote ausgesprochen werden. Wie groß das Mißbrauchspotential in dem Bereich der Server-Verwaltung ist, kann schlecht eingeschätzt werden. Insgesamt kann man sagen, daß ICQ eine komfortable Echtzeitkommunikation ermöglicht und darüber hinaus

Funktionalitäten für das Versenden von Nachrichten sowie die Bildung von multi-user Konferenzen bietet. Geht es aber um die Übertragung von sensiblen Informationen, sollte man geschützte Übertragungswege bzw.Verfahren wie PGP oder S/MIME nutzen.

2 Literatur- und Quellenverzeichnis

2.1 Printmedien

[ChZw95] D. B. Chapman; E. D. Zwicky. *Building Internet Firewalls*. O'Reilly, Sebastopol (CA, USA), 1995.

[Dama97] G. Damaschke. *Hard- und Softwarekauf im Internet*. PC Professionell. Ziff-Davis Verlag GmbH, Nr.8, 1997.

[GrWe97] F. Grieser; H. Weiss. *Der Schlüssel zur Sicherheit*. PC Professionell. Ziff-Davis Verlag GmbH, Nr. 6, 1997.

[CT97_1] C'T. *PGP unter Verdacht*. Computer und Technik, Nr. 12, 1997.

[CT97_2] C'T. *IN- und c't-CA crosszertifiziert*. Computer und Technik, Nr. 12, 1997.

[CT97_3] C'T. *Screensaver als Schlüsselknacker*. Computer und Technik, Nr. 12, 1997.

2.2 Elektronische Dokumente

[URL-1] Verein zur Föderung des bewegten und unbewegten Datenverkehrs, Stand 01.09.1996 (gesichtet 20.11.1997): `http://iaks-www.ira.uka.de/ta/Security/security.html`

[URL-2] IETF shows door to RSA secure e-mail proposal, by Ellen Messmer, Network World 8/25/97, Stand 25.08.1997 (gesichtet 16.12.1997): `http://iaks-www.ira.uka.de/ta/Security/security.html`

[URL-3] S/MIME frequently asked questions, Stand 16.12.1997 (gesichtet 16.12.1997): `http://www.rsa.com/rsa/s-mime/faq.html`

[URL-4] S/MIME Anatomy of a secure e-mail standard, Stand 16.12.1997 (gesichtet 16.12.1997): `http://www.rsa.com/rsa/s-mime/s-mime_anatomy.html`

[URL-5] Look out S/MIME and Open PGP, there is a new kid on the block,Charlotte Dunlap, Stand 15.09.1997 (gesichtet 16.12.1997): `http://www.techweb.com/se/directlink.cgi?CRN19970915S0083`

[URL-6] IETF to develop Open-PGP, John Fontana , Stand 27.09.1997 (gesichtet 16.12.1997): `http://www.techweb.com/se/directlink.cgi?WIR1997092702`

[URL-7] PGP reverses direct strategy, Charlotte Dunlap , Stand 27.11.1997 (gesichtet 16.12.1997):
`http://www.techweb.com/se/directlink.cgi?CRN19971124S0032`

[URL-8] Network Associates merges with PGP, Charlotte Dunlap , Stand 08.12.1997 (gesichtet 16.12.1997):
`http://www.techweb.com/se/directlink.cgi?CRN19971208S0036`

[URL-9] Mirabilis – Products Features (gesichtet 27.11.1997):
`http://www.mirabilis.com`

[URL-10] Mirabilis – What is ICQ (gesichtet 27.11.1997):
`http://www.mirabilis.com`

[URL-11] Mirabilis – How to use ICQ (gesichtet 27.11.1997):
`http://www.mirabilis.com`

[URL-12] ICQ – Home (gesichtet 28.01.1998):
`http://www.icq.com`

Firewalls

Manuel Brinkmann

1 Einleitung

Das Internet stellt für viele Institutionen und Firmen ein wichtiges Medium zum Informationsaustausch dar. Hier werden Produkte beworben, Forschungsergebnisse ausgetauscht und dargestellt, freie Software angeboten, usw. Durch das Internet sind die Benutzer nicht mehr auf ihren Rechner bzw. die Rechner im lokalen Netzwerk beschränkt, sondern können das Informationsangebot und die Rechenleistung jedes an das Internet angeschlossenen Computers nutzen. Letzteres wird durch Zugangsbeschränkungen eingeschränkt, das bedeutet, daß nur berechtigte Benutzer Zugriff auf einen Rechner erhalten, nachdem sie sich z.B. durch ein Paßwort authentifiziert haben. Aber auch für reine Informationsdienste wie das WWW sind Zugangsbeschränkungen entwickelt worden. Hierdurch soll verhindert werden, daß persönliche oder andere schützenswerte Informationen frei zugänglich sind.

Die Komplexität der heute benötigten leistungsfähigen Rechnersysteme und die Größe von Rechnernetzen erschweren es jedoch, einen wirklich „wasserdichten“ Zugriffsschutz auf vertrauliche Daten zu gewährleisten. Unauthorisierter Zugriff kann durch Softwarefehler oder sonstige „Hintertürchen“ in ein System geschehen. In beiden Fällen ist es häufig schwierig, einen solchen Zugriff festzustellen und dann schnell zu beheben. Grundsätzlich gibt es zwei Möglichkeiten der Vorbeugung: die Absicherung jedes Rechners vor bekannten Sicherheitslücken (*Host security*) oder das Verhindern von unberechtigten Aktionen über das Netzwerk (*Network security*).

Die Absicherung jedes Rechners kann, nach dem Bekanntwerden von Sicherheitslücken des Systems, durch das Einspielen von sogenannten *Patches*[67] geschehen. Diese Patches werden häufig vom Hersteller der Betriebssystem- oder sonstiger Software zur Verfügung gestellt. Dieses Sicherungskonzept ist jedoch mit einem hohen Arbeitsaufwand verbunden. Der Systemadministrator muß auf dem laufenden sein, welche aktuellen Patches es gibt, und entscheiden, ob sie für sein(e) System(e) benötigt werden. Dann muß er diese gegebenenfalls besorgen und auf jedem Rechner installieren. Dieser Vorgang läßt sich nur schwer automatisieren, da zwischen einzelnen Patches häufig Abhängigkeiten bestehen und die Rechner eines bestimmten Systems aufgrund geringer Unterschiede in der Installation unterschiedliche Patches benötigen können. Diese Schwierigkeiten treten um so mehr bei heterogener Rechnerausstattung auf.

[67] Patches modifizieren oder ersetzen lokale Dateien, um Fehler zu beseitigen.

Deshalb ist es für große Einrichtungen einfacher, unberechtigte Zugriffe von außen bereits durch das Netzwerk blockieren zu lassen, damit sie auf den nicht vollständig abgesicherten Rechnern keinen Schaden anrichten können. Diese Blockade wird durch Firewalls ermöglicht.

2 Was sind Firewalls?

Ein Firewall ist ein System, das nicht erwünschte Zugriffe über das Internet auf lokale Rechner bzw. von lokalen Rechnern ins Internet verhindert, ungefährliche Internet-Dienste jedoch (in beiden Richtungen) zuläßt. Was „nicht erwünschte Zugriffe“ und „ungefährliche Internet-Dienste“ sind, hängt vom Sicherheitskonzept der jeweiligen Institution ab. Dieses Sicherheitskonzept sollte einen Kompromiß zwischen den benötigten und den als sicher eingestuften Diensten darstellen, und ist deshalb von Institution zu Institution unterschiedlich. Für eine Firma ist es z.B. wichtig, daß sich Außendienstmitarbeiter per *Telnet*[68] über das Internet anmelden können, während eine andere Firma das damit verbundene Risiko scheut und nur einen *HTTP-Server*[69] nach außen anbieten möchte. Ebenso kann es Teil eines Sicherheitskonzeptes sein, lokalen Benutzern lediglich den *E-Mail*[70]-Dienst zur Verfügung zu stellen, während ein anderes Konzept auch *ftp*[71] erlaubt. Ein eigenes Sicherheitskonzept muß von jeder Institution individuell nach ihren Bedürfnissen erstellt werden.

Firewalls sind Teil von Sicherheitskonzepten. Sie überwachen die Einhaltung der Regeln des Konzeptes, verhindern Verstöße und melden die Versuche an den Administrator. Der Firewall stellt die einzige Verbindung zwischen Internet und dem lokalen Netzwerk dar. Solange die Aktionen den Sicherheitsregeln entsprechen, leitet der Firewall sie weiter, andernfalls werden sie blockiert. Die schematische Funktionsweise und die Anbindung eines lokalen Netzes via Firewall an das Internet zeigt Abb. .

Die in der Abbildung durchgestrichenen Verbindungen sind nicht zulässig. Es dürfen keine weiteren, nicht geschützten Verbindungen zum Internet existieren (im unteren Teil der Abbildung), da hierdurch der Firewall leicht umgangen werden kann und damit nutzlos wird. Solche Verbindungen sind für Systemadministratoren teilweise schwer zu entdecken, dies können z.B. temporäre Modemverbindungen, die das Internet-Protokoll benutzen, von lokalen Benutzern sein. Das ist ein Beispiel dafür, daß zur Umsetzung eines Sicherheitskonzeptes auch ein Benutzertraining gehört.

Da sämtliche Datenübertragungen durch den Firewall laufen, kann hier entschieden werden, ob es sich um einen legalen Zugriff handelt, der durchgelassen wird, oder ob ein Angriff bzw. eine ungewünschte Aktion durchgeführt wird, die der Firewall blockieren soll (im mittleren Teil von Abb.). Eine Sicherheitsverletzung kann dadurch an einer zentralen Stelle festgestellt werden, was die Über-

[68] Telnet erlaubt Benutzern das Anmelden auf entfernten Rechnern, für die sie eine Nutzungsberechtigung besitzen.
[69] HyperText Tranfer Protocol. Diese Server versenden Daten im World Wide Web.
[70] E-Mails sind elektronische Nachrichten, die über das Internet an bestimmte Benutzer gesendet werden können.
[71] File Transfer Protocol. Hiermit können Dateien über das Internet zwischen zwei Rechnern übertragen werden.

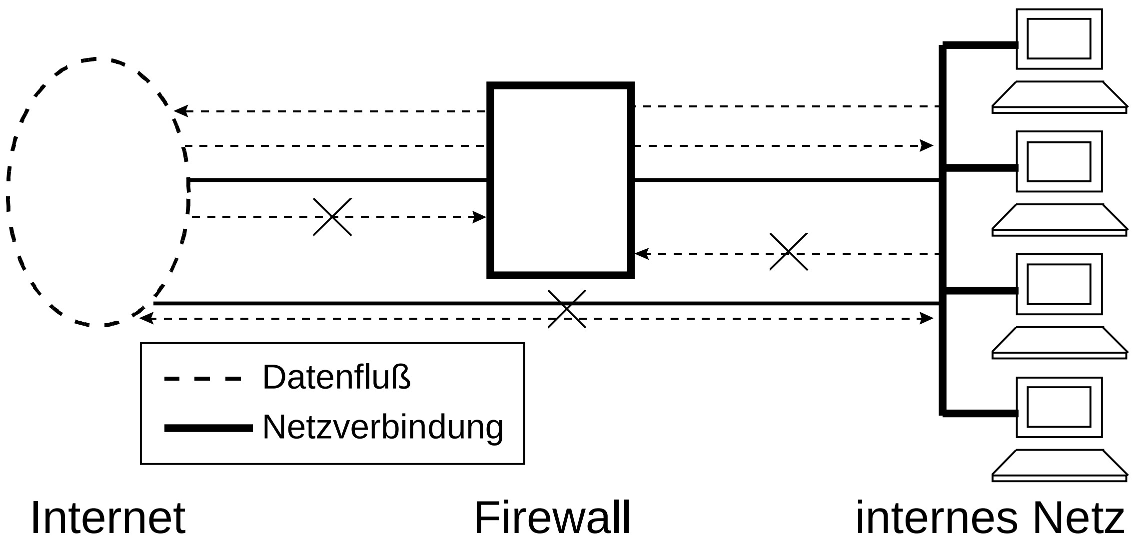

Abb. 1: Schematische Funktionsweise eines Firewalls.

wachung der Sicherheit einer Institution vereinfacht. Der Administrator kann sich auf diese eine Stelle konzentrieren, statt mehrere potentielle Angriffsquellen absichern zu müssen. Um die Überwachung zu erleichtern, sollten Firewalls ausführliche Zugriffsprotokolle erstellen, die häufig auch automatisch überprüft werden können. In diesem Fall muß der Verantwortliche nur vereinzelte Routinekontrollen machen, bei ernstzunehmenden Verstößen gegen die Sicherheit schlägt das System selbständig Alarm. Hat ein Angriff stattgefunden, können anhand der Protokollierungsdaten die Aktionen des Angreifers nachvollzogen werden und angerichtete Schäden und Sicherheitslücken des Systems entdeckt werden.

Ein Firewall kann das gesamte Netz einer Institution schützen, was sich schon bei einer relativ kleinen Rechnerzahl als Vorteil gegenüber der Absicherung jedes einzelnen Computers erweist. Diese Absicherung wird nicht überflüssig, kann aber in kleinerem Umfang vorgenommen werden, da der Großteil der schädigenden Aktionen aus dem Internet bereits am Firewall abgefangen wird. Den Rechnern droht daher nur Gefahr vor lokalen Benutzern oder einzelnen Angreifern, die einen Weg durch den Firewall gefunden haben.

Besitzt eine Institution mehrere Sub-Netze, die unterschiedliche Sicherheitslevel haben, kann ein Firewall dazu benutzt werden, Übergänge zwischen diesen Netzen zu beschränken. In einer Universität könnte z.B. ein Firewall das Verwaltungsnetzwerk, in dem Angestellten- und Studierendendaten aufbewahrt werden, vor Zugriffen anderer universitätsinterner Rechner schützen. Diese internen Firewalls besitzen die gleichen Eigenschaften wie Internet-Firewalls. Die unterschiedlichen Sicherheitslevel zwischen eigenen Subnetzen können aber differenzierter betrachtet werden als Verbindungen zum kompletten Internet. Daher genügt es häufig, weniger restriktive Formen der im nächsten Abschnitt beschriebenen Architekturen zu benutzen. Eine Diskussion der unterschiedlichen Kriterien für interne Firewalls findet man in [ChZw95; S. 82 ff.].

3 Architekturen von Firewalls

Es gibt eine Reihe von Architekturen, um die gewünschten Charakteristika der Firewalls zu realisieren. Sie unterscheiden sich in der Sicherheit gegenüber Angriffen von außen, im Arbeitsaufwand des Administrators und den Kosten.

Zunächst sollen häufig verwendete Komponenten vorgestellt werden, die in den einzelnen Architekturen zum Einsatz kommen. Danach werden der Aufbau und die Vor- bzw. Nachteile einiger Firewallarchitekturen beschrieben.

3.1 Komponenten

Es gibt einige wichtige Kriterien zur konkreten Auswahl der Komponenten, die später den Firewall bilden sollen. Diese Kriterien gelten sowohl für die verwendete Hardware als auch für die Software wie Betriebssystem u.ä. Da die Einrichtung eines Firewalls teilweise viele Modifikationen der eingesetzten Komponenten erfordert, sollte bereits genügend Erfahrung im Umgang mit diesen Geräten bzw. mit der verwendeten Software vorhanden sein. Beim Einsatz eines Unix-basierten Firewalls kann ein Administrator, der bisher ein PC-Netz verwaltet hat, die spezifischen Probleme und Risiken des Betriebssystems z.B. nur schwer erkennen. Hierdurch können Sicherheitslücken bereits bei der Einrichtung der Komponenten entstehen. Außerdem ist ein schnelles Eingreifen im Falle einer Fehlfunktion oder eines erfolgreichen Einbruchs in das System nur möglich, wenn der Administrator mit der Funktionsweise der Geräte und Software vertraut ist.

Wichtig ist auch die Bedienerfreundlichkeit. Die Absicherung eines Firewalls sollte nicht durch komplizierte Handhabung erschwert werden. Hierdurch können Fehler oder Schwächen in der Konfiguration der Komponenten leichter erkannt und behoben werden. Der Administrator sollte sich auf die Umsetzung des Sicherheitskonzepts konzentrieren können, die Komponenten sollen dies unterstützen und nicht durch überflüssige Komplexität behindern. Auch für die Benutzersicht ist zu bewerten, ob die Komponenten die Verwendung der freigegebenen Dienste unnötig erschweren.

Eine genaue Analyse kann helfen, die Kosten gering zu halten. Häufig bestimmt die Geschwindigkeit der Internetübertragung die benötigte Leistungsfähigkeit der Komponenten. Da diese im Vergleich zur Geschwindigkeit der Hardware gering ist, kann meistens auf preiswertere Geräte im mittleren Leistungsbereich zurückgegriffen werden.

3.1.1 Paketfilter

Im Internet werden die Informationen in Form von Paketen übertragen. Bei diesen Paketen handelt es sich um logische Einheiten, ein Paket ist die Grundstruktur der Informationsübertragung via Internet. Passen nicht alle Daten in ein einzelnes Paket, werden sie auf mehrere Pakete aufgeteilt. Jedes Paket besteht aus mehreren Schichten, die unterschiedliche Ebenen des Internet-Protokolls darstellen. Jede dieser Schichten besitzt einen Kopf und einen Datenteil. Im Kopf werden die für diese Schicht wichti-

gen Informationen gespeichert, der Datenteil enthält auf der untersten Ebene die zu übertragenden Daten, in den höheren Ebenen kapselt er den Kopf- und den Datenteil der jeweils darunterliegenden Schicht. Details zu der hier stark verkürzten Übersicht zur Datenübertragung im Internet und dem *TCP/IP*[72]-Schichtenmodell finden sich im Vortrag „Grundlagen des Internets und des TCP/IP Stacks“.
Paketfilter untersuchen sämtliche Pakete, die sie erreichen. Grundlage der Untersuchung sind vom Administrator festgelegte Regeln, durch die der Filter entscheiden kann, ob ein Paket weitergeleitet oder vernichtet wird. Für Firewalls sind dafür die Kopfteile der IP- und der darunterliegenden TCP- bzw. *UDP*[73]-Schicht interessant. Die IP-Schicht enthält u.a. die IP-Quell- und Zieladresse des Pakets und welches Protokoll im Datenteil verwendet wurde (TCP, UDP oder *ICMP*[74]). Der Kopfteil der TCP- bzw. UDP-Schicht enthält die *Port*nummer[75], von der das Paket abgesendet wurde und die Nummer, an die es weitergeleitet werden soll.
Bereits aufgrund der Quelladresse des Pakets können Paketfilter Entscheidungen treffen. Es können z.B. nur Pakete von einigen im Regelsatz freigegebenen Rechnern den Filter passieren, alle anderen werden vernichtet. Diese einfache Form der Kontrolle besitzt jedoch wenig Flexibilität, denn es ist nicht möglich, bestimmte Dienste für einzelne Rechner zu blockieren. Außerdem werden Angriffe durch gefälschte Quelladressen und *Man in the middle* Attacken (s. Vortrag „Angriffsstrategien“) möglich.
Betrachtet der Paketfilter auch die darunterliegenden Schichten, kann er speziellere Bewertungen vornehmen.
Die TCP-Schicht enthält die Quell- und Zielportnummer des Pakets, und eine Reihe von TCP-Flags, von denen nur das *Acknowledge*-Flag (ACK) für den Filter von Bedeutung ist. Anhand der Portnummern kann der Dienst ermittelt werden, der das Paket zur Übertragung nutzt. Beispielsweise ist ein TCP-Paket mit einer Quell-Portnummer größer 1023 und der Zielnummer 23 Teil einer *Telnet*-Verbindung, mit Zielnummer 80 Teil einer *HTTP*-Verbindung. Die Portnummern vieler Internet-Dienste sind festgelegt, und werden von sämtlichen Betriebssystemen eingehalten. Lediglich Ports über 1023 sind frei verfügbar: sie werden (in zufälliger Reichenfolge) von Clients benutzt, um ihre Server auf den reservierten, nur vom Systemadministrator belegbaren Ports anzusprechen. In dem freien Bereich können aber auch Server auf Verbindungen warten, die von normalen Benutzern gestartet wurden (z.B. private *HTTP-Server*, …).

---

[72] Transmission Control Protocol, Internet Protocol. Protokolle mit unterschiedlichen Aufgaben für die Versendung von Daten über das Internet.
[73] User Datagram Protocol. Ein weiteres Protokoll, vergleichbar mit TCP, hat aber weniger Kontrollfunktionen.
[74] Internet Control Message Protocol. Mit diesem Protokoll werden Nachrichten über den Zustand von Rechnern bzw. Netzen übertragen, z.B. Fehlermeldungen wie „host unreachable“.
[75] Ports sind logische Einheiten, über die das Betriebssystem feststellt, welcher lokale Dienst angesprochen werden soll.

Da es sich bei TCP um ein verbindungsorientiertes Protokoll handelt, das den beiden kommunizierenden Rechnern die korrekte Versendung einzelner Pakete garantiert, ist als Kontrollmöglichkeit das ACK-Flag im TCP-Kopf vorhanden. Dieses Flag besitzt im ersten Paket der Verbindung den Wert 0, beide Rechner setzen in ihrer Antwort auf ein erhaltenes Paket das ACK-Flag auf 1. Nachdem die Verbindung zwischen zwei Rechnern aufgebaut wurde und das erste Paket mit ACK=0 übertragen wurde, enthalten also alle weiteren Pakete den Wert 1 in diesem Flag. Die Filter müssen nur bei Paketen mit ACK=0 entscheiden, ob es weitergeleitet oder vernichtet werden soll: Passieren einzelne Pakete einer Verbindung den Filter, das Paket mit dem ACK-Wert 0 erreicht den Zielrechner aber nicht, vernichtet er von sich aus nach einer gewissen Zeitspanne die Pakete und verhindert damit die Ausführung der unerwünschten Aktion. Dieses Verfahren hat außerdem den Vorteil, daß nicht sämtliche Pakete protokolliert werden müssen. Es erscheinen jeweils nur die Daten beim Administrator, die die Kommunikation eingeleitet haben, zusammen mit der Information, ob der Filter diese Verbindung unterbunden hat oder nicht.

Das ACK-Flag ermöglicht auch die Unterscheidung, ob ein interner Rechner oder ein Rechner im Internet die Kommunikation zwischen beiden begonnen hat. Bei einer Anfrage eines lokalen Rechners an einen Rechner im Internet besitzt ein herausgehendes Paket (s.u.) den ACK-Wert 0, bei einer umgekehrten Anfrage enthält ein eingehendes Paket diesen Wert. Hierdurch ist es möglich, Anfragen aus dem Internet zu verhindern, die Antworten (mit ACK=1) auf Anfragen lokaler Rechner aber durchzulassen.

Das ACK-Flag kann nicht durch Hacker permanent auf 1 gesetzt werden, da bei externen Anfragen immer ein ACK=0 Paket vorhanden sein muß, und ist deshalb für Angriffe nutzlos.

Bei UDP handelt es sich um ein verbindungsloses Protokoll, deshalb ist das ACK-Bit hier nicht vorhanden. Das macht es schwierig, in den Filterregeln Anfragen aus dem Internet und Antworten auf Anfragen aus dem lokalen Netz zu unterscheiden. Verschiedene kommerzielle Firewalls merken sich herausgehende UDP-Pakete und lassen nur Pakete herein, deren Quelladresse mit der Zieladresse des Anfragepakets und deren Zieladresse mit der Quelladresse des Anfragepakets übereinstimmen. Dieses selbständige Anpassen der Filterregeln wird als dynamisches Routing bezeichnet.

Einige kommerzielle Firewalls analysieren neben den Köpfen der oberen Schichten mittlerweile auch die Daten der Applikationsschicht (z.B. Firewall-1 von CheckPoint).

Paketfilter erlauben eine sichere Kontrolle des Datenverkehrs, und sind normalerweise für die Benutzer transparent. Wird ein Paketfilter an der einzigen Verbindung zwischen Internet und dem lokalen Netz eingesetzt, kann er sämtliche Rechner des lokalen Netzes sichern. Ihre Regelsätze sind jedoch häufig schwer zu konfigurieren, es gibt wenig Möglichkeiten, die Korrektheit des gesamten Regelsystems zu überprüfen.

Bei der Erstellung der Regeln muß bedacht werden, daß es sich bei der Kommunikation zweier Rechner meistens um bidirektionalen Verkehr handelt: ins Internet gesendete Pakete erzeugen dabei auch zurückkommende Antwortpakete. Um die Pakete unterscheiden zu können, sind, neben den bereits oben beschriebenen protokollspezifischen Daten (ACK-Flag), auch Informationen darüber nötig, auf welchem Weg sie den Filter erreichen. Hierzu muß der Filter feststellen können, an welchem seiner Interfaces (das mit dem Internet oder das mit dem internen Netz verbundene) die Pakete ankommen. Auf diese Weise können eingehende und herausgehende Pakete unterschieden und Angriffe durch vorgetäuschte interne Quelladressen verhindert werden.

Schon einfache Regelsysteme können schwer zu überblicken sein. Dies soll anhand eines Beispiels kurz verdeutlicht werden, das dem Buch [ChZw95, S. 182ff.] entnommen wurde und dort wesentlich detaillierter behandelt wird.

Der Paketfilter soll so konfiguriert werden, daß E-Mails im internen Netz empfangen und versendet werden können. Das zugrundeliegende Protokoll für diesen Dienst ist *SMTP*[76]. SMTP-Server benutzen den Port 25, die Clients verwenden einen freien Port über 1023. Für jede Kommunikationsrichtung (Empfang wie Versenden von E-Mails) werden jeweils ein- und ausgehende Pakete benötigt. Die Spalten der Tabelle geben die Kriterien zur Bewertung eines Pakets an, die Zeilen enthalten jeweils eine Regel und werden von oben nach unten bearbeitet, bis eine passende Regel gefunden wird.

Regel	Richtung	Quelladresse	Zieladresse	Protokoll	Quellport	Zielport	Aktion
A	eingehend	extern	intern	TCP	> 1023	25	weiterleiten
B	ausgehend	intern	extern	TCP	25	> 1023	weiterleiten
C	ausgehend	intern	extern	TCP	> 1023	25	weiterleiten
D	eingehend	extern	intern	TCP	25	> 1023	weiterleiten
E	-	-	-	-	-	-	vernichten

Regeln A und B sind für das Empfangen von E-Mails zuständig, C und D für das Versenden. Alle anderen Pakete werden durch Regel E vernichtet (- bedeutet, daß an dieser Stelle beliebige Werte stehen können). Dieser Regelsatz ist bereits relativ sicher, er läßt lediglich Kommunikation zwischen Port 25 und einem Port über 1023 zu. Hierdurch werden auch Pakete blockiert, die z.B. das X Window-System erreichen sollen, solche Pakete besitzen einen beliebigen Quellport über 1023 und den Zielport 6000, den der X-Server standardmäßig benutzt. Das X Window-System läßt aufgrund seines offenen Designs einige unerwünschte Operationen zu (z.B. das Simulieren von Eingaben eines Benutzers durch einen anderen). Pakete, die diese Eigenschaften besitzen und eventuell einen Angriff auf einen X-Server darstellen, sollen in jedem Fall blockiert werden. Dies wird durch den obigen Regelsatz auch getan. Dieser versagt jedoch, wenn die Pakete eines Hackers vom Port 25 stammen. Dies ist z.B. möglich, wenn der SMTP-Server auf dem Angriffsrechner gestoppt wurde und dieser Port dann

[76] Simple Mail Transfer Protokoll.

für die Versendung der Pakete benutzt wird. Regel D läßt eingehende Pakete vom Port 25 an den Port 6000 zu, und Regel C erlaubt ausgehende Antwortpakete des Servers von Port 6000 an Port 25. Schon bei diesem übersichtlichen Regelsystem, das nur den E-Mail-Dienst berücksichtigt, ist also schon Vorsicht geboten. Vollständig abgesichert werden die Regeln durch Hinzunahme des ACK-Flags, was die folgende Tabelle liefert.

Regel	Rchtg.	Quelladresse	Zieladresse	Protkoll	Quellport	Zielport	ACK	Aktion
A’	ein	extern	intern	TCP	> 1023	25	-	weiterl.
B’	aus	intern	extern	TCP	25	> 1023	1	weiterl.
C’	aus	intern	extern	TCP	> 1023	25	-	weiterl.
D’	ein	extern	intern	TCP	25	> 1023	1	weiterl.
E’	-	-	-	-	-	-	-	vern.

Angriffe auf das X Window-System (oder andere Server, die Ports über 1023 benutzen) werden nun von Regel D’ verhindert. Da das erste Paket einer Anfrage immer das ACK-Flag auf 0 gesetzt hat, erfüllt es Regel D’ nicht mehr, und wird deshalb von Regel E (da keine andere paßt) vernichtet. Der E-Mail-Dienst kann ohne Einschränkungen angeboten werden. Dieses Beispiel zeigt auch, daß durch das Fehlen des ACK-Flags UDP-basierte Dienste nicht sicher gefiltert werden können.
Realisiert werden Paketfilter durch normale Rechner mit spezieller Software oder durch *Router*[77], die das Filtern von Paketen unterstützen.

3.1.2 Dual homed hosts und Proxy-Server

Dual homed hosts sind Rechner mit zwei oder mehr Verbindungen zu unterschiedlichen Sub-Netzen. Im Unterschied zu *Gateways* darf auf diesen Rechnern keine automatische Weiterleitung von Paketen zwischen den Netzen stattfinden (*IP-Forwarding*). Hierdurch soll ein direkter Zugriff von lokalen Clientprogrammen auf Server außerhalb des zu schützenden Netzes verhindert werden.
Auf *Dual homed hosts* laufen häufig *Proxy-Server*, die Zugriffe zwischen den verschiedenen Netzen abwickeln können. Client und Server kommunizieren hierbei nicht über eine gemeinsame Verbindung, sondern über den *Proxy*. Dieser leitet Anfragen im Auftrag der Clients an den entsprechenden Server weiter. Der Client hat bei einem vollständig transparenten *Proxy* den Eindruck, als würde er direkt mit dem adressierten Server kommunizieren. Der Server dagegen sieht nur Anfragen, die scheinbar alle vom *Proxy*-Rechner stammen, und behandelt diesen als seinen Client. Die an den *Proxy* gerichteten Antworten des Servers werden von ihm an den wahren Client weitergeleitet.

[77] Router sind Geräte, die für die optimale Weiterleitung der einzelnen Pakete im Internet über verschiedene Sub-Netze sorgen.

Durch diesen Ansatz ist eine Kontrolle von Verbindungen auf Applikationsebene möglich. Das bedeutet, daß ein Dienst zwar angeboten wird, aber nur die Operationen über den *Proxy-Server* zur Verfügung stellt, die als sicher gelten. Die Deaktivierung des *IP-Forwardings* verhindert dabei, daß modifizierte Pakete durch den *Dual homed host* gelangen, die vom *Proxy-Server* abgefangen werden sollten. Ist ein *Proxy* mit dem Protokoll eines Dienstes vertraut, ermöglicht er eine leichter verständliche Protokollierung der Verbindungsdaten, als dies z.B. ein Paketfilter erledigen kann. Der Administrator muß bei der Kontrolle nicht mehr aus den Internet-Adressen und Portnummern der Pakete herausfinden, welche Aktion durchgeführt wurde, sondern bekommt diese Information bereits durch den *Proxy* geliefert, der die Operationen eines Dienstes erkennt.

Der Einsatz eines *Proxy-Servers* erfordert Änderungen an der Client-Software oder spezielles Verhalten der Benutzer. Die Client-Software muß dahingehend angepaßt werden, daß sie keine direkte Verbindung zum adressierten Server aufbaut, sondern den *Proxy* kontaktiert und ihm die Adresse des gewünschten Servers mitteilt. Bei dieser Lösung ist für die Anwender die indirekte Verbindung relativ transparent, für interne (ohne *Proxy*) und externe Zugriffe (mit *Proxy*) müssen aber unterschiedliche Client-Programme eingesetzt werden, was bedienerunfreundlich ist. Zur Anpassung der Clients müssen Programmierkenntnisse und zusätzliche Software (Bibliotheken, Compiler, …) vorhanden sein.

Die Alternative sind spezielle Benutzereingaben, hierdurch können geeignete Standardclients verwendet werden, die nicht weiter verändert werden müssen. Durch den zusätzlichen Aufwand zum Aufbau der Verbindung ist diese Lösung für Anwender unattraktiv, und der Adminstrator muß eine Beschreibung der geänderten Funktionalität der Dienste zur Verfügung stellen. Eine frei verfügbare Sammlung von *Proxies*, die auf diesem Ansatz basieren, ist das TIS-Firewall-Toolkit für Unix-Systeme. Der *Telnet-Proxy* dieses Pakets erfordert, daß Benutzer statt der Adresse des Zielrechners den *Proxy-Server* angeben. Der fordert statt der Anmeldungsaufforderung (wie bei einer regulären *Telnet*-Verbindung) die Eingabe des gewünschten Zielrechners. Erst dadurch wird für den Anwender der Dienst nutzbar.

Ein anderes frei verfügbares *Proxy*-Paket ist SOCKS. Es stellt einen allgemeinen („generischen“) *Proxy* zur Verfügung, auf den mit veränderter Client-Software zugegriffen werden kann. Durch die hohe Verbreitung sind bereits viele Clients an dieses Paket angepaßt, häufig kann bei der Compilierung der im Quellcode vorliegenden Software die Nutzung von SOCKS ausgewählt werden. Bei kommerziellen Produkten läßt sich eine eventuell vorhandene Unterstützung oft zur Laufzeit aktivieren. Durch beide Maßnahmen werden einige Systemaufrufe für Netzfunktionen durch Funktionen der SOCKS-Bibliothek ersetzt, die für den SOCKS-*Proxy* angepaßt sind. Der *Proxy* muß auf Unix-Systemen laufen und kann von Clients sämtlicher Dienste benutzt werden. Die Bibliothek ist auch für andere Plattformen umgesetzt worden, Clients sind somit nicht an Unix-Rechner gebunden.

Die Konfiguration von *Proxy-Servern* erfordert zusätzlichen Aufwand vom Administrator, vor allem, wenn mehrere Dienst-spezifische *Proxies* verwendet werden sollen. Nicht alle Dienste unterstützen

das *Proxy*-Konzept, einige lassen sich leichter z.B. über Paketfilter absichern. Zur Entwicklung eines *Proxy-Servers* sind genaue Kenntnisse des Dienstes erforderlich, Vorgehensweisen anhand von Beispielen sind in [URL-10] oder [ChZw95; Kap. 8] beschrieben.
Proxy-Server werden nicht nur auf *Dual homed hosts* eingesetzt, sondern auch auf sog. *Bastion hosts*.

3.1.3 Bastion host

Einige Firewalls sollen auch kontrollierten Zugriff auf das interne Netz ermöglichen. Hierfür werden *Bastion hosts* benötigt. Das sind speziell gesicherte Rechner, die feindliche Angriffe abwehren und überwachten Zugriff im Rahmen des Sicherheitskonzeptes gewähren sollen. *Bastion hosts* sind die einzigen Rechner, auf die vom Internet aus zugegriffen werden kann, sie schirmen das interne Netz vor direkten Zugriffen ab. Daher sollten sie besonders durch die weiter oben beschriebenen Maßnahmen zur Absicherung einzelner Rechner (*Host security*) geschützt werden. Sie sollten keinerlei Sicherheitslücken für Angriffe bieten, da hierdurch das gesamte innere Netz in Gefahr ist. Müssen sie mit Rechnern im internen Netz kommunizieren, sollten diese ebenfalls stärker abgesichert werden. Eine solche Kommunikation ist z.B. nötig, wenn der *Bastion host* als zentraler E-Mail-Server fungiert, der die eingegangenen Nachrichten an einen Rechner im internen Netz weiterleitet. Von dort werden sie dann an die einzelnen Rechner und Benutzer verteilt.
Soll ein Dienst über das Internet abrufbar sein, wie z.B. ein öffentlicher WWW-Server, der Informationen über die Institution anbietet, wird dieser Dienst meistens über einen *Bastion host* zugänglich gemacht. Der Server läuft auf diesem abgesicherten Rechner, der häufig auch die benötigten Daten besitzt. Verbindungen in das interne Netz werden, wenn überhaupt nötig, nur vom *Bastion host* abgewickelt.
Darüberhinaus können *Proxy-Server* auf diesem Rechner eingesetzt werden, um internen Benutzern einen Zugriff auf das Internet zu ermöglichen. Ein Anmelden von Benutzern auf dem *Bastion host* sollte nicht möglich sein, da hierdurch neue Angriffsmöglichkeiten entstehen (nicht fehlerfreie *login*[78]-Software, Passwort-Attacken, …).
Die Hardware eines *Bastion hosts* muß nicht besonders schnell sein. *Proxy-Server* benötigen in der Regel nicht viel Rechenzeit, und die Datenübertragung ist durch die Geschwindigkeit des Internet-Anschlusses bestimmt. Dadurch wird ein *Bastion host* für einen Einbruchsversuch uninteressant gemacht, da ein mittelmäßiger Rechner ein unattraktives Ziel ist. Sollte es dennoch zu Perfomance-Problemen kommen, können die *Proxy-Server* auch auf mehrere *Bastion hosts* aufgeteilt werden. Wichtiger als die Schnelligkeit ist die Zuverlässigkeit und Stabilität der Hardware.
Bei der Einrichtung eines *Bastion hosts* sind viele Modifikationen nötig. Deshalb sollten hier Hardware und Betriebssystem so gewählt werden, daß Erfahrung im Umgang mit beiden vorhanden ist und

[78] Über ein login meldet sich ein Benutzer bei einem System an.

die Möglichkeit besteht, Komponenten auf einem anderen Rechner zu konfigurieren und erst dann in den *Bastion host* zu integrieren. Außerdem ist bei der Wahl des Betriebssystems zu beachten, daß die benötigten *Proxy-Server* für diese Plattform zur Verfügung stehen. Auch häufig erscheinende Sicherheits-Patches des Herstellers sind von Vorteil.

3.2 Interner Aufbau und Arbeitsweise

3.2.1 Paketfilter *(Screening router)*

Die Verbindung zwischen Internet und dem lokalen Netzwerk stellt bei diesem Ansatz lediglich ein Paketfilter her. Aufgrund der speziellen Aufgabenstellung (Untersuchung und Weiterleitung bzw. Vernichtung von Paketen) werden häufig Router als Paketfilter eingesetzt. Der eingesetzte Router muß die Eingabe von Regelsätzen erlauben, die durch das Sicherheitskonzept der Institution bestimmt sind. Der Aufbau ist in Abb. dargestellt.

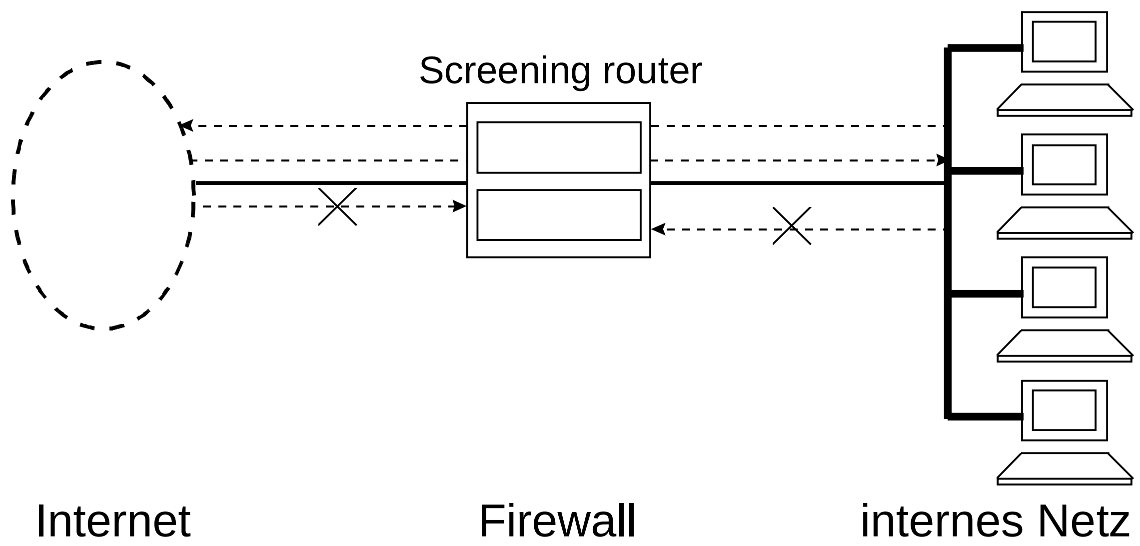

Abb. 2: Paketfilter-Architektur.

Ein Router, der als Paketfilter benutzt wird, heißt auch *Screening router*. Im Unterschied zu einem normalen Router entscheidet er nicht nur aufgrund der Zieladresse des IP-Pakets, ob und wie er das Paket weiterleiten soll, sondern überprüft jedes Paket zusätzlich durch den Regelsatz. Entspricht es nicht den Kriterien der programmierten Regeln, vernichtet ein *Screening router* das Paket, auch wenn er einen Weg zu dessen Ziel finden könnte.

Diese Architektur hat jedoch eine Reihe von Nachteilen. Der *Screening router* ist die einzige Komponente, die das interne Netz nach außen abschirmt. Falls ein Angreifer Zugriff auf den Router bekommt oder eine Fehlfunktion vorliegt, kann auf das Netz zugegriffen werden. Ein nicht gut gesicherter Rechner wird dadurch leicht angreifbar, die Daten dieses Rechners und sämtliche über das interne Netz übertragenen Daten sind gefährdet.

Eine teilweise Freischaltung eines Dienstes ist nicht möglich. Der *Screening router* kann durch die Portnummern des Paketes nur entscheiden, um welchen Dienst es sich handelt, nicht welche Operation ausgeführt oder welche Daten übertragen werden. In einer Filterregel kann nur angegeben werden, daß eine Verbindung zu Port 23 z.B. nicht erlaubt ist, und somit keine *Telnet*-Verbindungen von außen möglich sind. Schon relativ einfache Sicherheitskonzepte können komplizierte Regelsätze bewirken, deren Korrektheit nur schwer zu überprüfen ist.

Das Protokollieren von besonderen Vorkommnissen ist bei Routern nur eingeschränkt möglich. Dies hat einerseits technische Gründe (zusätzliche Hardware erforderlich, eventuell keine Anschlüsse vorhanden), andererseits kann auch nur protokolliert werden, ob ein Paket vernichtet oder weitergeleitet wurde. Da Router auf einer niedrigen Protokollschicht arbeiten, sind solche Einträge schwer lesbar und damit ist nur mit Mühe ein ernster Angriff von einem Fehler zu unterscheiden.

Vorteil ist der niedrige Kostenaufwand für Hardware.

3.2.2 Dual homed host Systeme

Der Firewall besteht bei diesem System nur aus einem *Dual homed host*, dessen Subnetze das Internet und das interne Netz sind (s. Abb.).

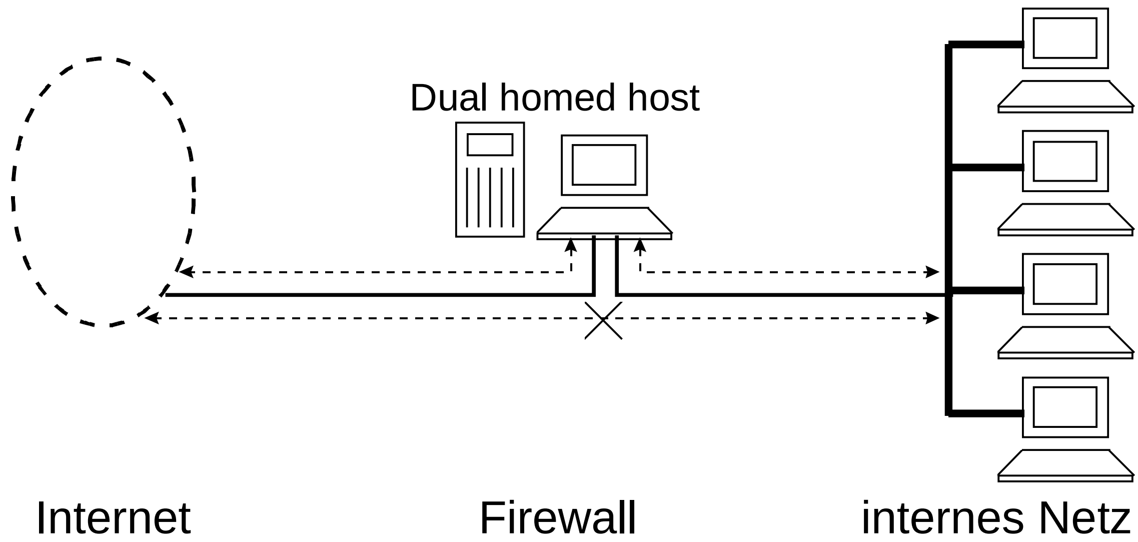

Abb. 3: Dual homed host Architektur.

Lokale Benutzer können auf das Internet über die oben beschriebenen *Proxy-Server* zugreifen. Für jeden Dienst muß ein eigener Server, der die als sicher angesehenen Funktionen zur Verfügung stellt, eingerichtet werden. Solche *Proxies* existieren aber nicht für jeden Dienst bzw. erlauben nur mit zusätzlichem Aufwand der Benutzer eine Verbindung.

Eine Alternative für Dienste, die sich nur schwer über einen *Proxy* anbieten lassen, ist das Aufrufen von Clientprogrammen auf dem *Dual homed host*. Die Benutzer müssen sich hierfür auf diesem Rechner anmelden und können hier eine Verbindung zum Internet aufnehmen. Dies hat jedoch den Nachteil, daß von Benutzern auf diesem Rechner eventuell eigene, vom Administrator als nicht sicher eingeschätzte Server gestartet werden können. Außerdem ist diese Alternative ebenfalls mit zusätzlichem Aufwand und eventuellen Unannehmlichkeiten (z.B. geringe Performance des Rechners bei vielen gleichzeitigen Zugriffen) für die Benutzer verbunden.

Vorteil der *Dual homed host* Architektur sind die geringen Kosten für die Hardware. Außerdem läßt sich die Funktionstüchtigkeit des Firewalls bei dieser Architektur leicht überprüfen: gelangen Pakete mit externen IP-Quelladressen ins interne Netz, existiert eine Sicherheitslücke. Denn dies sollte durch das Abschalten des *IP-Forwardings* und den Einsatz von *Proxy-Servern* (bzw. der Anmeldung auf dem *Dual homed host*) verhindert werden.

Nachteil ist ein hoher Arbeitsaufwand für den Administrator. Dies umfaßt das Einrichten und Überwachen der *Proxy-Server* und des *Dual homed hosts*, Benutzerschulungen zur Verwendung der *Proxies* (allgemein erhältliche Beschreibungen zu den Diensten sind u. U. für *Proxies* nicht verwendbar) oder das Einrichten von Accounts auf dem *Dual homed host* (mit Sicherheitsbelehrung für Benutzer). Abhängig von den spezifischen Eigenschaften eines Dienstes kann es eventuell keine elegante Lösung geben.

3.2.3 Screened host Architektur

Diese Architektur bietet eine Lösung zum Problem, daß einige Dienste am besten über *Proxy-Server* abgesichert werden können, während andere leichter über Paketfilter zu kontrollieren sind. Deshalb werden im *Screened host*-Ansatz Komponenten für beide Sicherungsarten verwendet: neben dem *Screening router* gibt es einen speziellen Rechner, einen *Bastion host*, auf dem *Proxy-Server* laufen können. Abb. zeigt ein Schema dieser Architektur.

Der *Bastion host* ist an das interne Netzwerk angeschlossen. Der *Screening router* sorgt dafür, daß er der einzige Rechner ist, auf den vom Internet aus zugegriffen werden kann. Alle Pakete, die für die auf dem *Bastion host* angebotenen Dienste bestimmt sind, werden weitergeleitet, Pakete für andere interne Rechner oder nicht von außen zugängliche Dienste werden dagegen blockiert.

Zugriffe aus dem lokalen Netz auf das Internet werden über *Proxy-Server* auf dem *Bastion host* erledigt, es besteht aber auch die Möglichkeit, bestimmte Dienste mit direktem Zugriff auf das Internet freizugeben. Hierfür müssen auf dem *Screening router* entsprechende Regeln vorhanden sein, die Pakete der zulässigen Dienste weiterleiten.

Von den *Proxy-Servern* auf dem *Bastion host* erzeugte Pakete müssen ebenfalls durch entsprechende Regeln den Paketfilter passieren können.

Die *Screened host*-Architektur besitzt den Vorteil der größeren Flexibilität gegenüber dem *Dual homed host* Ansatz. Neben *Proxy*-basierten Diensten können auch direkte Verbindungen ins Internet gestattet werden. Ein Router ist im allgemeinen aufgrund seiner Spezialisierung leichter zu schützen als ein Rechner, der eine Vielzahl von Angriffsmöglichkeiten bietet.

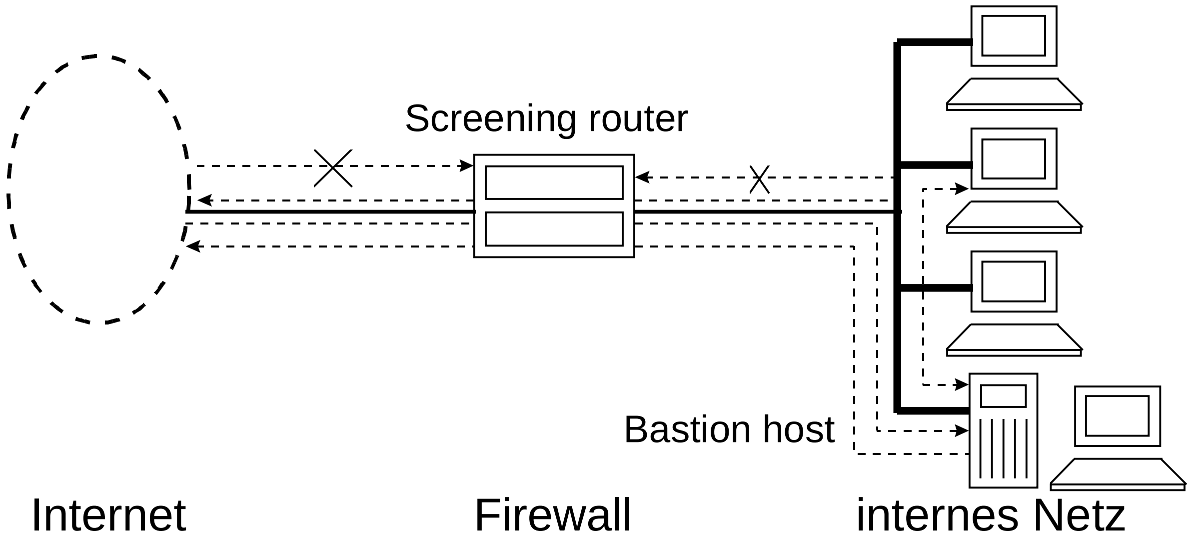

Abb. 4: Screened host-Architektur.

Durch den direkten Anschluß des *Bastion hosts* an das interne Netzwerk besteht jedoch die große Gefahr, daß ein erfolgreicher Angreifer sofort auf das komplette Netz zugreifen kann. Es gibt keine weitere Verteidigungslinie. Auch ein geglückter Angriff auf den Router kann die Sicherheit des gesamten Netzes gefährden.

3.2.4 Screened subnet Architektur

Die *Screened subnet*-Architektur stellt die sicherste der hier vorgestellten Architekturen dar. Sie besitzt mehrere Komponenten, die ein Angreifer überwinden muß, um in das interne Netz zu gelangen. Diesen mehrstufigen Aufbau zeigt Abb. , im folgenden wird die einfachste Form dargestellt. Es gibt eine Reihe von Erweiterungen dieser Architektur, wie z.B. mehrere Zwischennetzwerke (*Perimeter nets*), mehrere *Bastion hosts* für unterschiedliche Server (ein *ftp-Server* und ein *HTTP-Server*, …), usw.

Das *Perimeter net* ist ein Zwischennetzwerk zwischen dem Internet und dem lokalen Netz. Es ist durch den externen und den internen Router von beiden Netzwerken getrennt. An dieses Zwischennetzwerk wird der *Bastion host* angeschlossen. Ein Einbruch in diesen Rechner führt nicht automatisch zu einem Zugriff auf das interne Netz, wie es bei der *Screened host*-Architektur der Fall war. Ein Angreifer muß zusätzlich noch den internen Router überwinden, der einen zusätzlichen Verteidigungswall darstellt. Über das Zwischennetzwerk werden nur Daten vom und zum *Bastion host* übertragen, die nicht so vertraulich sind wie Daten im internen Netz. Wird in den *Bastion host* eingebrochen, kann der Einbrecher nur auf die wenig nützlichen Daten zugreifen.

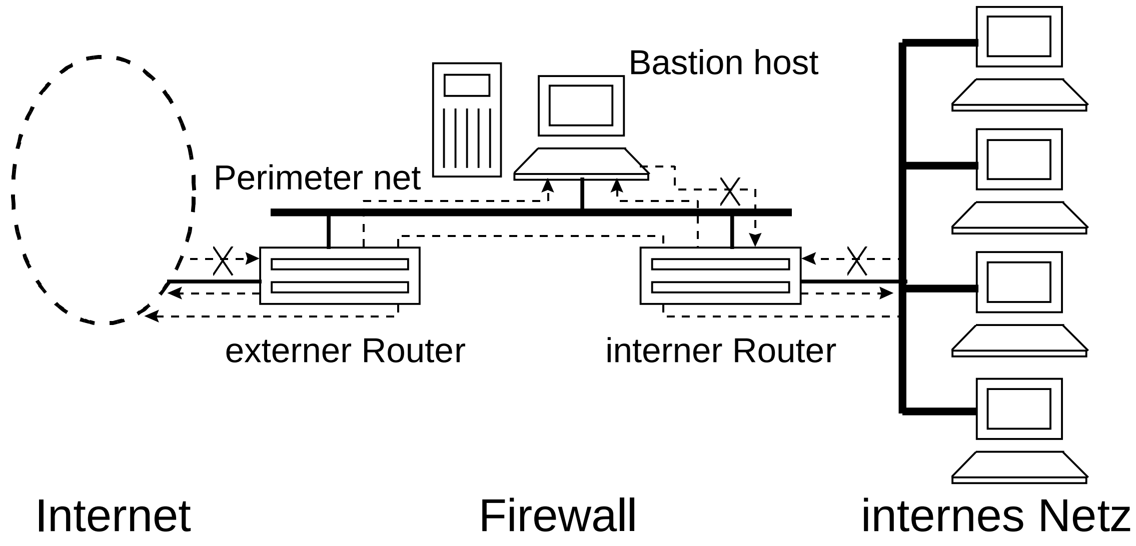

Abb. 5: Screened subnet-Architektur.

Der interne Router ist ein *Screening router* und läßt Verbindungen lokaler Rechner zum *Bastion host* und dem Internet zu. Direkter Zugriff auf das Internet muß ebenfalls vom externen Router weitergeleitet werden, hier müssen also die Filterregeln auf beiden Routern übereinstimmen. Nur für sichere Dienste sollte dieser Weg erlaubt sein. Lokale Zugriffe auf den *Bastion host* werden vom internen Router für Dienste zugelassen, die über *Proxy-Server* auf diesem Rechner angeboten werden. Der externe Router muß hierfür die Pakete ins Internet weiterleiten, die von den *Proxies* erzeugt wurden.

Kommunikation zwischen dem *Bastion host* und Rechnern im internen Netzwerk sollte auf einige wenige Rechner beschränkt werden. Dadurch kann ein Einbruch in den *Bastion host* die Mehrzahl der internen Rechner nicht gefährden, da durch den internen Router nur Verbindungen zwischen dem *Bastion host* und wenigen ausgewählten Rechnern zugelassen wird. Diese sollten ähnlich starken Schutzmaßnahmen unterliegen wie der *Bastion host*.

Der interne Router schützt das interne Netzwerk also vor dem Zwischennetz und dem Internet. Er enthält die Filterregeln, die durch das Sicherheitskonzept vorgegeben sind. Der externe Router wird hauptsächlich dazu genutzt, Angriffe durch gefälschte Pakete aus dem Internet zu verhindern. Solche gefälschten Pakete enthalten als Quelladresse die Adresse des *Bastion hosts* oder interner Rechner. Dadurch können z.B. Pakete durch den internen Router an Rechner im lokalen Netz weitergeleitet werden, da er durch die gefälschte Quelladresse annimmt, sie kämen vom *Bastion host*. Um zusätzliche Sicherheit durch Redundanz zu bieten, kann der externe Router ebenfalls einen Teil der Regeln des internen Routers enthalten.

Wie bereits oben erwähnt, ist diese Architektur die sicherste der hier vorgestellten. Sie verursacht allerdings hohe Kosten durch die eingesetzte Hardware. Die Einrichtung ist schwierig, da ein *Bastion host* mit den *Proxy-Servern* und zwei *Screening router* verwaltet werden müssen.

3.2.5 Weitere Architekturen

Es gibt weitere Architekturen, die teilweise Abwandlungen der oben beschriebenen sind. Aus Kostengründen können z.B. der interne und externe Router zusammengefaßt werden, wenn der verwendete Router flexible Filterregeln zur Verfügung stellt. Beschreibungen zu weiteren Architekturen finden sich z.B. in [ChZw95; Kap. 4], [Luck97], [URL-1] oder [URL-2].

Einige Modifikationen sind nicht zu empfehlen, da sie zu Sicherheitsproblemen führen. Wenn der *Bastion host* z.B. auch die Funktion des internen Routers übernimmt (durch entsprechende Software), wird dadurch bei einem erfolgreichen Angriff auf diesen Rechner sofort der Verkehr des internen Netzes sichtbar. Die Architektur des Firewalls wird durch diese Modifikation nämlich signifikant geändert: aus einer *Screened subnet*- wird eine *Screened host*-Architektur. Mit separatem internen Router ist der *Bastion host* nur mit dem Zwischennetz verbunden, ein Zugriff auf die Daten im internen Netz ist nicht möglich. Wird der Router aber in den *Bastion host* verlagert, besitzt dieser einen physikalischen Anschluß an das interne Netzwerk und hat dadurch über das Interface Zugriff auf den internen Datenverkehr. Das Zwischennetzwerk wird nutzlos, denn es sollte als zusätzliche Sicherheitsschicht von Internet und internem Netz getrennt sein.

Die Gefahren einiger weiterer Abwandlungen werden ebenfalls in Kapitel 4 von [ChZw95] behandelt.

4 Planung eines Firewalls

Grundlegend für die Auswahl eines Firewall-Produkts ist das darunterliegende Betriebssystem. Hier stehen im wesentlichen UNIX-basierte Systeme (Solaris, HP-UX, Linux, …) und Windows NT zur Auswahl. Wie bereits erwähnt, sollte der Administrator bereits Erfahrung mit dem einzusetzenden Betriebssystem besitzen. Denn die Einrichtung und Absicherung eines Firewalls ist nur möglich, wenn die spezifischen Vorgehensweisen zur Administration und die sicherheitsrelevanten Schwächen des Systems bekannt sind. [ChZw95] und [Mraz97] tendieren beide zu Unix: diese Systeme sind bereits lange Zeit im Einsatz, auch bei der Anbindung ans Internet. Viele Internet-Dienste sind unter Unix entwickelt worden, und deshalb gut in das Betriebssystem integriert. Außerdem existiert eine Vielzahl von Tools, die die genaue Überwachung des Systems bis in die Kernel-Ebene ermöglichen.

Da die Quellen einiger alter Unix-Versionen (auf denen die heutigen häufig basieren) frei verfügbar sind, sind bereits viele Sicherheitslöcher entdeckt und behoben worden. Das gilt ebenfalls für viele freie Firewall-Implementierungen für Unix: sie sind durch die Verfügbarkeit des Quellcodes gut analysiert und seit langer Zeit im Einsatz.

Besitzt ein Administrator (z.B. in einem rein PC-basierten Netzwerk) keinerlei Unix-Erfahrung, ist in diesem Fall sicher Windows NT die bessere Wahl.

Nach der Entscheidung für ein spezielles Betriebssystem stehen die dafür verfügbaren Firewall-Produkte zur Auswahl. Es gibt sowohl frei verfügbare Software (wie die bereits erwähnten Pakete SOCKS und TIS-Firewall-Toolkit), aber auch kommerzielle Systeme. Die einzuplanenden Kosten umfassen: den Anschaffungspreis der Soft- und Hardware, die Kosten für die Evaluierung verschiedener Systeme, die Einrichtung des Firewall und die Kosten für die Pflege und Administrierung.

Für die kostenlose Software spricht neben dem finanziellen Aspekt, daß sie häufig eingesetzt und analysiert wird, und dadurch ein hohes Maß an Sicherheit bietet. Die Nachteile sind die im Vergleich zu kommerziellen Systemen umständlichere Bedienung und eingeschränkte Funktionalität.

Kommerzieller Systeme bieten *Proxy-Server*, die für die Benutzer völlig transparent sind. Diese Firewall-Systeme bieten außerdem erweiterte Fähigkeiten gegenüber der freien Software. Beispiele sind die Analyse des Applikationsdatenteils bei Paketfiltern oder ein Virencheck bei *ftp*-Übertragungen oder E-Mail-*Attaches*[79]. Durch grafische Benutzeroberflächen ermöglichen sie eine übersichtlichere Konfiguration, was auch ein Vorteil für die Sicherheit des Systems bedeuten kann und die Kosten für die Administrierung senkt.

Allerdings fällt eine Sicherheitsbewertung oft schwer, da der Quellcode nicht frei verfügbar ist und nur vom Hersteller erweitert werden kann. Vergleichstests (wie z.B. [LuSc97], [URL-8], [URL-9]) zeigen, daß einige kommerzielle Firewall-Produkte nicht fehlerfrei und trotz grafischer Oberflächen unübersichtlich zu konfigurieren sind. [URL-8] enthält außerdem einen Vergleich der Performance der getesteten System, hier zeigten sich erhebliche Unterschiede. Die eigene Evaluierung eines Firewalls wird dadurch erschwert, daß hierzu viel Sachkenntnis über die zugrundeliegenden Protokolle und Techniken nötig ist, und außerdem bei gründlichem Vorgehen lange dauern kann.

Institutionen, die nicht über einen speziell ausgebildeten Administrator verfügen, können auf die Hilfe von Beratungsfirmen zurückgreifen, die die Planung, Einrichtung und Pflege des Firewalls übernehmen. Eine Übersicht findet sich z.B. bei [URL-3].

Laut [URL-4] sind für Firewall-Hardware Kosten von 6.000 DM bis 2.000.000 DM, abhängig von Ausbau und Komplexität des Systems, zu veranschlagen. Die Kosten für Installation werden hier mit ca. 2.000 DM bis 100.000 DM angegeben, es sind durchschnittlich 1-50 Manntage einzuplanen.

Firewall-Software für mittlere Netzwerke (ca. 50 bis 200 Benutzer) kostet, häufig nach der Anzahl der Benutzer gestaffelt, zwischen 6.700 DM und 60.000 DM ([LuSc97], [URL-5]). Beratungsfirmen verlangen als Stundenlohn für die Einrichtung zwischen ca. 150 DM und 250 DM [URL-6], [URL-7]. Die Kosten für eine monatliche Wartung betragen in [URL-6] 1% des Softwarekaufpreises.

5 Was können Firewalls leisten und was nicht?

Firewalls sorgen dafür, daß vertrauliche Daten nicht allgemein zugänglich sind und bieten einen gewissen Schutz gegen versehentliche Freigabe dieser Daten. Durch Überprüfung aller eingehenden und aller nach außen gerichteten Verbindungen können Firewalls die meisten mutwilligen oder versehentlichen Versuche der Freigabe vertraulicher Daten verhindern.

[79] E-Mail-Attaches erlauben das Anhängen von Dateien an E-Mails.

Firewalls erlauben die Überwachung der Sicherheit eines lokalen Rechnernetzes von einem einzigen Beobachtungspunkt. Administratoren können sich bei ihren Überprüfungen auf diesen Punkt konzentrieren, um besondere Vorkommnisse zu entdecken. Gäbe es mehrere Stellen, an denen eine Verbindung ins Internet bestünde, müßten die Protokolldaten dieser Übergänge miteinander verglichen werden, um einen Angriff entdecken zu können. Denn die Vorbereitung eines Angriffs und der Angriff selbst könnten über verschiedene Verbindungsstellen geschehen. Um den Angriff nachvollziehen zu können, müßten die Protokolldaten in chronologischer Reihenfolge zusammengefügt und schließlich ausgewertet werden. Daraus resultiert eine schlechtere Übersicht und ein erhöhter Arbeitsaufwand gegenüber einer einzigen Verbindung, die ein zentrales Protokoll ermöglicht.

Dieses zentrale Protokoll kann aber nicht nur zum Entdecken von Angriffen benutzt werden. Da sämtliche Verbindungen zwischen dem Internet und dem internen Netz durch den Firewall geleitet werden, können die Protokolldaten auch zur Erstellung einer Nutzungsstatistik dienen. Diese erlaubt eine Kostenkontrolle, liefert aber auch Daten über die Verwendung einzelner Dienste. Auf diese Weise können selten oder nur von wenigen Anwendern benötigte Dienste anders abgesichert werden als solche, die häufig verwendet werden und deshalb komfortabler in der Bedienung sein müssen.

Firewalls sind flexibel und richten sich nach dem individuellen Sicherheitskonzept, das abhängig von den Erfordernissen und den speziellen Einschätzungen verschiedener Gruppen der Institution ist. Die bekannten und verbreiteten Dienste lassen sich über ein Firewall absichern, die konkrete Realisierung wird durch die spezifischen Eigenschaften des Dienstes und die Erfordernisse der Institution (Sicherheit vs. Bedienkomfort) bestimmt.

Ein Firewall ist aber auch kein Allheilmittel. In einer Reihe von Fällen kann er keinen oder nur eingeschränkten Schutz bieten.

Nur wenn Daten durch den Firewall fließen, kann er sie analysieren und weiterleiten bzw. blockieren. Er kann nicht vor Angriffen über weitere Verbindungen des internen Netzes mit dem Internet schützen, die nicht abgesichert sind.

Firewalls können nicht vor Angriffen interner Benutzer schützen. Durch ihre Zugangsberechtigung ist es für diese Benutzer möglich, Konfigurationsdateien zu lesen und dadurch schneller Sicherheitslücken zu entdecken oder Programme zu starten, die zur Lahmlegung bzw. zum Absturz des Rechners führen. Solche ungewollten Zustände werden häufig unbeabsichtigt durch Bedienfehler oder Unachtsamkeit hervorgerufen, auch hier kann ein Firewall nicht helfen. Hat ein Angreifer einen Weg in das interne Netz gefunden, können weitere Aktionen durch den Firewall ebenfalls nicht verhindert werden. Gegen Angriffe interner Benutzer und erfolgreicher Eindringlinge hilft nur die Absicherung der einzelnen Rechner.

Die ständige Überwachung und Anpassung des Firewalls ist nötig, um neuartige Attacken verhindern zu können. Es ist nicht möglich, den Firewall von vornherein so zu konfigurieren, daß er sämtliche Einbruchsmöglichkeiten verhindert. Eine restriktive Freigabe von Diensten kann viele Angriffe verhindern, aber nicht alle Lücken schließen. Nur durch die ständige Kontrolle der Protokolldateien und die Analyse von Einbruchsversuchen können Firewalls Sicherheit bieten.

Ein Firewall kann nur eingeschränkt vor Viren schützen. Da es viele verschiedene Möglichkeiten der Versendung von Programmcode gibt (über *ftp*, E-Mail-Attach, gepackt, …), ist ein automatisches Erkennen von Viren nur sehr eingeschränkt möglich. Einige kommerzielle Firewalls bieten Filter für bestimmte Übertragungsarten, wirkungsvoller ist aber ein Virenscanner, der auf dem Rechner des Benutzers läuft. Dieser kann nicht nur die Viren von Übertragungen aus dem Internet entdecken, sondern kann auch auf andere Weise (private Disketten, …) ins System gelangte Viren finden.

6 Literatur- und Quellenverzeichnis

6.1 Printmedien

[ChZw95] D. B. Chapman; E. D. Zwicky. *Building Internet Firewalls*. O'Reilly, Sebastopol (CA, USA), 1995

[Luck97] N. Luckhardt. „Schwer entflammbar“. *c't Magazin für Computertechnik*. Verlag Heinz Heise, Hannover, April 1997

[LuSc97] N. Luckhardt, J. Schmidt. „Trau, schau, wem!“. *c't Magazin für Computertechnik*. Verlag Heinz Heise, Hannover, Juni 1997

[Mraz97] V. Mraz. „Welches Schweinderl hätten's denn gern?“. *c't Magazin für Computertechnik*. Verlag Heinz Heise, Hannover, Juni 1997

6.2 Elektronische Dokumente

[URL-1] U. Ellermann, Stand 1994/95 (zuletzt gesichtet am 3.2.1998)
`http://www.cert.dfn.de/team/ue/fw/workshop/`

[URL-2] M. Ranum, Stand 1995 (zuletzt gesichtet am 3.2.1998)
`http://www.clark.net/pub/mjr/pubs/fwfaq/`

[URL-3] DFN-CERT, Stand 1997 (zuletzt gesichtet am 3.2.1998)
`http://www.fwl.dfn.de/fwl/fw/fw-prod.html`

[URL-4] S. Strobel, Stand Mai 1997 (zuletzt gesichtet am 3.2.1998)
`http://www.centaur.de/Internet/Sicherheit/fw/tsld069.htm`

[URL-5] DEC, Stand 1997 (zuletzt gesichtet am 27.1.1998)
`http://avsoft6.pa-x.dec.com/msg/firewall/products/techview/`

[URL-6] Siemens-Nixdorf, Stand Juli 1997 (zuletzt gesichtet am 3.2.1998)
`http://www.sbs.at/is/services/secpreis.htm`

[URL-7] oops GmbH, Stand 1995/96 (zuletzt gesichtet am 27.1.1998)
http://www.freising-pop.de/tarife/tar_dien.html

[URL-8] Data Communcations Magazine, Stand November 1995 (zuletzt gesichtet am 27.1.1998)
http://www.data.com/Lab_Tests/Firewall.html

[URL-9] Data Communications Magazine, Stand März 1997 (zuletzt gesichtet am 27.1.1998)
http://www.data.com/lab_tests/firewalls97.html

[URL-10] G. W. Treese, A. Wolman, Stand Mai 1993 (zuletzt gesichtet am 3.2.1998)
http://www.alw.nih.gov/Security/FIRST/papers/firewall/xthrufw.ps

Electronic Commerce

Rouven Fröleke

1 Einführung

Zur Zeit bietet das Internet ein schier unerschöpfliches Angebot an Informationen, welche größtenteils kostenlos und ohne Gegenleistungen abrufbar sind. Bis zur selbstverständlichen und kommerziellen Nutzung des Internets als universelles Handelssystem für Waren und Dienstleistungen aller Art sind allerdings noch viele Hürden zu nehmen. Neben der noch mangelhaften Akzeptanz dieses Mediums in breiten Bevölkerungsschichten ist u.a. das Fehlen eines einheitlichen und universellen Zahlungssystems ein Grund für diese Situation. Viele Konzerne und Informationstechnologie-Unternehmen haben diese Problematik erkannt und versuchen nun, mit ihren neu entwickelten Zahlungssystemen diese äußerst lukrative Marktlücke zu füllen.

Der elektronische Handel (*Electronic Commerce*) wird den Charakter des Internets stark verändern, womit jedoch nicht zwangsläufig negative Veränderungen gemeint sein müssen - die professionelle Kommerzialisierung kann durchaus positiven Einfluß auf den Inhalt der publizierten Angebote haben (z.B. durch gründlicher recherchierte Informationen).

In diesem Seminarbeitrag sollen die Grundlagen von internetfähigen Zahlungssystemen erläutert und einige bereits realisierte Konzepte vorgestellt werden.

2 Grundlagen elektronischer Zahlungssysteme

Für eine Klassifizierung und Bewertung der aktuell entwickelten Zahlungssysteme, ist eine kurze Einführung in die Grundlagen des elektronischen Zahlungsverkehrs zwingend notwendig. Es soll ein Überblick über die wichtigsten Eigenschaften elektronischer Zahlungssysteme und die in ihnen verwendeten Basiskonzepte gegeben werden.

2.1 Eigenschaften elektronischer Zahlungssysteme

Elektronische Zahlungssysteme haben eine Vielzahl von Eigenschaften und Voraussetzungen zu erfüllen, um den Anforderungen des elektronischen Zahlungsverkehrs zu genügen. Als wichtigste Eigenschaften lassen sich sicherlich die Schutzvorrichtungen gegen Betrug, Vortäuschung nicht erfolgter Zahlungen oder Mißbrauch von Transaktionsdaten nennen, da insbesondere im Internet der Faktor Sicherheit eine äußerst wichtige Rolle für die Akzeptanz der Zahlungssysteme spielt.

In diesem Zusammenhang sind auch die Kosten der Zahlungssysteme als wichtiger Faktor einzustufen, da sich z.B. nicht alle Zahlungssysteme für alle Transaktionen nutzen lassen (Mikro- und Makrozahlungen).

Aufgrund der Komplexität des Themas und dem eher technischen Themenbereich des Seminars, wird auf rechtliche Aspekte in diesem Seminarbeitrag nicht eingegangen. In [SFE97] ist eine Auflistung von Quellen zu diesem Themenbereich enthalten.

2.1.1 Sicherheit

Die sichere Übertragung der kritischen Daten einer Finanztransaktion ist eine maßgebliche Voraussetzung für die Akzeptanz eines Zahlungssystems im Internet. Ein marktfähiges Zahlungssystem muß vor betrügerischen Aktionen wie Mitschneiden, Umleiten, Vortäuschen und Manipulieren von Transaktionsdaten sicheren Schutz bieten [FuWr97; Kapitel 2].
Ein möglicher Ansatz um den o.g. Mißbrauchsmöglichkeiten entgegenzuwirken ist der Aufbau einer isolierten **Übertragungsinfrastruktur**.
Diese kostspielige und aufwendige Variante wird z.B. für die Kommunikation zwischen Banken und militärischen Einrichtungen benutzt. SWIFT (Society for Worldwide Interbank Financial Transactions) ist ein gutes Beispiel für ein solches isoliertes Netzwerk [Reif96].
Wenn jedoch private Finanztransaktionen und Erwerbsvorgänge über das Internet abgewickelt werden sollen, bleibt nur die Realisierung der Übertragungssicherheit durch kryptographische Verfahren und Methoden.
Bei kryptographischen Verfahren zur sicheren Nachrichtenübertragung bzw. der Nachrichtenverschlüsselung lassen sich zwei Grundtypen feststellen [SFE97]: **symmetrische Verfahren** und **asymmetrische Verfahren.**
Symmetrische Verfahren kodieren und decodieren ein Nachricht stets mit dem gleichen Schlüssel, was bedeutet, daß sowohl Empfänger als auch Sender diesen Schlüssel kennen müssen. Der Austausch dieses Schlüssels muß vor der eigentlichen Nachrichtenübertragung über einen sicheren Kanal erfolgen. Genau dieses Vorgehen stellt jedoch im unsicheren Internet ein schier unlösbares Problem dar, da ein sicherer Kanal nur über andere (sichere) Netze, Kuriere oder aber persönliche Treffen realisiert werden kann.
Einer der größten Vorteile der symmetrischen Verschlüsselung ist die relativ geringe Rechenleistung, die zum Kodieren bzw. Decodieren von Nachrichten benötigt wird.
Ein weit verbreitetes symmetrisches Verschlüsselungsverfahren stellt das amerikanische DES Verfahren dar (= DEA-1-Verfahren), welches auch in aktuellen Zahlungssystemen praktische Anwendung findet (s.u.).
Das wohl bekannteste asymmetrische Verschlüsselungsverfahren ist das von Ron Rivest, Adi Shamir und Leonard Adleman entwickelte RSA-Verfahren. Das besondere bei diesen Verfahren ist die Verwendung unterschiedlicher Schlüssel zur Kodierung und Dekodierung der zu übertragenden Nachrichten. Zur Verschlüsselung wird der sogenannte öffentliche Schlüssel des Kommunikationspartners verwendet, der z.B. auf sogenannten Schlüsselservern via WWW publiziert wird. Die so verschlüsselte

Nachricht kann nur mit dem (geheimen) privaten Schlüssel des Kommunikationspartners dekodiert werden. Die Rechenzeit dieser Verfahren ist allerdings deutlich höher als die der symmetrischen Verfahren, weshalb sie sich zur Kodierung umfangreicher Nachrichten nur bedingt eignen.

Um diesen Nachteil auszugleichen, verwenden moderne Zahlungssysteme wie z.B. SET eine Kombination der o.g. Verfahren, die sogenannten **hybriden Verfahren**.

Hybride Verfahren benutzen sowohl symmetrische, als auch asymmetrische Verschlüsselungsverfahren zur sicheren Nachrichtenübertragung. Das asymmetrische Verfahren wird hierbei nicht zur Kodierung der eigentlichen Nachricht benutzt, sondern lediglich um einen sicheren Kanal für die Übertragung des zur eigentlichen Kodierung verwendeten symmetrischen Schlüssels zu ermöglichen.

Eine weitere wichtige Eigenschaft von sicheren Übertragungsverfahren ist die mögliche **Authentifizierung** der Kommunikationspartner. Hierbei ist allerdings eine wechselseitige Beziehung mit der später noch erläuterten Eigenschaft der Anonymität zu beachten.

Durch Authentifizierungsverfahren ist es möglich, die Identität eines oder beider Kommunikationspartner eindeutig festzustellen. Dies ist z.B. notwendig, um Kaufverträge rechtsgültig zu unterzeichnen oder Liquiditätsnachweise bei übergeordneten Instanzen einholen zu können. Auch setzten einige Zahlungssysteme die Identifizierung vor der Verwendung elektronischer Geldbörsen (*Wallets*) voraus [FuWr97] [RiKo97].

Gängige Verfahren zu Authentifizierung sind u.a.:

- PIN-Nummern
- Paßwortsysteme
- Netzwerkprotokolle
- elektronische Unterschriften

PIN-Nummern und Paßwortsysteme gelten als relativ unsicher, da durch geschickte Ausnutzung menschlicher Schwächen deren Sicherheitspotential drastisch gesenkt werden kann. Als weit verbreitetes Beispiel läßt sich die Verwendung personifizierter PIN's bzw. Paßwörter nennen (Namen von Angehörigen als Paßwort, Telefonnummer als PIN etc.).

Bei PIN-Systemen kommt zusätzlich eine relativ geringe Anzahl von möglichen Kombinationen zum Tragen, wodurch die Wahrscheinlichkeit von Zufallstreffern gegenüber anderen Systemen erhöht wird.

Die o.g. asymmetrischen Verschlüsselungsverfahren bieten jedoch die Möglichkeit, die aus der „realen" Welt bekannten Unterschriften in abgewandelter bzw. elektronischer Form in Internet-Zahlungssystemen zu verwenden. Das heutige Kreditkartensystem benutzt handgeschriebene Unterschriften als Authentifizierungsnachweis – warum also nicht ein bewährtes System in „modernisierter" und verbesserter Form einsetzen.

Der zu übertragenden Nachricht wird hierbei eine elektronische Unterschrift (Signatur) hinzugefügt, die einen eindeutigen Bezug zwischen Absender und Nachricht herstellt. Dazu ist es notwendig, daß

der Absender die Nachricht mit seinem bereits privatem Schlüssel kodiert. Der Empfänger kann nun diese „Unterschrift“ mit dem öffentlichen Schlüssel des Absenders dekodieren und mit der eigentlichen Nachricht (mit seinem öffentlichen Schlüssel kodiert) vergleichen. Er weiß nun, daß die Nachricht tatsächlich von dem Besitzer des öffentlichen Senderschlüssels stammt.
Zusätzlich wird so das Problem der **Integrität** der Nachricht gewährleistet, da eine nachträgliche Manipulation der übertragenden Daten beim Vergleich mit der Signatur auffallen würde.
Um den Rechen- und Übertragungsaufwand zu minimieren wird allerdings nicht die gesamte Nachricht signiert, sondern lediglich eine daraus berechnete Prüfsumme („message digest“). Diese Prüfsumme wird mittels einer Hash-Funktion aus der Nachricht berechnet. Die Wahrscheinlichkeit zwei identische Prüfsummen aus unterschiedlichen Nachrichten zu erzeugen ist dabei möglichst niedrig zu halten. Als gängige Algorithmen sind z.B. MD5, MD4, SHA-1 und CS4 zu nennen.

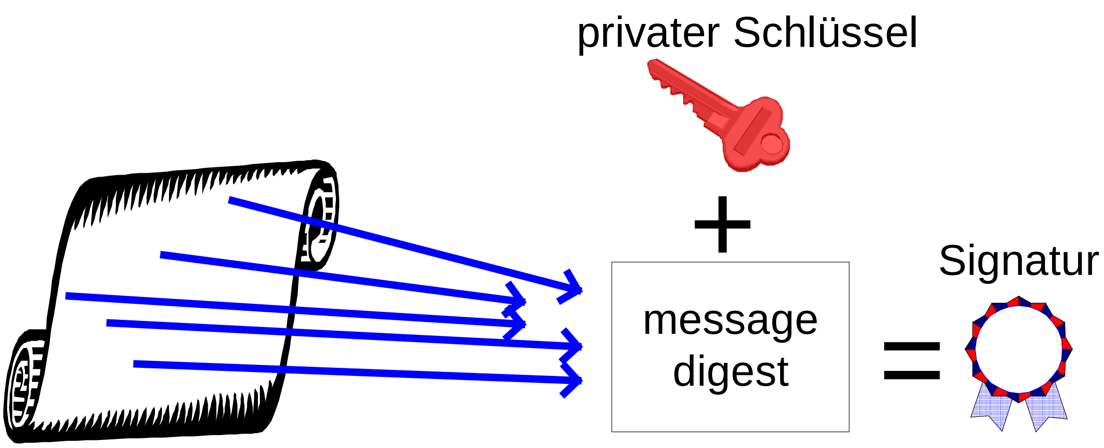

Abb. 1: Digitale Signaturen

Ein weiteres Problem bei der Authentifizierung ist die eindeutige Verbindung zwischen öffentlichem Schlüssel und Schlüsselinhaber. Ein Betrüger kann sich z.B. durch *Spoofing* in die Kommunikation einschalten und einen falschen (von ihm erzeugten) Schlüssel weitergeben. Sämtliche Kommunikation über den so entstandenen (zwar sicheren) Kanal wird nun unbemerkt mit dem Betrüger geführt. Um dieses zu verhindern, gibt es die Möglichkeit, Schlüsselpaare (öffentlich&privat) von einer anerkannten Zertifizierungsstelle beglaubigen zu lassen, um so der willkürlichen Schlüsselerzeugung entgegenzuwirken. Diese Zertifizierungsstellen (Trust-Center, CA = *certification authority*) überprüfen die Identität des Schlüsselinhabers vor Erteilung der Beglaubigung (digitale Signatur des Schlüssels) durch einen sicheren Kanal. In Zukunft kommen z.B. auch staatliche Institutionen für solche Aufgaben in Frage. Die Kommunikationspartner können die erhaltenen öffentlichen Schlüssel nun mit Hilfe der (bekannten) öffentlichen Schlüssel der Zertifizierungsstellen vor der Benutzung auf ihre Echtheit prüfen [SFE97].

2.1.2 Offline-Fähigkeit

Eine weitere Eigenschaft von elektronischen Zahlungssysteme ist die potentielle Offline-Fähigkeit. In reinen Internet-Zahlungssystemen ist deren Relevanz allerdings eingeschränkt und hauptsächlich beim Problem der Skalierbarkeit zu beachten.
Ein offline-fähiges Zahlungssystem kann Betrugsabsichten und Manipulationen ohne ständige Anbindung an eine kontrollierende Instanz entdecken und verhindern (ggf. verzögerte Entdeckung).
Als Beispiel kann hier das später beschriebene, allerdings nicht praktisch umgesetzte „Universal Electronic Cash“ (UEC) dienen.

2.1.3 Anonymität

Bei Zahlungsvorgängen ist die oben erläuterte Authentifizierung nicht immer eine gewünschte Eigenschaft. Banken könnten die laufenden Zahlungsvorgänge automatisch sammeln und auswerten – der gläserne Kunde ist entstanden. Bei Kreditkartenzahlungen unvermeidlich, gibt es z.B. bei bargeldähnlichen Systemen durchaus Möglichkeiten die Zahlung anonym, aber trotzdem sicher abzuwickeln. Das aktuell in Pilotprojekten laufende und später erklärte System „ECash“ der Firma DigiCash benutzt ein Verfahren namens „blinding“, um die Anonymität der Zahlungspartner geheim zu halten.

2.1.4 Unabstreitbarkeit

Die Eigenschaft der Unabstreitbarkeit eines Zahlungssystems gibt sowohl Kunden, als auch Händlern die Möglichkeit, erfolgte Transaktionen nachweisen zu können. Dies kann bei betrügerischen Absichten eines Transaktionspartners notwendig werden.

2.1.5 Transaktionskosten

Zur kommerziellen Nutzung eines Zahlungssystems sind die Betriebskosten möglichst gering zu halten. Die Betriebskosten eines Zahlungssystems ergeben sich im allgemeinen aus den Anschaffungskosten, den Fixkosten und den (variablen) Transaktionskosten. Durch ihre Struktur bedingt eignen sich nicht alle Zahlungssysteme für alle Arten von Zahlungen. Grundsätzlich unterscheidet man in diesem Zusammenhang zwischen **Mikrozahlungen** und **Makrozahlungen**. Bei Mikrozahlungen (Transaktionen bis ca. 5-10 DM) ist darauf zu achten, daß die Transaktionskosten des Zahlungssystems nicht überproportional hoch angesetzt sind und so das Zahlungssystem unwirtschaftlich machen. Andererseits dürfen denjenigen Händlern, welche das elektronische Zahlungssystem nicht allzu intensiv nutzen, keine allzu großen Anschaffungs- und Fixkosten zugemutet werden, um so eine entsprechende Verbreitung des Systems zu fördern. Generell kann man sagen, daß sich Kreditkartensysteme nicht für Mikrozahlungen eignen, da deren Transaktionskosten durch den Aufwand für Buchung, Kommunikation und Verwaltung unwirtschaftlich hoch anzusetzen sind [SFE97; Kap. 4.3]. Zusätzlich zu den Mikro- und Makrozahlungen, ist häufig auch die Rede von sogenannten **Pikozahlungen,** mit denen

Transaktionen von Transaktionswerten unterhalb eines Pfennigs realisiert werden sollen, welche z.B. bei der seitenorientierten Abrechnung von WWW-Seiten anfallen könnten.

Im Zusammenhang mit den Transaktionskosten ist natürlich auch die zu erwartende notwendige Rechenleistung für Kodierung- und Dekodierungsvorgänge zu beachten, welche durch die Anwendung entsprechender Verfahren möglichst gering gehalten werden sollte (RSA, DES, Hashfunktionen).

2.1.6 Skalierbarkeit

Ein gut skalierbares Zahlungssystems stellt durch seinen Systemaufbau sicher, daß im Extremfall beliebige Teilnehmerzahlen möglich sind und keine technischen Flaschenhälse und Beschränkungen eben diese Skalierbarkeit einschränken. Nicht offline-fähige Zahlungssysteme (z.B. ECash) sind hierbei besonders zu beachten, da diese Systeme auf Kontrollfunktionen einer übergeordneten Instanz angewiesen sind, deren maximale Leistungsfähigkeit ggf. durch technische Aspekte (Netzanbindung, Rechnerleistung, hohe Benutzerzahl) eingeschränkt wird.

2.1.7 Bedienbarkeit

Analog zur „normalen“ Anwendersoftware müssen die Benutzeroberflächen von Zahlungssystemen adressatengerecht konzipiert werden. Der Zahlungsvorgang muß sowohl für den Händler, als auch für den Kunden transparent und verständlich gestaltet werden. In diesem Zusammenhang müssen alle relevanten Aktionen (Bestellung der Ware, Einleiten der Zahlung, Quittung bzw. Ablehnung) offensichtlich sein und niemals unbemerkt ausgelöst werden können.

Generell ist in noch stärkeren Maße als bei anderen Softwareprodukten auf einfache Bedienbarkeit zu achten, da eine breite (nicht technisch orientierte) Benutzerschicht angesprochen werden soll und der Erfolg des Systems in großem Maße von der Akzeptanz dieser Benutzerschicht abhängt.

2.1.8 Akzeptanz

Bei der Akzeptanz ist zwischen der o.g. Akzeptanz durch Endbenutzer und der Akzeptanz des Zahlungssystems als Währungsersatz zwischen Kreditinstituten zu unterscheiden. In [FuWr97] wird Akzeptanzfähigkeit als die „Eigenschaft eines Zahlungssystems, überall angenommen zu werden“ ([FuWr97; S.23]) definiert.

Gerade im Bereich der Internetzahlungen spielt dieser Faktor eine wesentliche Rolle, da nationale Grenzen und Währungssysteme „verwischen“ und möglicherweise in naher Zukunft eine eher untergeordnete Rolle spielen (Beispiel: heutige Kreditkartentransaktionen). Die gegenseitige Akzeptanz (elektronischer) Transaktionen kann z.B. durch sogenannte „Clearing-Center“ erfolgen, welche als anerkannte Vermittler zwischen beiden Banken operieren.

2.2 Basiskonzepte für Zahlungssysteme in Internet

Die meisten Autoren unterscheiden zwischen zwei grundsätzlichen Kategorien von Internet-Zahlungssystemen [Reif96] [RiKo97]: **Systeme auf Kreditkartenbasis** und **elektronisches Bargeld** oder „bargeldähnliche“ Systeme. Bei diesen Systemen wird versucht, ein elektronischen Abbild konventioneller und herkömmlicher Verfahren zu entwickeln (bei Bargeldsystemen mit eher mäßigem Erfolg). Kreditkartensysteme benutzen beispielsweise die bereits seit Jahren erprobte und bewährte Infrastruktur zu Kreditkartenabrechnung - lediglich die Übertragung der Transaktionsdaten wurde an das Internet angepaßt.

Als weitere Kategorie kann man noch **Systeme auf Basis von Kundenkonten** hinzunehmen, welche zumindest zur Zeit noch eine relativ große Verbreitung finden (z.B. T-Online, First Virtual, AOL) [SFE97; Kap. 4]. Bei diesen Systeme führt z.B. ein Online-Dienst-Anbieter ein Benutzerkonto, dessen Beträge mit der normalen Online-Rechnung beglichen werden. Dieses Verfahren ist allerdings auf Transaktionen zwischen Teilnehmern des entsprechenden Systems beschränkt.

Auf die auch noch verbreiteten Zahlungsverfahren „**Rechnung**“ und „**Lastschrift**“ soll in diesem Seminarbeitrag nicht eingegangen werden, da diese kein eigenständiges Zahlungssystem darstellen, sondern das Internet lediglich als reines Bestellmedium verwenden (vergleichbar mit Bestellungen via Fax und Telefon).

Im folgenden sollen nun die unterschiedlichen Eigenschaften der verschiedenen Kategorien erläutert werden, um dann im weiteren Verlauf einen Überblick über aktuelle Systeme zu geben und einige dieser Systeme detaillierter zu diskutieren.

2.2.1 Kontosysteme

Zahlungssysteme auf der Basis von Kundenkonten befinden seit vielen Jahren sich in geschlossenen Systemen im praktischen Einsatz (z.B. **T-Online**) und finden seit einiger Zeit auch im offenen und „unsicherem“ Internet Anwendung (z.B. **First Virtual**). Der Betreiber eines solchen Systems führt demnach ein virtuelles Konto für jeden Netzteilnehmer, dessen Salden in regelmäßigen Abständen über konventionelle Zahlungssysteme (im Falle T-Online die Telefonrechnung) beglichen werden müssen. Sämtliche Teilnehmer eines solchen Systems können sowohl als Kunde, als auch als Händler auftreten (letzteres ggf. gegen zusätzliche Gebühren).

Durch die effiziente Verwaltung der Konten durch nur einen Systembetreiber und der Sammlung von Transaktionen zur periodischer Abrechnung, eignen sich solche Zahlungssysteme sehr gut für die Zahlung von Kleinstbeträgen, wie sie im Internet z.B. bei seitenorientierten Abrechnungen auftreten können. Durch diese Konzeption fallen nur relativ geringe Kosten für Kommunikation, Prüfung und Buchung der elektronischen Transaktionen an.

Als beteiligte Interessengruppen treten bei solchen Systeme Kunden, Händler, Banken und Systembetreiber auf, wobei dem Systembetreiber eine zentrale Funktion zukommt. Händler und Kunde müssen vor der Transaktion in einem festen Vertragsverhältnis zum Systembetreiber stehen, demnach ist das Zahlungssystem auf einen **geschlossenen** und abgegrenzten **Benutzerkreis** eingeschränkt. „Spontane“ Kaufvorgänge systemfremder Kunden sind nicht zu realisieren, da Identitäts- und Bonitätsprüfungen nicht kurzfristig durchzuführen sind.

2.2.2 Kreditkartensysteme

Die wohl ältesten Zahlungssysteme im Internet sind die Zahlungssysteme auf Kreditkartenbasis. Kamen die ersten Systeme noch gänzlich ohne Schutzmechanismen aus, so werden an heutige Systeme hohe Anforderungen bezüglich der Sicherheit gestellt – kein Kunde schickt sensible Transaktionsdaten wie z.B. seine Kreditkartennummer guten Gewissens über eine ungesicherte Internet-Verbindung.

Um von vornherein Mißverständnissen vorzubeugen: Kreditkartensysteme im Internet sind keine Zahlungssysteme im engeren Sinne, da hierbei das Internet lediglich eine einleitende Funktion besitzt und diese bestehenden Systeme lediglich um einige zusätzliche Funktionen ergänzt. Im Gegensatz zu den hier nicht behandelten Rechnungen und Lastschriften, bilden diese Systeme allerdings einen wichtigen Stützpfeiler für den nationalen und internationalen Handel im Internet. Großen Anteil daran hat sicherlich die bewährte, währungsunabhängige und weltweit zur Verfügung stehende Infrastruktur der Kreditkartenkartenanbieter. Im wesentlichen erweitern diese Systeme die bestehende Infrastruktur nur um ein weiteres Medium zur Übertragung der Transaktionsinformationen, analog zur schriftlichen oder telefonischen Bestellung von Waren und Dienstleistungen – dem Internet.

Als beteiligte Interessengruppen lassen sich hierbei Kunde, Händler, Banken und Kreditkarteninstitutionen ausmachen. Ein Systembetreiber wie bei den Kontosystemen ist nicht zwingend erforderlich (auch wenn aktuell realisierte Systeme ähnlich „vertrauenswürdige“ Institutionen voraussetzen – die z.B. von SET geforderten Zertifizierungsstellen und *payment gateways* sind gute Beispiele dafür).

Ein Schwerpunkt bei der Entwicklung von Kreditkartensystemen im Internet ist sicherlich die Gewährleistung der sicheren Übertragung der sensiblen Transaktionsdaten, wie z.B. Kreditkarteninformationen, Transaktionswerte und Identifikationsdaten der Beteiligten.

Kreditkartensysteme bieten systembedingt nicht die Möglichkeit der anonymen Bezahlung, was sie für bestimmte Anwendungen unattraktiv macht. Wie bereits erwähnt, eignen sie sich auch nicht für die im Internet anfallenden Mikrozahlungen, da die Kosten solcher Transaktionen schlichtweg zu hoch sind. Ein weiterer Nachteil ist die notwendige Vertragsbindung zwischen einem potentiellen Händler und dem Kreditkarteninstitut, was die Nutzung für Transaktionen zwischen Privatpersonen ausschließt.

Kreditkartensysteme können also nicht als universelles Zahlungssystem betrachtet werden, doch trotz aller Nachteile werden sie durch ihren vielfältigen Anwendungsbereich und der vorhandenen Infrastruktur sicherlich einen großen Marktanteil im elektronischen Handel erreichen.

Viele Konzepte zur Nutzung von Kreditkarten im Internet wurden bereits realisiert. Angefangen von der Entwicklung **sicherer Übertragungsprotokolle** wie z.B. SSL und SHTTP, über spezielle **Finanztransaktionsprotokolle** und Implementierungen wie SET, bis hin zum bereits erprobten und integriertem System **CyberCash**.

2.2.3 Bargeldsysteme

Auch bei den nun beschriebenen Bargeldsystemen treten als notwendige Institutionen nur Kunden, Händler und Banken auf, allerdings ergänzen viele Umsetzungen dieser Systeme diese – analog zu den Kreditkartensystemen – um weitere Institutionen.

Bei der Entwicklung bargeldähnlicher Zahlungssysteme sind einige systembedingte Probleme zu lösen: Zum einen ist durch entsprechende Maßnahmen ein Kopieren der elektronischen Münzen bzw. deren Mehrfachverwendung zu verhindern, anderseits sollten die dabei angewendeten Verfahren möglichst geringen Einfluß auf die Skalierbarkeit und Offline-Fähigkeit haben. Ebenso wäre eine Teilbarkeit der elektronischen Münzen wünschenswert, um so auch eine weniger gezielte Verwendung der Münzen zu ermöglichen.

In [FuWr97] wird eine Klassifizierung in **Einweg-Token-Systeme** und **Mehrweg-Token-Systeme** vorgenommen. Einweg-Token-Systeme sind bargeldähnliche Zahlungssysteme, deren charakteristische Eigenschaft die nur einmalige Nutzung eines Tokens bzw. einer elektronischen Münze ist. Bei Mehrweg-Token-Systemen kann ein Token beliebig häufig zwischen Benutzern des Systems weitergegeben werden, ohne daß eine übergeordnete Instanz dessen Gültigkeit und Echtheit ständig bestätigen muß. Diese, bislang echtem Bargeld vorbehaltene Eigenschaft, läßt sich bis heute allerdings technisch nicht vollständig realisieren. Ein erwähnenswerter Ansatz ist z.B. die Koppelung eines Tokens mit einem Transaktionsprotokoll. Jede Transaktion wird protokolliert und dem neuen Besitzer automatisch mit übergeben. Bei Mehrfachverwendung oder sonstigem Mißbrauch kann die Bank dieses Transaktionsprotokoll entschlüsseln und so den Betrüger identifizieren. Der Nachteil dieses Konzeptes ist offensichtlich: Bei jeder Transaktion nimmt die Größe des Tokens (bzw. des Transaktionsprotokolls) zu, so daß eine Übertragung des Tokens unverhältnismäßig hohe Kosten verursachen würde.

Diese (und viele weitere) Probleme haben dazu geführt, daß keines der zur Zeit verfügbaren Systeme in der Lage ist, alle Eigenschaften „echten“ Bargelds nachzuahmen bzw. zu implementieren. Man kann all diese Systeme deshalb lediglich als „bargeldähnliche“ Systeme bezeichnen.

Bargeldähnliche Systeme haben gegenüber Kreditkartensystemen den Vorteil, Mikrozahlungen und anonyme Zahlungen wirtschaftlich abwickeln zu können, wodurch sie für viele Anwendungsfälle im Internet prädestiniert sind.

Kleinhändler und private Verkäufer werden auf jeden Fall von diesen Systemen profitieren, da ihnen nun das Internet als „neuer Markt“ zur Verfügung steht – feste Verträge mit z.B. Kreditkarteninstituten sind nicht notwendig.

3 Beispiele für Zahlungssysteme im Internet

Im nun folgenden Kapitel soll eine Übersicht über aktuelle Ansätze, Prototypen und bereits kommerziell genutzte elektronische Zahlungssysteme gegeben werden. Den o.g. Kategorien werden einige Beispiele zugeordnet, wobei jeweils einige ausgewählte Beispiele detaillierter vorgestellt werden.

3.1 Kontosysteme

Die klassischen Systeme auf Basis von Kundenkonten sind wohl die **Online-Dienste** der großen Online-Anbieter wie z.B. AOL, CompuServe und T-Online. Im Internet findet man bereits Umsetzungen dieser Konzepte in Form von **geschlossenen Systemen** (z.B. „DOWNTOWN-Anywhere“, [URL-1]). Diese, zwar im Internet präsentierten Dienste, stellen ihr volles Leistungsspektrum lediglich einem geschlossenen Benutzerkreis zur Verfügung. Eine eher ungewöhnlich Umsetzung von Kontosystemen findet man bei dem weit Verbreiteten Zahlungssystem **First Virtual**, welches gänzlich ohne Verschlüsselungsverfahren auskommt.

3.1.1 Online-Dienste

Die großen Online-Dienste bieten neben dem obligatorischen Internet-Zugang auch eigene, geschlossene Bereiche an, in denen Kunden und Händlern Gelegenheit gegeben wird, ohne großen Aufwand Geschäfte zu tätigen. Der Erwerb der dort angebotenen Waren und Dienstleistungen steht allerdings nur Mitgliedern des entsprechenden Online-Dienstanbieters zur Verfügung.

Ein großer Vorteil dieser Systeme ist die Möglichkeit für Kleinhändler und Privatpersonen als Händler aufzutreten und so die Vorteile des elektronischen Handels zu nutzen.

Die Sicherheit in diesen geschlossenen Systemen wird meist durch proprietäre Netzwerke gewährleistet (z.B. CEPT), wodurch potentielle Angriffe von betrügerischen und anonymen Internet-Benutzern unmöglich gemacht werden. Der eigentliche Abrechnungsvorgang wird durch das bereits vorhandene Abrechnungssystem bewerkstelligt, wodurch die Transaktionskosten solcher Systeme auf extrem niedrigem Niveau gehalten werden – es entsteht kein Mehraufwand für den Betreiber.

Diese Abrechnungsmodalitäten verlangen allerdings eine vollständige Protokollierung aller Transaktionen durch den Betreiber. Anonyme Transaktionen sind also nicht möglich – einer Erstellung und Vermarktung von Kundenprofilen steht nichts mehr im Wege (gesetzliche Regelungen ausgenommen).

Äußerst kritisch wird dieses Konzept bei einer zentralistischen Organisationsstruktur, wie sie z.B. von CompuServe betrieben wird. Ein solches System macht eine mögliche Auswertung des Datenbestandes in Verbindung mit z.B. Data-Mining-Systemen noch effizienter.
Ein erheblicher Nachteil dieser Systeme ist die nur schwierig zu realisierende Ausdehnung auf das Internet, da hierbei viele Sicherheitsaspekte zu berücksichtigen und zu bewältigen sind. Es werden zur Zeit z.B. von T-Online Anstrengungen unternommen, zumindest Teile der proprieträren Bereiche im Internet zu präsentieren (CEPT-„Plug-In").

3.1.2 Geschlossene Systeme

Eine direkte Umsetzung der o.g. Systeme im Internet findet man bei den sog. „geschlossenen Marktplätzen" vor. Der hauptsächliche Unterschied besteht in den verwendeten Zugangsmechanismen zu diesen virtuellen Marktplätzen. Häufig kommen hier Paßwort oder PIN-Systeme zum Einsatz.
In [SFE97] wird das Konzept des Systembetreibers „DOWNTOWN-Anywhere" vorgestellt, welcher auf diesem Wege seinen Mitgliedern die Möglichkeit bietet, konventionelle Kreditkartensysteme und ein eigenes Abrechnungssystem für Mikrozahlungen zu nutzen. Das eigene Abrechnungssystem basiert auf einer PIN-basierten Bezahlung durch Kundenkonten mit monatlicher Abrechnung.
Benutzern dieses geschlossenen Marktplatzes stehen alle dort angebotenen Waren und Dienstleitungen zur Verfügung. Durch die Präsentation im Internet, haben auch Nicht-Mitglieder Zugang zu dem System, allerdings stehen Ihnen u.a. die Abrechnungssysteme (und damit die Möglichkeit zum Erwerb von Waren oder Dienstleistungen) nicht zur Verfügung.

3.1.3 First Virtual

Ein eher ungewöhnliches Konzept verfolgt der Systembetreiber von „First Virtual". Das System kommt gänzlich ohne Verschlüsselungsverfahren aus. Kunden registrieren sich per Telefon und hinterlegen auf diesem „sicheren" Kanal ihre Identifikations- und Kreditkarteninformationen. Der eigentliche Kaufvorgang wird anschließend ausschließlich über Email abgewickelt. Der Systembetreiber ermöglicht gegen eine Gebühr jeder Person Händleraktivitäten zu führen.

Ablauf einer typischen Transaktion:

1. Der Kunde gibt via Internet seine Bestellung beim Händler auf.
2. Der Kunde authentifiziert sich dabei durch seine persönliche PIN („VirtualPIN").
3. Der Händler reicht die erhaltenen Transaktionsinformationen an „First Virtual" weiter.
4. Der Systembetreiber kontaktiert den Kunden per Email, um sich die Transaktion betätigen zu lassen.
5. Der Systembetreiber wartet auf die Reaktion des Kunden:
   → Falls der Kunde ***nicht antwortet***, so verweigert er die Transaktion.

- → Zeigt der Kunde einen ***Betrug*** an, so löst er das Konto des Kunden auf und sperrt die zugehörige PIN. Anschließend bietet er dem Kunden die Option an, ein neues Konto zu eröffnen.
- → ***Verweigert*** der Kunde die Transaktion, so wird dieses protokolliert – bei zu vielen Verweigerungen wird das Konto aufgelöst.
- → ***Bestätigt*** der Kunde die Transaktion, so erfolgt die Buchung auf das Kundenkonto und die Benachrichtigung des Händlers, damit dieser seine Leistung erbringen kann.

6. Der Saldo des Kundenkontos wird monatlich über die Kreditkarte des Kunden beglichen.

Als zusätzliche Sicherheitsmaßnahmen dienen ein monatlicher Wechsel der PIN und eine „Geld-zurück-Garantie“ zu Lasten des Händlers.

Ein maßgeblicher Schwachpunkt des Systems sind die Email-basierten Transaktionen, die potentiellen Betrügern das Abfangen und Täuschen von Daten relativ einfach machen. Ein Betrug in gewissen Dimensionen wird so bewußt einkalkuliert. Nachteilig sind auch die relativ hohen Transaktionskosten anzumerken, die durch den hohen Kommunikationsbedarf und zeitlichen Aufwand einer Transaktion ausgelöst werden.

Die Systembetreiber von „First Virtual“ protokollieren sämtliche Transaktionen ihrer Kunden, um so ein Benutzerprofil anzufertigen und dieses kommerziell zu vermarkten.

3.2 Kreditkartensysteme

Die ersten Implementierungen zur sicheren Kreditkartenanwendung im Internet basierten auf der reinen Bereitstellung eines sicheren Kanals, welcher zur Übertragung der sensiblen Transaktionsdaten genutzt wurde bzw. wird. Die **sicheren Übertragungsprotokolle** SSL und SHTTP gehören z.B. in diese Kategorie. Integrierte Systeme, wie z.B. das verbreitete **CyberCash**-System, bieten jedoch deutlich mehr als sichere Übertragungskanäle – sie realisieren eine marktfähige und kommerziell einsatzfähige Oberfläche zur elektronischen Nutzung von Kreditkarten.

Als Finanztransaktionsprotokoll könnte man das relativ neue **SET** (Secure Electronic Transaction) bezeichnen, welches aus einer Kooperation der an dessen Vorgängern **SEPP** und **STT** beteiligten Firmen und Institutionen entstanden ist (MasterCard, VisaCard, Microsoft, Netscape, IBM, CyberCash, VeriSign, American Express u.a.m.)

Dieses System bietet eine sichere Übertragung von speziell auf den elektronischen Dienstleistungs- und Warenverkehr abgestimmten Dokumenten an. Es ist als offener Industriestandard konzipiert, aber kein vollständiges Zahlungssystem wie z.B. CyberCash. Es wird eine Integration in solche integrierten Systeme angestrebt.

Es existieren noch weitere kreditkartenorientierte Systeme auf dem Markt, dessen Konzepte aus Gründen des Umfangs dieser Arbeit allerdings keine Beachtung finden konnten. Dem zuvor erwähnten SET werden zur Zeit aufgrund der mächtigen Kooperationspartner auch die größten Marktchancen eingeräumt.

3.2.1 Sichere Übertragungsprotokolle

Eine mögliche Anwendung sicherer Übertragungsprotokolle ist die Übertragung sensibler Transaktionsdaten in kommerziellen Internet-Anwendungen. Die wohl bekanntesten Implementierungen sind SSL und SHTTP.

SSL arbeitet als Zwischenschicht zwischen Transport und Anwendungsschicht. Es setzt auf die vorhandene Socketschnittstelle auf und erweitert diese um sicherheitsspezifische Funktionen. Durch diese Architektur profitieren auch andere Anwendungen wie z.B. TELNET oder FTP von den bereitgestellten Sicherheitsfunktionen.

Das eigentliche Verschlüsselung der Daten erfolgt durch ein hybrides Verschlüsselungsverfahren. Die Kodierung und Dekodierung der Nachricht erfolgt durch einen symmetrischen DES-Schlüssel, wobei die Übertragung des DES-Schlüssels durch das asymmetrische RSA-Verfahren bewerkstelligt wird (s.o.). Die authentische Übertragung und Authentifizierung wird durch Zertifizierung der verwendeten Schlüssel mit Hilfe übergeordneter Instanzen sichergestellt (CA, Trust-Center). Aufgrund von Exportbeschränkungen des RSA-Verfahrens ist die Schlüssellänge allerdings zur Zeit in Europa auf bis zu 40 Bit (Netscape) beschränkt, wodurch die Sicherheitsfunktionen als sehr schwach einzustufen sind. Es bleibt zu hoffen, daß (noch) sichere 128-Bit Schlüssel schnellstens zur breiten Anwendung kommen.

Im Gegensatz zu SSL ist **S-HTTP** kein Zwischenprotokoll, sondern eine reine Erweiterung des HTTP-Protokolls. Es stellt Mechanismen und einheitliche Formate zur Verschlüsselung, Authentifizierung, Signierung und Integritätskontrolle zur Verfügung. Da S-HTTP keinen einheitlichen kryptographischen Standard vorschreibt, lassen sich keine generellen Aussagen über deren Wirksamkeit treffen.

Das System schreibt vor, daß sich die beteiligten Systeme auf ein gemeinsames Verschlüsselungsverfahren einigen. Als Basisverfahren bereits implementiert sind z.B. RSA, PEM (Private Enhanced Mail) ,PGP u.a.m..

3.2.2 CyberCash

Das CyberCash-System ist eines der wenigen Systeme, die bereits mit einer vollständig implementierten Anwendersoftware aufwarten können. Es stellt im Gegensatz zu den Übertragungsprotokollen und SET ein vollständiges Zahlungssystem dar.

Die Systembetreiber des CyberCash-Systems entwickeln seit 1994 Zahlungssysteme für das Internet und sind ebenfalls an der Entwicklung des nachfolgend vorgestellten SET beteiligt. Zur Zeit läuft ein Pilotprojekt in Kooperation mit der Dresdener Bank und der Landesbank Sachsen, das helfen soll, die Marktchancen und Ausgereiftheit des Systems zu beurteilen.

Basis des Systems sind spezielle Software-Produkte:

- die Kundensoftware, die eine elektronische Geldbörse implementiert - das sogenannte ***Wallet***. Dieses enthält verschiedene (virtuelle) Zahlungsobjekte wie z.B. Kreditkarten, Schecks oder Bargeld
- die Händlersoftware, die eine elektronische Registrierkasse implementiert – das sogenannte ***Register*** oder „*Secure Merchant Payment System*“. Der als *Gateway* operierende CyberCash-Server stellt eine Verbindung zu den internationalen Finanz-Netzwerken her.

Ablauf einer typischen Transaktion:

1. Der Kunde gibt durch drücken der Pay-Taste auf der WWW-Seite des Händlers seine Bestellung auf.

2. Die Händler-Software sendet alle Transaktionsdaten zur automatisch startenden Kunden-Software.

3. Die Kunden-Software überträgt z.B. die bei der Installation des *Wallets* eingegebenen Kreditkarteninformationen mit Hilfe eines 768 Bit-RSA-Schlüssels an den Händler (demnächst 1024-Bit).

4. Der Händler fügt seine Identitätsinformationen hinzu, sendet die Daten an den CyberCash-Server und wartet auf die Autorisierung der Transaktion.

5. Der CyberCash-Server entschlüsselt die für den Händler nicht lesbaren Kreditkarteninformationen und autorisiert ggf. die Transaktion mit Hilfe des Kreditkartenunternehmens bzw. der zuständigen Bank. Zur Zeit werden die CyberCash-Server in Eigenregie betrieben, in absehbarer Zeit soll allerdings ein Lizenzbetrieb z.B. durch Banken realisiert werden.

6. Nach erfolgreicher Autorisierung durch den CyberCash-Server sendet die Händler-Software eine Quittung zum Kunden und der Händler kann seine Leistung erbringen.

Als Ergänzung zum CyberCash-System wird noch die entsprechende Mikropayment-Ergänzung CyberCoin betrieben, die hier jedoch nicht näher erläutert wird. CyberCoin gehört in die Kategorie der bargeldähnlichen Systeme, obwohl es eigentlich kein Bargeld-System, sondern eine Art Buchgeld-System implementiert, bei dem die beteiligten Banken ein Konto über die virtuellen Beträge des *Wallets* führen und Transaktionen lediglich Umbuchungen auslösen.

3.2.3 SET

Das im folgenden SET-Verfahren soll demnächst in das gerade beschriebene CyberCash-System integriert werden und als Basis für Kreditkartentransaktionen genutzt werden. SET umfaßt die sichere Abwicklung von Kaufvorgängen, Bestellungen, Quittungen, bis hin zur Spezifikation von Prozeduren zur Zertifikatserteilung und der Struktur der entsprechenden Zertifizierungsstellen. SET ist also eher eine Art Spezifikation oder Richtlinie als ein Zahlungssystem. Es beinhaltet eine Spezifikation der benötigten Dokumenttypen und beschreibt deren Verwendung. Ebenso existieren bereits erste Implementierungen des Verfahrens, welche z.B. den Dokumenten entsprechende Datentypen umfassen.

Weitere Informationen zu aktuellen Implementierungen und deren API's findet man bei [URL-2] und [URL-3].

Im folgenden soll eine typische Transaktion erläutert werden, die die spezifizierten Vorgänge Bestellung (**purchase request**), Autorisierung *(**authorisation request***) und Abrechnung **(payment capture**) verwendet.

Innerhalb dieser Vorgänge werden u.a. die folgenden (spezifizierten) Dokumente verwendet:

- *purchase request:*
  - *initiate request*
  - *initiate response*
  - *purchase request*
  - *purchase response*
- *authorisation request:*
  - *merchant authorisation request*
  - *payment gateway authorisation response*
- *payment capture:*
  - *capture request*
  - *capture response*

In der folgenden beispielhaften Beschreibung einer Transaktion werden die SET-Dokumenttypen in eckigen Klammer („[]") dargestellt.

Bestellung (purchase request):

1. Der Kunde möchte eine Transaktion tätigen und sendet einen [initiate request] and den Händler.

2. Der Händler generiert eine Antwort [payment initiate response] und unterschreibt diese bzw. dessen Nachrichten-Prüfsumme mit seinem privaten Signatur-Zertifikat. Händler und Kunde besitzen bei SET-Transaktionen jeweils zwei Schlüsselpaare – eins zum Signieren und eins zum Verschlüsseln.

3. Der Händler sendet die Antwort [payment initiate response], sein öffentliches Signatur-Zertifikat und das öffentliche Verschlüsselungs-Zertifikat seiner Händler-Bank (*payment gateway*). Diese drei Objekte ergeben die [initiate response].

4. Der Kunde überprüft nun beide Zertifikate durch die Signatur einer Zertifizierungsstelle.

5. Der Kunde verifiziert das Händler-Signatur-Zertifikat durch Vergleich mit der neu zu berechnenden Prüfsumme der dekodierten Nachricht und der, mit dem öffentlichen Signatur-Zertifikat des Händlers dekodierten, digitalen Unterschrift. Dadurch wird gewährleistet, daß der Verfasser der Nachricht auch der tatsächliche Inhaber des Zertifikats ist (ein Betrüger könnte sonst z.B. ein fremdes Zertifikat mitschicken).

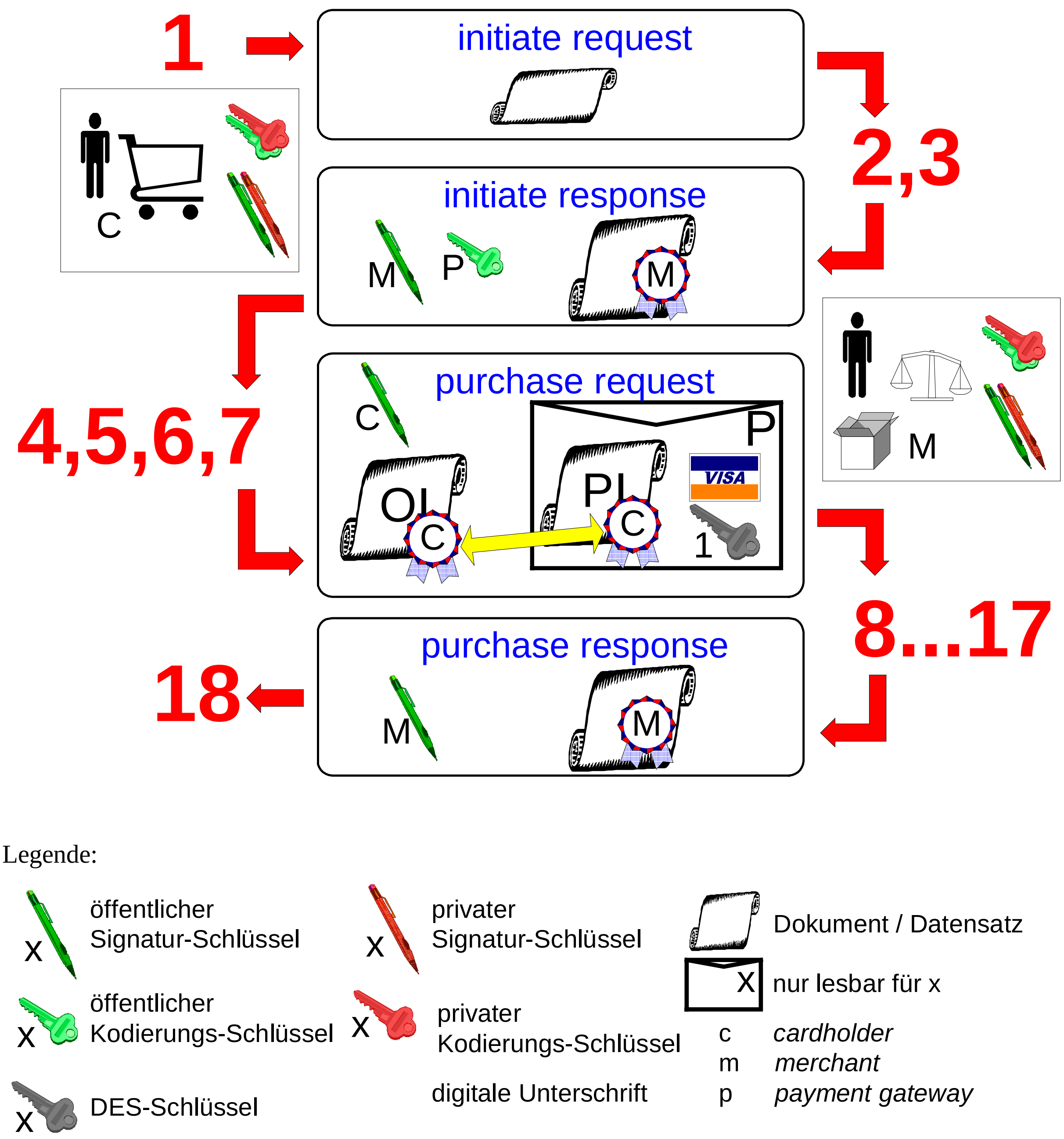

Abb. 2: Bestellung (purchase request)

6. Der Kunde fertigt nun zwei 2 Dokumente an: die Bestellung [order information] und die Zahlungsanweisung [payment information]. Beide werden nun mit einer dualen Signatur signiert, wodurch die Dokumente gegenseitig zugeordnet werden. Zur späteren Verifikation enthält die Bestellung die Prüfsumme der Zahlungsanweisung. Die [pament information] wird anschließend mit einem zufällig generierten DES-Schlüssel kodiert. Der DES-Schlüssel und die sensiblen Kreditkarteninformationen werden mit dem öffentlichen Schlüssel des *payment gateways* - für den Händler unlesbar - verschlüsselt (RSA, 1024 Bit).

7. Der Kunde sendet nun die Nachricht [purchase request], bestehend aus Bestellung, Zahlungsanweisung, DES-Schlüssel, Kreditkarteninfomationen seinem Signatur-Zertifikat. Dieses Signatur-Zertifikat kann optional durch eine Zertifizierungsstelle zertifiziert sein, welches jedoch nicht zwingend notwendig ist und lediglich der schnellen Markteinführung des Systems entgegenkommen soll. Die Entscheidung, ein nicht zertifiziertes Zertifikat anzunehmen, obliegt den entsprechenden Kreditkarteninstituten.

8. Der Händler überprüft nun ggf. das Signatur-Zertifikat des Kunden und die duale Signatur der Bestellung (Vergleich von berechneter Prüfsumme der Bestellung und übertragener Prüfsumme der Zahlungsanweisung mit der dekodierten Unterschrift)

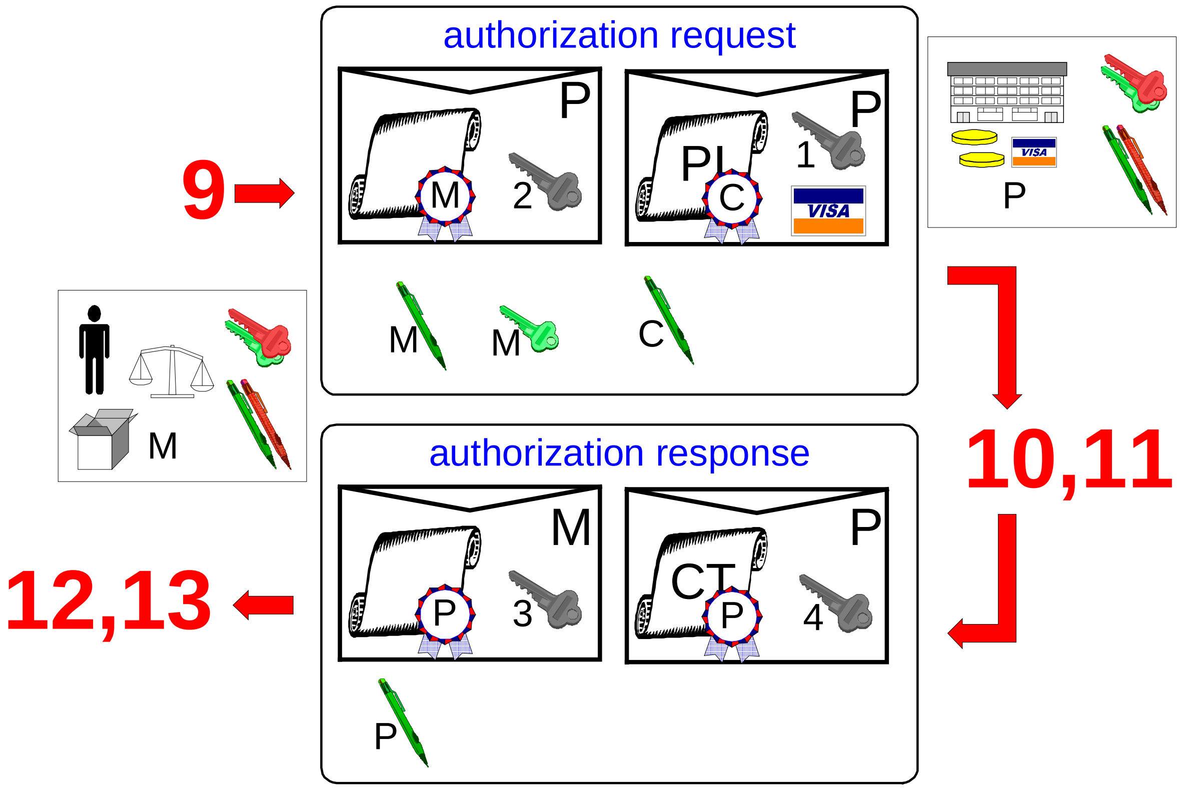

Abb. 3: Autorisierung (authorisation request)

Autorisierung (authorisation request):

9. Der Händler generiert eine Autorisierungs-Anfrage [merchant authorization request], bestehend aus der von ihm unterschriebenen und DES-verschlüsselten Anfrage [authorization request], dem dazu benutzten DES-Schlüssel (verschlüsselt mit dem öffentlichen Verschlüsselungszertifikat des *payment gateways*), der Zahlungsanweisung des Kunden inkl. dessen DES-Schlüssel, dem Signaturzertifikat des Kunden und seinen beiden eigenen Zertifikaten.

10. Das *payment gateway* prüft nun analog zu 4. und 5. die Zertifikate. Es besitzt nun alle notwendigen Informationen, um die Autorisierung durchzuführen. Die Zahlungsanweisung wird entschlüsselt und mit den Daten den Händler verglichen. Die Liquidität des Kunden wird über das geschlossene Banken-Netzwerk (z.B. SWIFT) durchgeführt.

11. Es wird nun eine Antwort ([payment gateway authorization response]) generiert, bestehend aus unterschriebener und DES-verschlüsselter Antwort ([auth. response]), DES-Schlüssel, dem öffentlichen Signatur-Zertifikat des *payment gateway* und einem sogenannten [capture token]. Dieses [capture token] enthält Informationen, die ggf. zur späteren Bearbeitung der Anfrage notwendig sind und ist analog zu z.B. 6. und 9. verschlüsselt werden (allerdings nur für das *payment gateway* selbst lesbar, da das eigene Verschlüsselungszertifikat verwendet wird).

12. Der Händler führt nun analog zu 4. und 5. einige Prüfungen durch und speichert das [capture token] zur späteren Verwendung.

13. Der Händler erbringt nun seine angebotene Leistung.

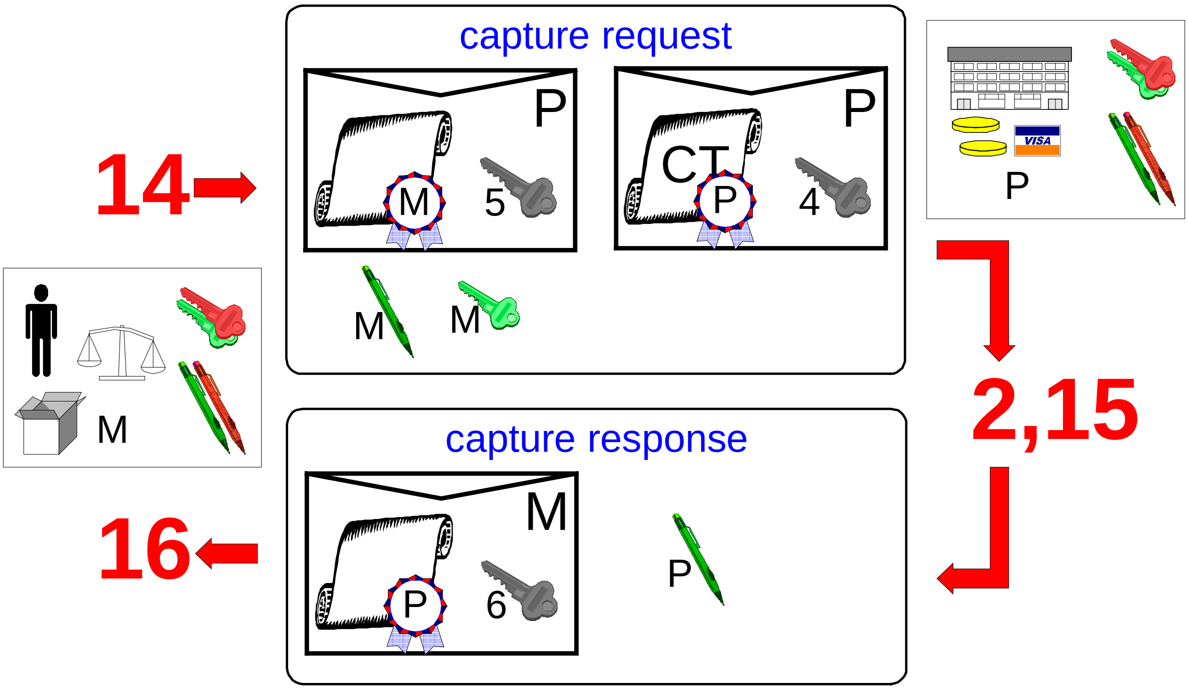

Abb. 4: Abrechnung (payment capture)

Abrechnung (payment capture):

14. Der Händler erzeugt eine Abrechnungs-Anfrage [merchant capture request], die aus dem [capture token] samt DES-Schlüssel, der Anfrage [capture request] samt DES-Schlüssel und den öffentlichen Händler-Zertifikaten besteht.

15. Das *payment gateway* überprüft nun die Gültigkeit und Konsistenz der Anfrage und führt die Zahlungsanweisung über das Banken-Netzwerk aus.

16. Der Händler enthält eine entsprechende Bestätigung oder Fehlermeldung.

17. Der Händler schickt dem Kunden eine Quittung und beendet die Transaktion.

Neben den gerade vorgestellten Vorgängen deckt die SET-Spezifikation noch viele weitere kommerzielle Vorgänge ab, die in dieser Ausarbeitung allerdings nicht vorgestellt werden.

3.3 Bargeldsysteme

Analog zu den Kreditkartensystemen, macht es die Vielzahl der sich zur Zeit in kommerzieller Nutzung oder in Feldversuchen befindlichen bargeldähnlichen Zahlungssysteme unmöglich, im Rahmen dieser Arbeit eine umfassend Bewertung und Analyse dieser Systeme zu geben. Als Beispiele für solche Systeme sind z.B. NetCash, CyberCoin, Millicent, **ECash**, GlobeID, **UEC** („universal electronic cash") und Mondex zu nennen. Aufgrund der aktuellen Pilotprojekte und der teilweise herausragenden Eigenschaften werden jedoch nur das ECash-System und das theoretische UEC beschrieben. Erwähnenswert erscheint auch das Millicent-System, da dieses eines der wenigen bereits realisierten Systeme ist, die Pikozahlungen durchführen können [Lang98].

3.3.1 ECash

Das zentralistisch organisierte Zahlungssystem ECash wurde entwickelt von der Firma DigiCash unter Leitung von Dr. David Chaum. Die ursprünglich Mauterhebungs- und Zahlungssysteme entwickelnde Firma ist auch an EU-Projekten wie CAFE oder dem Nachfolgeprojekt SEMPER beteiligt. Aufgrund der Fokussierung auf „intelligente" Kreditkarten und SmartCards, gehören diese Projekte jedoch nicht unmittelbar in das hier diskutierte Themengebiet. Obwohl die Firma auch Technologien für offline-fähige Systeme auf Basis des später noch erläuterten *secret-sharing*-Verfahrens entwickelt, ist das ECash-System ein online-basiertes System.

In einem Pilotprojekt der Deutschen Bank wird seit Oktober 97 ein Versuch mit 1500 Kunden durchgeführt, der die Marktreife des Systems ergründen soll.

Als Beteiligte treten in diesem System neben Kunden, Händlern und Banken auch sogenannte ***Double-Spending-Server*** auf, die die Mehrfachverwendung von elektronischen Münzen durch das Führen entsprechender Listen verhindern sollen.

Ablauf einer typischen Transaktion:

1. Der Kunde hebt mit der Kundensoftware online Geld von seinem ECash-Bankkonto ab. Bei diesem Vorgang wird der entsprechende Betrag von dem konventionellen Konto abgebucht und in Form von elektronische Münzen zum Kunden gesendet. Um Anonymität bei Zahlungen zu gewährleisten, verwendet die Bank zur Erzeugung der Münzen das sogenannte „blinding"-Verfahren (s.u.). Die eigentliche Übertragung der elektronischen Münzen erfolgt auf konventionelle Weise mit Hilfe eines hybriden Verschlüsselungsverfahrens. Zur Authentifizierung des Kunden vergibt die Deutschen Bank bei der Anmeldung zu ihrem Pilotprojekt ein Kontopaßwort. Der Kunde hat nun den entsprechenden Betrag in elektronische Form auf seiner Festplatte gespeichert.

2. Der Kunde löst den Kauf der Waren oder Dienstleistungen durch eine entsprechende Schaltfläche auf der WWW-Seite des Händlers aus – es erscheint eine entsprechende Zahlungsaufforderung. Nach der anschließenden Übertragung der Münzen an den Händler, wird diesem eine sofortige Validierung der Münzen auf deren Gültigkeit empfohlen, wozu er den *Double-Spending-Server* der Bank nutzt. Nach der erfolgten Bestätigung durch die Bank kann der seine Leistung erbringen. Weiterhin hat er die Möglichkeit, die erhaltenen Münzen umzutauschen, um so Mißbrauch vorzubeugen (die Seriennummer wird dadurch sofort ungültig).
3. Der Händler überträgt in regelmäßigen Abständen den Wert der angesammelten elektronischen Münzen auf sein konventionelles Bankkonto.

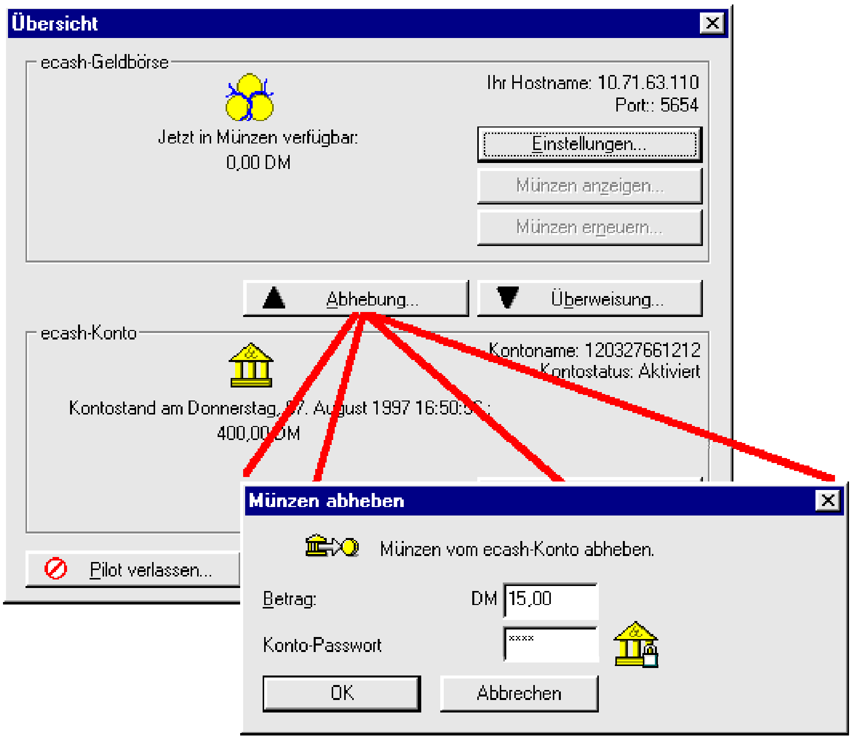

Abb. 5: ECash-Pilot der Deutschen Bank

Das bereits oben erwähnte ***blinding*-Verfahren** stellt die Anonymität der Zahlungen sicher und verhindert so die Erstellung von Kundenprofilem. Das von Dr. David Chaum entwickelte Verfahren wird durch einen modifizierten RSA-Algorithmus realisiert [].

Die Bank generiert dazu einen privaten Schlüssel **d**, einen öffentlichen Schlüssel **e** und einen öffentlichen Modulus **n**.

1. Der Kunde generiert selbst die von ihm gewünschten Münzen mit einer zufälligen Seriennummer [Kunde generiert Nachricht **m**]
2. Der Kunde verfremdet diese Seriennummer durch einen von ihm ebenfalls zufällig ausgewählten *blinding*-Faktor.

[Kunde generiert *blinding*-Faktor **k** mit **1<=k<=n** und verdeckt die Nachricht m durch **m‘ = mk^{e} mod n**]

3. Der Kunde reicht die Münzen bei seiner Bank zur Signierung ein. Die Bank signiert nun „blind“ die eingereichten Münzen, ohne die Seriennummer lesen zu können. Die Echtheit der Münzen wird also nicht durch die Seriennummer, sondern durch die Signatur der Bank gewährleistet. Die Seriennummer dient also lediglich als Schutzmechanismus gegen Mehrfachverwendungen.
   [Die Bank signiert m‘ mit ihrem privaten Schlüssel d: **s‘ = m‘^{d} mod n**]
4. Der Kunde entfernt anschließend die Verfremdung durch Kenntnis des *blinding*-Faktors, ohne dabei die Signatur der Bank zu zerstören. Der Kunde ist nun im Besitz anonymisierter, aber durch die Bank signierter elektronischer Münzen.
   [Der Kunde entfernt die Verfremdung mit dem *blinding*-Faktor**: s = s‘ / k mod n**]
5. Die Prüfung der Bank-Signatur erfolgt analog zum normalen RSA-Signaturverfahren.
   [Die Gleichung **s = m^{e} mod n** muß erfüllt sein]

Durch eine zusätzliche **Protokollierung aller Transaktionen** auf Bank- und Kundenseite, kann der Kunden jederzeit alle Transaktionen nachweisen und belegen. Dieses Verfahren schützt vor betrügerischen Absichten potentieller Händler (Unabstreitbarkeit).
Als größten Nachteil des Systems kann man sicherlich die notwendige Online-Verifizierung der Münzen betrachten. Weiterhin sind ECash-Münzen nicht teilbar, d.h. es muß (zumindest bei der von der Deutschen Bank eingesetzten Version 2.3) bei jeder Transaktion der exakte Betrag übermittelt werden. Das folgende System UEC zeigt einen Lösungsansatz für diese Probleme.

3.3.2 UEC

Tatsuaki Okamoto und Kazuo Ohta beschreiben in ihrem Artikel [OkOh91] das (praktisch nicht realisierte) Universal Electronic Cash-System – dem ihrer Meinung nach ersten idealen elektronischen Bezahlungssystem [URL-5].
Es ähnelt dem ECash-System, realisiert allerdings zusätzlich die Eigenschaften der Offline-Fähigkeit und der Teilbarkeit.
Letzteres wird durch das Konzept eines **hierarchischen Strukturbaumes** realisiert, wodurch es möglich wird, elektronischen Münzen bzw. Geldscheine in beliebig viele Untereinheiten aufzuteilen.
Zur Realisierung der Offline-Fähigkeit wird ein Verfahren namens ***secret sharing*** eingesetzt. Bei diesem Verfahren werden jeder elektronischen Münze Information über die Identität des Besitzers beigefügt. Diese Identitätsinformationen sind jedoch erst bei mehrfacher Verwendung (d.h. es werden mind. zwei Münzen der gleichen Seriennummer benötigt) auszuwerten. Ein potentieller Betrüger kann also von den beteiligten Banken identifiziert werden. Durch das zwingende notwendige Vorhandensein von

zwei Münzen, kann die wichtige Eigenschaft der Anonymität auch nur im Betrugsfall aufgehoben werden.
Es bleibt abzuwarten, ob diese Technologien in kommerziellen Systemen realisiert werden, da noch einige praktische und theoretische Probleme zu lösen sind.

4 Zusammenfassung

Welches der o.g. Zahlungssysteme die größten Marktanteile erreichen wird, läßt sich zum aktuellen Zeitpunkt aufgrund der mangelnden praktischen Erfahrungen nicht sagen. Mit ziemlicher Sicherheit wird sich jedoch eine Kombination mehrerer Systeme durchsetzen, die alle unterschiedliche Schwerpunkte aufweisen werden. Diese Integration verschiedener Systeme sollte für den Benutzer unsichtbar unter einer entsprechenden Benutzeroberfläche verborgen bleiben.
Die größten Marktchancen werden zur Zeit dem kreditkartenorientierten SET und dem bargeldähnlichen ECash eingeräumt – nicht zuletzt aufgrund der internationalen Konzerne, die hinter den entsprechenden Konsortien stehen.
Auf jeden Fall existiert noch ein hoher Entwicklungsbedarf an solchen Systemen, denn mit den zur Zeit implementierten und eher unübersichtlichen Benutzeroberflächen wird man Probleme haben, neue Benutzerkreise zu erschließen. Mindestens genauso wichtig erscheint die lückenlose Aufklärung über sicherheitstechnische Risiken der Zahlungssysteme und deren Kostendeckung im Schadensfall.

5 Literatur- und Quellenverzeichnis

5.1 Printmedien

[DrDu97] S. Dresen, T. Dunne. „Penny Lane – Wie das ecash-Projekt der Deutschen Bank praktisch funktioniert". iX. Heise Verlag, Hannover, Nr. 12, 1997

[FuWr97] A. Furche, G. Wrightson. Computer Money – Zahlungssysteme im Internet. dpunkt-Verlag, 1997

[Lang97] B. Lange. „Secure Electronic Transaction: Kreditkarten im Internet". iX. Heise Verlag, Hannover, Nr. 10, 1997

[Lang98] B. Lange. „Mausklick-Preise: Abrechnung von Kleinstbeträgen im Internet". iX. Heise Verlag, Hannover, Nr. 1, 1998

[Luck97a] N. Luckhardt. „Trend-SETer – Sichere Kartenzahlung über unsichere Netze". c't-Report Geld online. Heise Verlag, Hannover, Nr. 3, 1997

[OkOh91] T. Okamoto, K.Ohta. „Electronic Digital Cash – Advances in Cryptology" CRYPTO '91, Spinger Verlag, 1991

[Reif96] H.Reif. „Cyber-Dollars – Elektronisches Geld im Internet". c't. Heise Verlag, Hannover, Nr.5, 1996

[RiKo97] H. Reif, A. Kossel. „Bits statt Bares – Elektronisches Geld im Internet“. c’t-Report Geld online. Heise Verlag, Hannover, Nr. 2, 1997

[SFE97] R. Schuster, J. Färber, M. Eberl. Digital Cash – Zahlungssysteme im Internet. Springer Verlag, 1997

5.2 Elektronische Dokumente

[URL-1] DOWNTOWN-Anywhere (gesichtet 29.12.1997):
`http://www.awa.com`

[URL-2] SET-Informationen (gesichtet 4.11.1997):
`http://www.mastercard.com/set/`

[URL-3] SET-Implemetierung Version 1.0, Sourcen & Dokumentation. Paßwörter für den FTP-Server können durch eine Mail an „Paul_Hollis@mastercard.com“ angefordert werden (gesichtet 4.11.1997):
`ftp://ftp.mastercard.com/pub/`

[URL-4] SET-Informationen (gesichtet 4.11.1997):
`http://www.tenthmountain.com/html/set.html`

[URL-5] A. Benne. Rechtsinformatik-Seminar – „Zahlungen im Internet“ (gesichtet 4.11.97, genaue URL nicht mehr bekannt)
`http://www.uni-sb.de/...`

[URL-6] Informationen zum ECash-Pilotprojekt der Deutschen Bank (gesichtet 10.1.98)
`http://www.deutsche-bank.de/wwwforum/ecash/`

[URL-7] WWW-Seiten der Firma DigiCash mit Informationen zum ECash-System (gesichtet 10.1.98)
`http://www.digicash.com`

Sicherheitskonzepte für Firmen

Jörg Schramek

1 Einleitung

Innerhalb der letzten Jahre hat bei Unternehmen, Behörden, Instituten und Privatpersonen der Einsatz von IT-Systemen (Informationstechnologie-Systemen) stark zugenommen. In modernen, fortschrittlichen Unternehmen ist ein PC an keinem Arbeitsplatz mehr wegzudenken. Informationen und Unternehmensdaten werden zum Teil ausschließlich in digitaler Form gespeichert.
Sabotage, Diebstahl, physische Gewalt, Hard- und Softwarefehler, falsche Bedienung etc. stellen Bedrohungen für die Unternehmenssicherheit bzgl. der IT-Systeme dar. Seitens der Unternehmen wird die Sicherheit jedoch nicht ausreichend beachtet. Verkannt wird, daß es sich bei Sicherungsmaßnahmen um einen Vermeidungsnutzen und ggf. um Existenzsicherung handelt.
In dieser Arbeit wird zunächst auf eine Begriffserklärung von IT-Sicherheit eingegangen. Danach werden die Mängel in der Unternehmenssicherheit und ein ggf. resultierender Schaden erläutert. Prinzipielle Vorkehrungen gegen Bedrohungen werden durch die Erstellung eines Sicherheitskonzeptes dargestellt. Im vierten Kapitel werden konkrete bauliche, organisatorische und technische Maßnahmen angesprochen, die das Sicherheitsniveau erhöhen können.

2 Begriffserklärung

Unternehmenssicherheit umfaßt sowohl Datenschutz, Datensicherheit als auch IT-Sicherheit. Diese oftmals bedeutungsgleich verwendeten Begriffe sollen nachfolgend kurz erläutert werden.
Aus einem Rechtswörterbuch ist folgende Definition zum *Datenschutz* zu entnehmen: “Sicherung gespeicherter, personenbezogener Daten sowie Unterlagen und Ergebnisse vor Mißbrauch durch Einsichtnahme, Veränderung oder Verwertung unter Beeinträchtigung schutzwürdiger Belange des Betroffenen. Er dient dem Ausgleich zwischen dem Recht des Bürgers aber auch von Behörden und Unternehmen auf Information und dem Schutz des Persönlichkeitsrechts.”[Creif94]
Eine weitere Beschreibung von Datenschutz ist dem BDSG §1 Absatz 1 zu entnehmen: „Zweck dieses Gesetzes ist es, den einzelnen davor zu schützen, daß er durch den Umgang mit seinen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird.“

Datensicherheit kann als Ergebnis der Verwirklichung von Datenschutz und Katastrophenschutz (Sabotage, Brand, Blitzschlag, Überschwemmung etc.) angesehen werden. Sie umfaßt die Sicherstellung von [URL-5]:

- Verfügbarkeit,
- Integrität,
- Verbindlichkeit und
- Vertraulichkeit von Daten.

Nicht nur die Qualität von Informationen hat einen hohen Stellenwert, ebenso deren *Verfügbarkeit*, welche die funktionale Korrektheit aller Systemkomponenten eines IT-Systems einschließt. Daten oder Informationen, auf denen kein Zugriff besteht, haben keinen Wert.

Bei Daten, die verarbeitet und/oder kommuniziert werden, muß sichergestellt sein, daß diese nicht unbemerkt oder unautorisiert durch Fehlfunktionen oder durch Dritte verändert werden. Weiterhin muß vorausgesetzt werden, daß die Daten zusammenpassen, d.h. Eingriffe müssen so erfolgen, daß ein System oder Datenbestand in seiner Gesamtheit funktionsfähig und nicht manipuliert ist. In diesem Sachverhalt spricht man von der *Integrität* der Daten.

Die *Verbindlichkeit* elektronisch gespeicherter oder kommunizierter Daten ist als Vertrauensbasis für den elektronischen Geschäftsverkehr unerläßlich. Eine Nachricht gilt als verbindlich, wenn der Schutz der Urheberschaft und der Schutz der Originalität gewährleistet ist.

Sensible unternehmens- und personenbezogene Daten wurden, bereits vor der Nutzung von IT-Systemen, vor dem Zugriff Unbefugter geschützt, denn die Weitergabe personenbezogener Daten an Dritte unterliegt dem Datenschutzgesetz. Dieser Schutz muß auch gegeben sein, wenn Informationen in elektronischer Form gespeichert und kommuniziert werden. Es ist von großer Bedeutung, daß Privatsphäre und Betriebsgeheimnisse in Informationsnetzen verläßlich gesichert werden, d.h. die *Vertraulichkeit* geschützt wird.

Die *IT-Sicherheit* ist die Datensicherheit, bezogen auf die Systeme der Informationstechnik.

3 Sicherheitsaspekte

Vorhandene Sicherheitsmängel erhöhen die Verletzlichkeit eines Unternehmens. Welches Ausmaß ein Schaden (durch Naturgewalten, Spionage etc.) für ein Unternehmen annehmen kann, wird im folgendem Punkt verdeutlicht. Weiterhin werden Schwachstellen in der Unternehmenssicherheit beschrieben.

3.1 Schadensumfang

Aus den bereits genannten Bedrohungen kann ein Schaden resultieren, der direkte und/oder indirekte Kosten zur Folge hat. Letztere entstehen durch:

- Image-Verlust und folgende Kundenabwanderung
- Zeitverlust bzw. Produktionsausfall
- Folgekosten durch Geschäftsverlust
- Juristische Folgen durch Bekanntwerden personenbezogener Daten

Der Kostenausfall kann im Extremfall erheblich sein. Durch einen Totalausfall der EDV ist beispielsweise die zukünftige Existenz eines Unternehmens in Gefahr:

> „Demnach überleben Versicherungen einen Totalausfall ihrer DV 5,5 Tage, Produktionsunternehmen halten 5 Tage durch, Handelsunternehmen 2,5 und Banken nur noch ganze zwei Tage.“

> „[...] habe nachgewiesen, daß sogar 40 Prozent aller Betriebe, die einen Totalausfall ihrer DV ohne Notfallplan durchstehen mußten, innerhalb von zwei Jahren zusammenbrachen“ [URL-1].

Selbst in weniger schwerwiegenden Fällen, besteht das Problem der Rückstandsaufholung. Schon eine geplante Betriebsunterbrechung wie z.B. durch Umbau, verursacht eine Rückstandsaufholung, die einen fünffachen Zeitaufwand umfaßt. Ist die Betriebsunterbrechung ungeplant, liegt die Rückstandsaufholung in einer fünf- bis zehnfachen Zeit. [URL-2]

> „Betriebsunterbrechungen und Fehlerfolgen ergeben ein Schadensbündel, das kaum einer bislang kalkulieren kann. Aus Ereignissen wissen wir, daß sich die Verluste z.B. eines Handelsunternehmens infolge eines fünftägigen RZ-Ausfalls auf 16% des Jahresumsatzes belaufen haben“ [URL-2].

3.2 Schwachstellen in der Unternehmenssicherheit

Offensichtlich besteht durch die genannten Bedrohungen starker Handlungsbedarf. Man sollte annehmen, daß das Top-Management der Unternehmen mit angemessenen Maßnahmen die Bedrohungen auf ein Mindestmaß zu reduzieren versuchen. In der Praxis ist dieser Sachverhalt anders zu beobachten. Aus verschiedenen Erhebungen ist zu entnehmen, daß Unternehmen Sicherheitsmaßnahmen unzureichend umsetzen. Der SecuMedia Verlag erstellte 1996 eine Sicherheitsstudie [Hunn96] über die Sicherheit in der Informationstechnik, an der 183 Unternehmen beteiligt waren. Folgende Punkte geben einige Kernaussagen der Studie wieder:

- 62 % der Unternehmen antworteten, daß der Gefahrenbereich „Irrtum und Nachlässigkeit eigener Mitarbeiter“ die höchste Bedeutung hat. 34 % schätzten, daß die zukünftige Entwicklung in diesem Gefahrenbereich zunimmt, 12 % meinten, daß diese abnimmt.
- 62 % gaben an, daß keine schriftlich fixierte Strategie für die Informationssicherheit existiert.
- 56% offenbarten, daß schriftlich fixierte, spezifische Informationssicherheits-Konzepte / Richtlinien existieren. 67 % deckten hierbei Schwachstellen auf, bei 62 % dauert die Beseitigung von bekannten Schwachstellen an.

- 20 % sagten aus, daß für größere Katastrophen (Feuer, Wasser usw.) keine detaillierten Pläne / konkreten Handlungsanweisungen vorliegen.

Aufgrund dieser Zahlen kann man auf eine gewisse Sorglosigkeit von Seiten der Unternehmen schließen, welche begründet werden kann durch:

- den Glauben, daß Sicherheitsbestrebungen keinen direkten Nutzen erfüllen,
- den Glauben, daß Sicherheitsbestrebungen lediglich hohe Kosten und hohen Zeitaufwand verursachen,
- Konzeptlosigkeit und
- definierte Aufgabenverteilung.

Resultierend ergibt sich eine punktuelle Sicherheitsplanung, die sich in folgenden (und weiteren) Mängeln widerspiegelt:

- Nichtvorhandensein einer Einbruchsicherung,
- Lückenhafter Schutz vor Brand oder Wassereinbruch,
- Nichtvorhandensein von Zugangsbeschränkungen zu Räumen,
- Fehler („Bugs“) im Betriebssystem oder in Netzwerk-Dienstprogrammen,
- Schlecht konfigurierte Software (Bsp.: „Access Control Lists“ von Firewalls),
- Großzügige Vergabe von Benutzerrechten,
- Unkontrollierter Datenzugriff von Außen und
- Nichtvorhandensein eines Notfall- und/oder Wiederanlaufplans.

4 Erstellung eines Sicherheitskonzeptes

Um den oben aufgeführten Sicherheitsmängeln entgegenzuwirken, ist es notwendig ein Sicherheitskonzept zu entwickeln. Dieses muß von den Sicherheitsbeauftragten regelmäßig geprüft und ggf. angepaßt werden. Sicherheit stellt einen dynamischen Aspekt dar (z.B. aufgrund von neu installierter Software und den daraus folgenden neuen Sicherheitslücken). Es ist somit ein iterativer Prozeß, der die im weiteren erläuterten Punkte umfaßt [Frei97]:

- Istanalyse,
- Bedarfsanalyse,
- Bedrohungsanalyse,
- Erstellung von Sicherheitsmaßnahmen,
- Umsetzung,
- Schulung,
- Validierung und Aktualisierung.

In der Istanalyse wird festgestellt, welche Maßnahmen zur Sicherheit bereits implementiert wurden. Sie umfaßt sowohl die Gebäudesicherheit, die technische Sicherheit, die System- und Netzsicherheit als auch die Notfall-Planung.
Aufgabe der Bedarfsanalyse ist es, festzustellen, welche Daten und/oder welche Bereiche des Unternehmens besonders geschützt werden müssen. Hier ist eine Einschätzung über den möglichen Schadensumfang mit eingeschlossen. Zudem gibt es eine erste Schätzung über den Aufwand, der für die Firmensicherheit eingeplant werden muß. Dieser Aspekt ist wichtig, da die angestrebte Sicherheit stets im ausgewogenen monetären Verhältnis zum potentiellen Schaden stehen muß.
In diesem Zusammenhang stellt sich die Frage, vor wem oder was man sich schützen möchte Diese Frage ist Bestandteil der Bedrohungsanalyse. Hier muß zwischen externen und internen Bedrohungen unterschieden werden. Beide erfordern verschiedene Gegenmaßnahmen. Interne Sabotageversuche treten laut verschiedener Statistiken häufiger auf. [URL-1, URL-11]
Bei der Erstellung von Sicherheitsmaßnahmen muß zwischen verschiedenen Wirksamkeitsstufen unterschieden werden [URL-6]. Dazu gehören:

- Verhindernde Maßnahmen

Zum Beispiel die Aufstellung eines Servers oberhalb des Kellergeschosses verhindert mögliche Wasserschäden.

- Behindernde Maßnahmen

Die Entfernung eines Diskettenlaufwerkes aus einem PC erschwert beispielsweise das Einspielen oder unbefugte Kopieren von Dateien oder Programmen.

- Entdeckende Maßnahmen

Protokollierende Einrichtungen, die unzulässige Zugriffe festhalten, haben oftmals einen abschreckenden Effekt und ermöglichen, einen Angreifer zu entdecken.

- Bekämpfende Maßnahmen

Diese Maßnahmen können differenziert werden, zwischen der unmittelbaren Bekämpfung eines eingetretenen Ereignisses und der Eindämmung von Schadensfolgen.

Neben den technischen Fragen, sind zudem zeitaufwendigere organisatorische Fragen, welche die Verantwortlichkeiten und die Handlungsbefugnisse betreffen, zu klären. Eine anschließende *Umsetzung* der Maßnahmen sollte sukzessiv nach einem Zeitplan erfolgen, da technische und organisatorische Maßnahmen sehr zeitaufwendig sein können. Eine genaue Dokumentation über die durchgeführten Arbeitsschritte erleichtert auch unerfahrenen Administratoren eine Einarbeitung. Durch *Schulung* von Administratoren, Sicherheitsbeauftragten und Anwendern soll, das angestrebte Sicherheitsniveau erreicht und zukünftig gewährleistet werden. Besonders wichtig ist in diesem Zusammenhang die Mitarbeiterakzeptanz und Mitarbeitersensibilisierung für dieses Thema. Die Mitarbeiter müssen den Nutzen einer angemessenen Sicherheit erkennen. ‚*Validierung und Aktualisierung*‘ sollten in einem letzen Schritt regelmäßig durchgeführt werden. Dieser Schritt kann und soll bei hohem Sicher-

heitsbedarf von anderen Abteilungen oder externen Unternehmen durchgeführt werden, um einer „Betriebsblindheit“ für das selbst erstellte Konzept entgegenzuwirken.
Zu den allgemeinen und einschlägig bekannten Sicherheitskonzepten gehören beispielsweise das „Orange Book“ des amerikanischen Verteidigungsministeriums oder auch der IT-Katalog der IT-Sicherheitskriterien der Zentralstelle für Sicherheit in der Informationstechnik. Diese und andere Informationsquellen sind zahlreich im Internet vorhanden. [URL-12, URL-13, URL-14]

5 Gegenmaßnahmen

Es gibt eine Vielzahl von möglichen Gegenmaßnahmen, um sich sowohl vor Hackern, Großbränden oder anderen Ereignissen zu schützen oder zumindest die Schadensfolgen einzuschränken. Grundsätzlich ist zu beachten, daß Gegenmaßnahmen aufeinander abgestimmt werden müssen. Ein höherer Sicherheitsstandard ergibt sich nicht notwendigerweise durch die Addition von einigen Sicherherheitsmaßnahmen.

5.1 Bauliche Maßnahmen

Bauliche Maßnahmen verfolgen das Ziel Personal, Informationen und Hardware zu schützen. Sie sollten, wenn möglich, schon bei der Planung ausreichende Beachtung finden. Nachträglich umgesetzte Maßnahmen sind meist erheblich kostenintensiver und teilweise weniger wirkungsvoll. Bedrohungen entstehen in diesem Zusammenhang durch:

- Naturgewalten, Feuer und Überschwemmungen,
- Unbefugten Zutritt zu schutzbedürftigen Räumen,
- Stromausfälle,
- Überhitzung.

Eine zu kurzfristige Planungsperspektive ist zu vermeiden. Zu eng bemessene Stellflächen für besonders schutzwürdige Hardware (allgemein Rechenzentrumsfläche oder Server mit bedeutsamen Daten) führen zu einer Auslagerung von Funktionen in weniger gut gesicherte Bereiche, in Bereiche mit erhöhtem Brandlastumfeld, mit schlechter Zugangssicherung und anderen Problemen. IT-Planung sollte daher durch eine Flächenentwicklungsprognose ergänzt werden. Basis einer solchen Prognose können Branchenvergleiche, Trendaussagen, historische Entwicklungen oder andere Quellen bilden [URL-7].
Im folgendem werden einige bauliche Maßnahmen einschließlich haustechnischer Einrichtungen wie Klimaanlagen usw. nach Bedrohungsarten kategorisiert:
Brand:
Durch den Einsatz von Brandschutzmeldeanlagen kann ein Brand frühzeitig gemeldet werden. Eine Ausdehnung eines Brandes kann durch entsprechende Brandschutztüren, Brandschutzklappen (mit

nekaldichten[80] Rauchabschlüssen) und Brandschutzwänden eingeschränkt werden. Zu beachten ist, daß diese baulichen Elemente eine geringe Strahlungswärme aufweisen. Unterschiedliche Ausdehnungskoeffizienten der Materialien einer Platine oder der Chips führen bei Überschreitung der zulässigen Wärmetoleranzen zu Totalschäden. Weiterhin ist auf die Verwendung geeigneter Baustoffe zu achten, d.h.:

- keine brennbaren Wärmedämm- und Schallschutzmaßnahmen,
- keine brennbaren Füllungen für Dehnungsfugen in Trennwänden,
- Schottung gegen Leitungsdurchbrüche.

Überflutung und Wasserschäden:

Der Standort bedeutsamer IT-Geräte sollte so gewählt sein, daß diese nicht durch Überflutung oder durch Wasserschäden beschädigt werden können. Zusätzlich können in der Umgebung der Geräte Wassermelder installiert werden.

Unbefugter Zutritt:

Schutzbedürftige Räume sollten z.B. durch den Einsatz eines Sicherheitsdienstes oder durch verschließbare Türen gesichert werden. Hochsicherheitsbereiche können durch Fingerabdrucksysteme, Netzhautabtastung oder Stimmanalyse (biometrische Zugangskontrollen) gesichert werden.

Kompromittierende Strahlung:

Durch die Verwendung von abstrahlsicheren oder abstrahlarmen Geräte ist es möglich, sich zu schützen. Eine Abschirmung in Form eines faradayischen Käfigs bietet einen vollständigen Schutz gegen kompromittierende Strahlung

Stromausfälle:

Eine unterbrechungsfreie Stromversorgung schützt IT-Systeme vor Stromausfällen bzw. Stromschwankungen. Ein erhöhter Ausfallschutz wird durch Ersatzgeräte erreicht.

Überhitzung:

Die Benutzung von Klimageräten mit Ausfallschutz durch Ersatzgeräte bietet einen ausreichenden Schutz. Die Aufstellung solcher haustechnischen Einrichtungen sollten nicht in der Nähe von EDV-Bereichen erfolgen.

Wichtige Programme und Dokumente sollten zudem in Safes oder eigenen Tresorräumen gelagert werden, die ein Feuer mehrere Stunden überstehen und gegen Löschwasser Schutz bieten.

---

[80] Nekal ist eine Bezeichnung für höchste Dichtigkeit gegen Leckagen aller Art. Nekaldichte Brandschutzklappen schließen hochwertig fest. D.h., es kann kein meßbarer Rauch bei den üblich zu erwartenden Drücken auf die dem Feuer abgewandte Seite dringen.

5.2 Organisatorische Maßnahmen

Voraussetzung einer Erhöhung der Unternehmenssicherheit ist die Schaffung verbindlicher organisatorischer Regelungen. Arbeitsabläufe sind verbindlich festzulegen, Zuständigkeiten und Befugnisse sind zuzuordnen, der Umgang mit der Informationstechnik und vor allem mit den vorhandenen technischen Sicherheitseinrichtungen ist vorzugeben. [URL-8]
Ein höherer Sicherheitsgrad kann durch folgende Maßnahmen bzw. Regelungen erreicht werden (vgl. [Heid96]):

Zuweisung, Abgrenzung und Verwaltung von Rechten:

Die Vergabe von Zutrittsberechtigungen zu Räumen, Rechnern und Netzen ist kritisch und regelmäßig zu überdenken. Es muß eine Verwaltung und Kontrolle der Zugangsmittel durchgeführt werden. Zwecks der Absicherung der Netzzugänge sollte dem „Need-to-Do“-Prinzip gefolgt werden, d.h. daß nur Benutzer Zugang erhalten, bei denen es aktuell erforderlich ist. Desweiteren ist eine zeitliche Einschränkung von Zugängen denkbar, nachts kann somit ggf. der Netzwerkzugang verwehrt werden. Eine Rollentrennung sollte so erfolgen, daß die Administration von Sicherheitsmaßnahmen und die Kontrolle der getroffenen Maßnahmen in unterschiedliche Zuständigkeitsbereiche fällt.

Externe Datenschutzbeauftragte:

Der Einsatz von Datenschutzbeauftragten anderer Unternehmen weist Vorteile gegenüber unternehmenseigener Datenschutzbeauftragten auf. Externe Sicherheitsexperten verfügen über betriebsübergreifende Erfahrungen, ihr Sicherheitswissen steht zum geforderten Zeitpunkt zur Verfügung. Weiterhin sind sie nicht durch eine voreingenommene Sichtweise des Unternehmens beeinträchtigt. Ihre alleinige Aufgabe ist der Datenschutz und die Datensicherheit [URL-9]

Verwaltung und Kontrolle von Betriebsmitteln:

Der Einsatz von eingesetzter Hard- und Software ist zu dokumentieren und von einer zentralen Stelle zu regeln. Auf diese Weise ist die Einbringung von illegaler Software feststellbar (durch Softwaremaßnahmen automatisierbar). Regelungen für die kontrollierte Entsorgung von Datenträgern und Papier sind zu treffen. Außerdem ist der Einsatz privater Datenträger, Soft- und Hardware zu verbieten.

Betriebsvorschriften:

Bedienungsanweisungen für Softwareanwendungen, Anweisungen zur Systembedienung, Datensicherungsmaßnahmen (siehe unten) und Dokumente zur Katastrophenvorsorge sind Bestandteile der Betriebsvorschriften. Diese sollten den Mitarbeitern in Form von Schulungen vermittelt werden. Die Katastrophenvorsorge schließt einen Wiederanlaufplan mit entsprechenden Maßnahmen ein, so daß die Betriebsbereitschaft möglichst schnell wieder sichergestellt wird.

Sicherheitspezifische Vorschriften:

Die Vorschriften besitzen präventiven Charakter. Es handelt sich hierbei um Verhaltensregeln wie sie z.B. in [Heid94, S. 54 ff.] zu finden sind.

Überwachungsmaßnahmen:

Die Einhaltung der eingeführten Sicherheitsmaßnahmen muß am Arbeitsplatz, bei Betreten und Verlassen des Betriebsgeländes und bei Begleitung fremder Personen regelmäßig kontrolliert werden.

Mitarbeiter- und Unternehmensleitungssensibilisierung:

Mitarbeiter und Unternehmensleitung tragen entscheidend zur Sicherheit bei. Es muß eine Schulung der Anwender erfolgen, so daß:

- die Wichtigkeit von Sicherheit und Sicherheitsrichtlinien im Unternehmen erkannt wird,
- die Benutzung von Hard-, Software, Handhabung von Akten usw. bekannt ist,
- Sicherheitsprobleme erkannt werden können und
- auf Sicherheitsprobleme reagiert werden kann.

Weiterhin ist die Unternehmensleitung hinsichtlich der Notwendigkeit von baulichen, organisatorischen und technischen Sicherheitsvorkehrungen zu sensibilisieren.

5.3 Technische Maßnahmen

Bei den technischen Maßnahmen handelt es sich um software- und hardwaretechnische Maßnahmen sowie Verschlüsselungsverfahren und den Einsatz von Firewallsystemen.

5.3.1 Softwaremaßnahmen

Durch softwaretechnische Maßnahmen können Angreifer und Schadprogramme erkannt und letztere ggf. bekämpft werden. Weiterhin ist es möglich, den Gebrauch von Software zu reglementieren.

Zugangsberechtigung:

Identifikations- und Autorisierungsverfahren, die durch netzwerkfähige Betriebssysteme unterstützt werden, gewähren nur legalen Benutzern Zugriff auf Daten bzw. Peripherie. Die Verfahren schließen die Prüfung und Verwaltung von Rechten mit ein.

Softwareverteilungsprogramme:

In größeren Netzwerken kann durch Softwareverteilungsprgramme (halb-/voll-) automatisch Software von einem Server auf angeschlossene Clients installiert werden. Dies hat den Vorteil der Zeitersparnis und wirkt einer falschen Konfiguration auf den Clients entgegen. Arbeitsplatzrechner müssen durch diese Vorgehensweise nicht über CD-ROM- oder Diskettenlaufwerken zu Installationszwecken verfügen. Das Einspielen fremder oder illegaler Software wird dadurch verhindert.

Lizenzüberwachungsprogramme:

Durch diese Art von Programmen wird die Software in einem Netzwerk überwacht. Dabei wird die Anwenderanzahl der Programme mit begrenzter Lizenzenanzahl kontrolliert und die Ausführung illegaler Programme unterbunden.

Inventarisierungsprogramme:

Diese Programme sammeln Informationen über installierte Hard- und Software. Dadurch ist nachvollziehbar, welche Software sich in welcher Version auf einem Rechner befindet. Das Aufspüren illegaler Software wird hierdurch erleichtert.

Protokollierungsprogramme:

Sicherheitsrelevante Informationen wie z.B. die Kontrolle und Inanspruchnahme von Benutzerrechten, der Verbrauch von Systemresourcen und Zustand von Systemkomponenten werden durch diese Programme gesammelt. Mögliche Einbruchssversuche können durch eine Analyse der erstellten Protokolldateien (sogenannten Log-Dateien) erkannt werden.

Sicherheitsüberprüfungsprogramme:

Durch falsch konfigurierte Software oder durch andere Nachlässigkeiten wie z.B. die Wahl von „Standard“-Paßwörtern eröffnen Angriffsmöglichkeiten. Sicherheitsüberprüfungsprogramme können diese Sicherheitslücken aufspüren und auf erkannte Schwachstellen hinweisen. Bekannte Programme, auch als Hacker-Tools bezeichnet, sind beispielsweise „Satan“ für den Scan eines Netzwerkes oder „Crack“ zum Erraten von Paßwörtern. [URL-14]

5.3.2 Hardwaremaßnahmen

Durch den Einsatz von Hardwaremaßnahmen wird die Ausfallsicherheit erhöht. Dies wird in den meisten Fällen durch redundante Komponenten erreicht. Aus Sicherheitsgründen gibt es jedoch auch Softwarelösungen auf Hardwarebasis, da diese keine Manipulationen zulassen.

Redundante Geräte:

- Spiegelung von Festplatten
  Zwei Festplatten werden mit identischen Daten über einen Controller betrieben. Bei einem Ausfall erfüllt die zweite Festplatte die Aufgabe.
- Duplexing
  Über zwei Controller wird jeweils eine Festplatte betrieben. Bei dem Versagen eines Controllers oder einer Festplatte übernimmt der zweite Controller die Arbeit des anderen.

- RAID-Systeme

  RAID steht für „Redundant Array of Inexpensive Disks". Es werden mehrere physikalische getrennte Festplatten verwendet, die logisch zu einer einzigen Festplatte zusammengefaßt werden. Durch eine redundante Datenspeicherung kann eine hohe Sicherheit erreicht werden. Es existieren unterschiedliche Verfahren (Level 1-5), die in einfachster Form eine Spiegelung darstellen und in der intelligenteren Form Fehlerkorrekturdaten auf verschiedene Festplatten verteilen. RAID-Systeme führen in den höheren Stufen zu einer relativ hohen Ausnutzung der Kapazität mit Ausfallsicherheit bei geringen Leistungsverlusten.
- Clustering

In einem Server-Cluster werden mindestens zwei Ein- oder Mehrprozessorsysteme („Knoten") verbunden. Diese Gruppe von Knoten nutzt gemeinsame Datenbereiche und wird wie ein einziges System verwaltet. Durch diese Technik wird eine sehr hohe Verfügbarkeit der Anwendungen und Dienste sichergestellt.

Anti-Viren- und kryptographische Maßnahmen:

Die entsprechende Softwarelösungen befindet sich in einem ROM auf einer Steckkarte, die in einen Slot eines PCs gesteckt werden kann. Vor dem eigentlichen Bootvorgang wird das Programm aktiv. Aufgrund der Speicherung in einem ROM kann das Programm nicht durch einen Virus manipuliert oder deaktiviert werden. Bei kryptographischen Verfahren ist die höhere Leistung gegenüber von Softwarelösungen als vorteilhaft zu erwähnen.

Verwendung isolierter Testrechner bzw. Schleusenrechner:

Bei hohen Sicherheitsanforderungen sollte neue Software auf Testrechnern oder Testrechnernetzen installiert werden. Diese kann dann ohne Gefahr für andere Komponenten auf ihre Funktionalität geprüft werden. CDs und Disketten könne gefahrlos auf Viren untersucht werden.

5.3.3 Firewallsysteme

Bei einer Firewall handelt es sich um einen oder mehrere Rechner. Das zu schützende Netz ist mit dem bedrohenden Fremdnetz durch die Firewall verbunden, so daß jegliche Datenkommunikation über diese erfolgen muß. Durch entsprechende Software werden Datenpakete gefiltert. Auf diese Weise wird nur ein zulässiger Datenverkehr zugelassen. Für weitere Informationen siehe den Seminarbeitrag „Firewalls".

5.4 Weitere Maßnahmen

Es existieren weitere Maßnahmen wie Datensicherungsmaßnahmen und die Beauftragung von Dienstleistern für Notfallprogramme, die sich nicht in die oben aufgeführten Bereiche zuordnen lassen. Dabei handelt es sich um eine Mischung von organisatorischen und technischen Maßnahmen.

5.4.1 Datensicherung

Von der technischen Seite gehört die entsprechende Hardware (z.B. DAT-Streamer) und Software (Datensicherungsprogramme) zu dieser Maßnahme. Organisatorisch sind Regelungen festzulegen,

- wer für die Datensicherung verantwortlich ist,
- auf welche Weise und wie oft diese durchzuführen ist und
- wo Sicherungsmedien aufzubewahren sind (zentral oder dezentral).

5.4.2 Backup Anbieter

Nach der Schätzung eines möglichen Schadensumfanges und nach der Abwägung, wie lange ein Ausfall der Rechenleistung vertretbar ist, kann ein Unternehmen einen Dienstleister mit „kalten“ oder „warmen“ Backup-Lösungen beauftragen. Bei der „warmen“ Backup-Strategie wird ein betriebsbereites Computersystem bereitgehalten, das im Notfall eingesetzt wird. Abhängig von der Hardwarekonstellation ist ein Wiederanlauf binnen 24 Stunden [URL-10] durchführbar. Bei der „kalten“ Variante werden kurzfristig Gebäude und Rechner angemietet. Die Wiederanlauf-Zeit ist hierbei jedoch wesentlich länger.

6 Zusammenfassung

Unternehmen sind von gespeicherten und kommunizierten Daten und deren Integrität, Verfügbarkeit etc. stark abhängig, so daß mögliche Schadensfälle die Unternehmensexistenz gefährden können. Wie aufgezeigt wurde, existieren Bedrohungen unterschiedlichster Herkunft. Diese spiegeln sich in Schwachstellen der Unternehmenssicherheit wider. Durch die Umsetzung eines erstellten Sicherheitskonzeptes kann ein erhöhter Sicherheitsgrad erreicht und ein potentieller Schaden auf ein Minimum reduziert werden. Die vorgestellten Gegenmaßnahmen veranschaulichen Möglichkeiten mit verschiedenen Wirksamkeitsstufen wie die Sicherheitsmaßnahmen eines Sicherheitskonzeptes umgesetzt werden können.

7 Literatur- und Quellenverzeichnis

7.1 Printmedien

[Crei94] Creifelds, Rechtswörterbuch 12. Auflage 1994, C.H. Beck Verlag, München 1994.

[Fras95] H.-J. Frase. „Katastrophenvorsorge in IT-Bereichen – eine Aufgabe ganzheitlichen Managements“. Datensicherheitsreport. Vogel Verlag, Würzburg, Nr. 9, 1995.

[Heid96] B. Heidecke. „Analyse und Bewertung von Sicherungskonzepten in Client/Server-Systemen am Beispiel von Novell-NetWare“. Diplomarbeit Fachbereich Informatik, Universität Dortmund, 1996.

[Hunn96] G. Hunnius. „So schätzen DV-Anwender ihre Sicherheit ein“. KES Zeitschrift für Kommunikations- und EDV-Sicherheit. SecuMedia Verlag, Ingelheim, Nr. 3, 1996.

[Frei97] Freiss, Martin. „SATAN: Sicherheitsmängel erkennen und beheben“. O‘Reilly / Thomson Verlag, 1997.

7.2 Elektronische Dokumente

[URL-1] G. Schmidt. „DV-Sicherheit – den wenigsten Anwendern ist klar, wie sehr sie von ihrer Datenverarbeitung abhängig sind“. Computerwoche Nr. 23. Stand 08.06.1990 (gesichert am 29.11.97)
`http://www.computerwoche.de/archiv/1990/9023c112.html`

[URL-2] SiLine. Rainer von zur Mühlen. “Backup auch für kleine DV-Systeme – ist das denn nötig?“, Stand 05.08.1997 (gesichert am 01.12.1997)
`http://www.siline.com/890_itsicher/890_backup.html`

[URL-3] Zbinden Infosec. „Datensicherheit: Eine Aufgabe des Managements“. Stand 01.01.1998 (gesichert 04.01.1998)
`http://www.infosec.ch/artikel/a_024.htm`

[URL-4] Deutsche Forschungsanstalt für Luft- und Raumfahrt. „Sicherheit für die Informationsgesellschaft“. Schriftenreihe des Fachverbands Informationstechnik. Stand unbekannt (gesichert am 06.01.1998)
`http://www.bmwi-info2000.de/gip/studien/sicher/sicher_1.html`

[URL-5] K. Pommerening. Vorlesung Datenschutz und Datensicherheit. Stand 23.09.1996 (gesichert 01.12.1997)
`http://www.uni-mainz/~pommeren/DSVorlesung/Begriffe.html`

[URL-6] SiLine. Rainer von zur Mühlen. “Wege zur Sicherung in der Informationsverarbeitung am Beispiel strategischer Ansätze“. Stand 23.02.1997 (gesichert am 01.12.1997)
`http://www.siline.com/890_itsicher/890_strategie.html`

[URL-7] SiLine. „Rechenzentren unter Sicherheitsaspekten planen“. Stand 23.02.1997 (gesichert am 01.12.1997)
`http://www.siline.com/890_itsicher/890_rz.html`

[URL-8] Berliner Datenschutzbeauftragter. „Begriffe im Zusammenhang mit dem technisch-organisatorischen Datenschutz“. Stand 28.02.97 (gesichert am 06.01.1998)
`http://www.datenschutz-berlin.de/to/begriffe.htm`

[URL-9] Martin Lackmann-Gubela. „Datenschutz nützlich machen“. Stand 05.08.97 (gesichert am 06.01.1998)
`http://www.siline.com/890_itsicher/890_datenschutz.html`

[URL-10] Hewlett Packard. Dienstleistungen. „Sicherungskonzepte“. Stand 23.10.97 (gesichert 29.11.97)
`http://www-1.hewlett-packard.de/germany/CSS-G/produkte/sicherungskonzepte.html`

[URL-11] H. Honermann. „Umfassende Bewertung der Sicherheit von Netzwerken in Unternehmen“. Stand 1997 (gesichert 01.12.97)
`http://www.bdg.de/Wpapers/ISS97.htm`

[URL-12] Zahlreiche Links zum Thema „Sicherheit“. Europäisches Institut für Systemsicher-heit E.I.S.S.. Stand 30.06.97 (gesichert 02.02.97)
`http://iaks-www.ira.uka.de/ta/Security/security.html`

[URL-13] Zahlreiche Links zum Thema „Sicherheit“. Universität Mainz. Stand 27.11.97 (gesichert 02.02.97)
`http://www.uni-mainz.de/FB/Medizin/IMSD/AGDatenschutz/Ressourcen.html`

[URL-14] CERT. Computer-Notfall-Team für das Deutsche Forschungsnetz und seine Dienste. Stand 1997 (gesichert 01.12.97)
`http://www.cert.dfn.de`