Summary

Kurs 1866
Sicherheit im Internet 1

Inhaltsverzeichnis

1. Sicherheit in der Informationstechnik

1.1 Einführung: siehe Skript.

1.2 Einleitung

1.2.1 Warum ist Sicherheit erforderlich?

a) Hauptgrund ist die große Popularität des Internet.

Private Geschäftstätigkeiten über Internet (Einkauf, Versteigerungen, Bankgeschäfte, Reise-/Ticketbuchungen u.v.m.):

- keine Bindung an Ladenöffnungszeiten
- keine Ortsgebundenheit des Geschäfts
- leichter Vergleichbarkeit der Angebote

Wirtschaftliche Bedeutung des Internet:
Wird zur Abwicklung von Geschäften zwischen Firmen (B2B) und zum Endkunden (B2C) genutzt. Bsp.:

- In Deutschland werden ca. 12,4 Mill. Bankkonten (ca. 80% der Girokonten bei privaten Banken) online geführt *(Stand 2003)*
- 2002 wurden in Deutschland etwa 767 Mill. Online-Überweisungen mit einem Umsatz von ca. 660 Mrd. Euro durchgeführt.
- 2003 wurden in den USA im E-Commerce ein Umsatz von ca. 17,2 Mrd. US-$ gemacht.

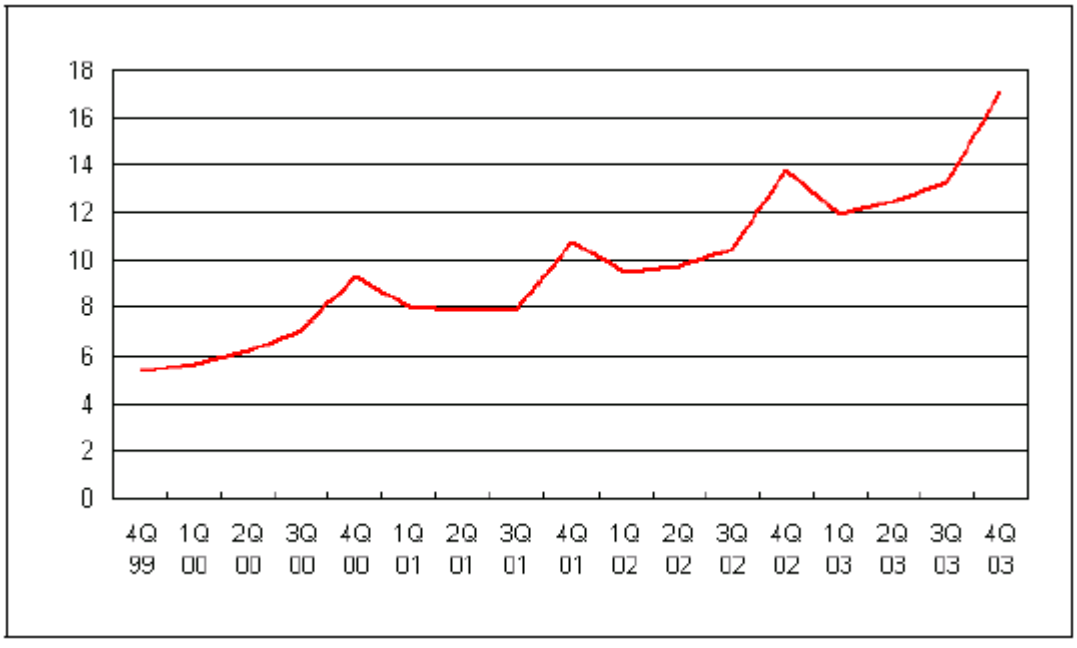


Abb. 1.1: E-Commerce Umsätze in Mrd. US-$ in den USA

- Amazon hat im Weihnachtsgeschäft 2003 an einem Spitzentag weltweit mehr als 2,1 Mill. Artikel verkauft (ca. 24 Artikel/sek.) und im 4.Quartal 2003 einen Umsatz von 2 Mrd. US-$ erwirtschaftet.
- Der amerikanische Aktien-Broker Charles Schwab hatte Ende 2003 etwa 7,5 Mill. aktive Kundenkonten, davon haben 4 Mill. mindestens einen Online-Trade durchgeführt. Die Kunden haben dort bereits 967 Mrd. US-$ angelegt.

b) Sicherheitsprobleme richten nicht unerheblichen Schaden an.

Anlaufstelle für Meldungen von Sicherheitsvorfällen und für Informationen über Abwehrmaßnahmen:

- CERT (Computer Emergency Response Team): www.cert.org
- DFN (Deutsches Forschungsnetz): www.cert.dfn.de

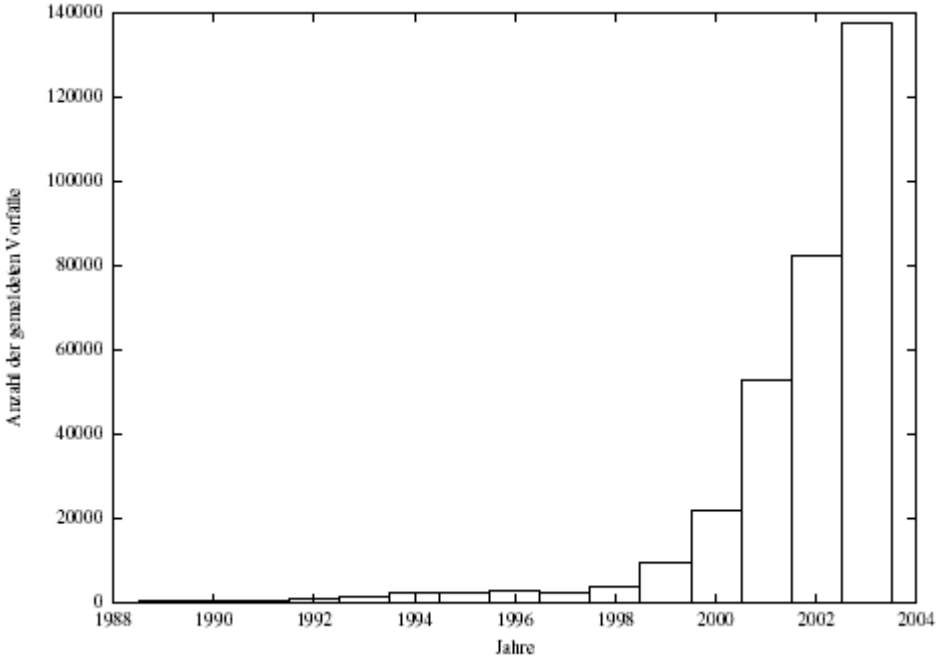


Abb. 1.2: Dem CERT gemeldete Sicherheitsvorfälle

c) Gesetzliche Verpflichtungen:

- KonTraG – Gesetz zur Kontrolle und Transparenz im Unternehmensbereich. Es zwingt börsenorientierte AGs zur Risikovorsorge, d.h. auch jene Risiken, die durch das IT-System entstehen auf ein vertretbares Restrisiko zu minimieren.
- Basel II – Basel Committee on Banking Supervision. Die Höhe des Eigenkapitalanteiles, der für vergebene Kredite vorgehalten werden muss, richtet sich nach der Höhe des Risikos des betroffenen Kredites bzw. Kreditnehmers. D.h. u.a. dass auch das operationelle Risiko von Unternehmen bewertet wird, was bei IT-lastigen Firmen letztlich auch eine Bewertung des Sicherheitsrisikos des IT-Systems des Unternehmens bedingt.

1.2.2 Was heißt eigentlich Sicherheit?

Lexikal. Def.: Sicherheit bedeutet objektiv das Nichtvorhandensein von Gefahr , subjektiv die Gewissheit, vor möglichen Gefahren geschützt zu sein.

In der IT-Sicherheit sind neben der abstrakten Bedrohung auch die potentiellen Schäden zu betrachten.

1.2.3 Angriffsziele

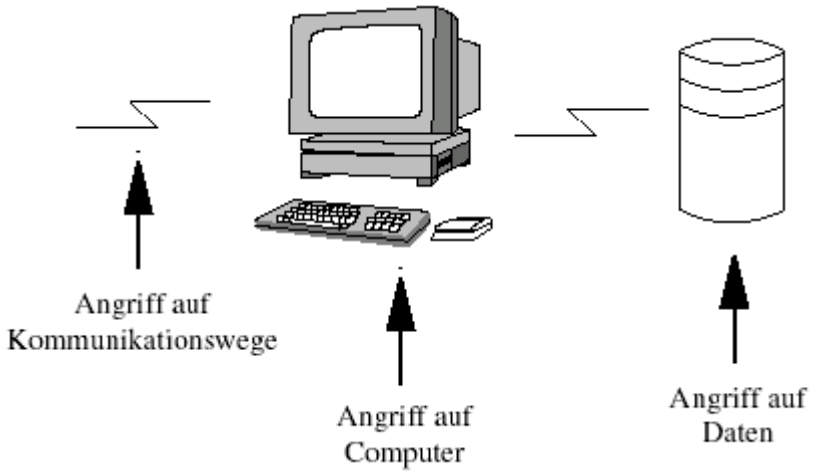


Abb. 1.3: Ziele von potentiellen Angriffen auf die Sicherheit

- Abhören der Kommunikationswege
- Programme
  - zum Ausspionieren und verfälschen der eigenen Daten
  - um andere Programme zu beeinflussen
  - den Computer unbenutzbar zu machen
- unbefugter Zugriff auf Datenbanken od. Datenträger od. Rekonstruktion von Bildschirminhalten durch Messung der emittierten Strahlung bei CRT's.

1.2.4 Systematik der Bedrohung

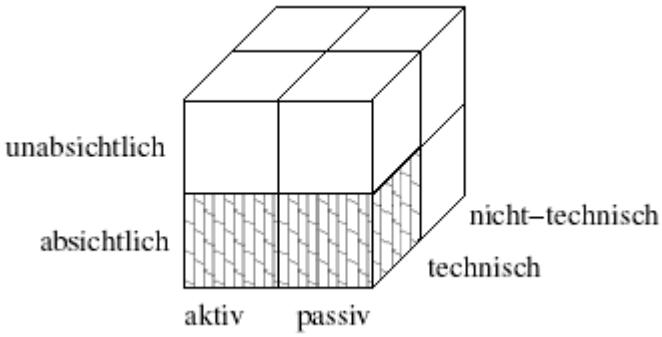


Abb. 1.4: Klassifikation von Bedrohungen

- *technische*: z.B. elektrische Probleme auf dem Übertragungsmedium, Ausfall der Stromversorgung
- *nicht-technische*: z.B. Person, die absichtlich Daten verfälscht und überträgt
- *absichtlich*: DoS, Spionage
- *unabsichtlich*: Bedienungsfehler, offener Umgang mit Passwörtern
- *aktiv*: erzeugen neuer Nachrichten, Unterdrücken od. Verzögern von Nachrichten, Fälschen von Nachrichten
- *passiv*: jede Form des Abhörens

1.2.5 Klassische Bedrohungen

Angriff auf die Vertraulichkeit:

Unbefugter Informationsgewinn aus den übertragenen Daten, ist ein *passiver* Angriff, der *absichtlich* oder *unabsichtlich* erfolgen kann.

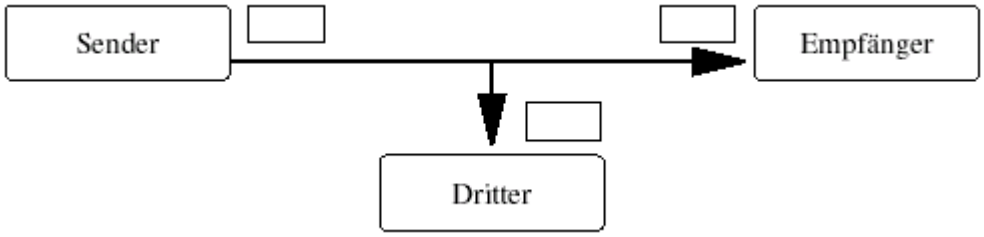


Abb. 1.5: Unbefugter Informationsgewinn

Gegenmaßnahmen:

- Verschlüsselung
- Steganographie – Nachricht in einer anderen, unverfänglichen größeren Nachricht verstecken.

Angriff ist nicht auf Abhören von Datenübertragungen begrenzt, sondern ist z.B. durch Zugangsverschaffung direkt zu einem Computer mit sensiblen Daten ebenfalls realisierbar.

Angriff auf Integrität:

Unbefugte Modifikation der übertragenen Daten, ist *aktiver* Angriff der i.d.R. *absichtlich* erfolgt. Eine unbefugte Modifikation ist aber auch aufgrund technische Probleme möglich.

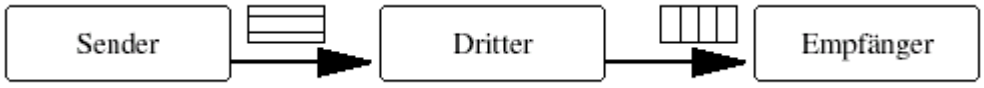


Abb. 1.6: Unbefugte Modifikation

Gegenmaßnahmen:

- Redundanz in den Daten – z.B. Parity-Check. Ist jedoch nur zur Absicherung gegen technische Probleme brauchbar. Zum Schutz der Datenintegrität aber nicht brauchbar, da
  - diese Verfahren nur bestimmte Veränderungen an den Daten auf Bitebene erkennen, andere Veränderungen (Inhaltliche) bleiben unerkannt,
  - ein Angreifer bei bekannten Redundanzverfahren ebenso die Redundanzdaten mitverändern kann.
- Verschlüsselung

Angriffe könne hier ebenfalls direkt an einem Computer mit sensiblen Daten erfolgen. Darüber hinaus besteht ein bestimmtes Gefahrenpotential durch ungeübte Benutzer, die Programme falsch bedienen und so unbeabsichtigt Daten modifizieren.

Angriff auf Authentizität:

Unbefugte Erzeugung von Daten. Angreifer erzeugt Nachricht unter Vorgabe einer falschen Identität.

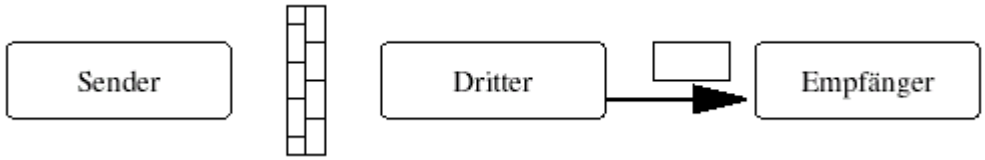


Abb. 1.7: Unbefugte Erzeugung von Daten

Das Problem ist eng verwandt mit dem Problem der Nicht-Zurückweisbarkeit, d.h. weder Sender noch Empfänger können die stattgefundene Kommunikation abstreiten, also

- der Empfänger muss beweisen können, dass die Nachricht tatsächlich vom vermeintlichen Absender kommt
- der Sender muss beweisen können, dass die Nachricht tatsächlich beim Empfänger und nicht bei jemand anderem angekommen ist.

Gegenmaßnahmen: siehe Kapitel 2.6

Angriff auf Verfügbarkeit:

Unbefugte Unterbrechung ist ein Angriff auf die Verfügbarkeit von Daten, Computern oder Kommunikationsmitteln. Es sind *aktive* Angriffe die i.d.R. *absichtlich* erfolgen wie z.B. DoS oder spezielle Programme (z.B. als Anhang an Nachrichten), die das OS um Absturz bringen. Außerdem sind natürlich auch unbeabsichtigte Unterbrechungen aufgrund technischer Probleme möglich.

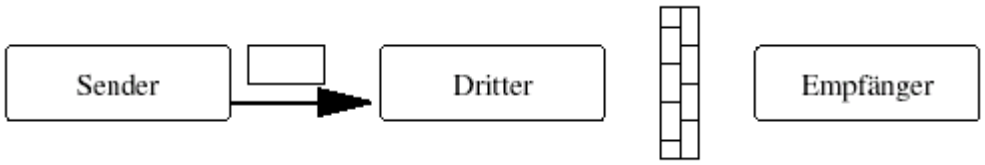


Abb. 1.8: Unbefugte Unterbrechung

Gegenmaßnahmen:

- Hochverfügbare System – redundante Systeme mit identischer Konfiguration – gut geeignet bei Unterbrechungen aufgrund technischer Probleme, nicht jedoch bei gezielten Angriffen, die das OS zum Absturz bringen.
- Architekturen, die nicht durch einen Fehler an einer einzelnen Stelle (*single point of failure*) lahm gelegt werden können.

1.2.6 Nicht-technische Aspekte von Sicherheit

Aus der Sicht des Benutzers sollte ein System *verlässlich* (technische Sicherheit) und *beherrschbar* (nicht-technische Sicherheit) sein. Die Beherrschbarkeit bedeutet letztlich, dass der Einzelne und auch die gesamte Gesellschaft vor unerwünschten Auswirkungen neuer Technologien und Systemen geschützt werden müssen. Bestes Beispiel für solche unerwünschten Auswirkungen ist die Liste der möglichen Daten, die über eine Person aufgrund der weltweiten elektronischen Vernetzung, E-Commerce, E-Banking usw. bereits heute gesammelt werden können:

- **Finanzdaten** – Konten, Verdienst ...
- **Konsumdaten** – Kaufverhalten ...
- **Kommunikationsdaten** – e-mail, Telefon ...
- **Aufenthaltsdaten** – Überwachungskameras, biometrische Reisepässe ...

Mittels ***data mining*** kann man daraus erschreckend komplexe Dossiers über eine Person erhalten.

Die Beherrschbarkeit von Systemen wird aufgrund der zunehmenden Komplexität, durch z.B. Vernetzung, eine immer bedeutendere Frage.

1.3 Netze

1.3.1 Lokale Netze – LANs

Sterntopologie:

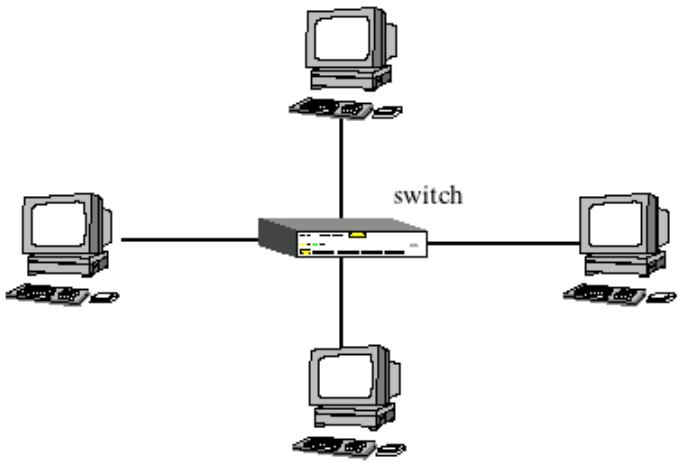


Abb. 1.9: Prinzip der Sterntopologie

Bsp.: ATM (Asynchronous Transfer Mode)-Netz der Telefongesellschaften
Auswirkungen auf die Sicherheit:

- Der Ausfall eines Computers wirkt sich nicht auf die *Verfügbarkeit* des restl. Netzes aus.
- Der Ausfall des Switch unterbricht die Kommunikation (*Verfügbarkeit*).
- Da die Daten zwischen 2 Computern nur den Sender, den Switch und den Empfänger passieren, können andere Computer die Kommunikation nicht stören oder abhören (*Vertraulichkeit*). Voraussetzung: Angeschlossene Computer täuschen keine falschen Absenderadressen vor, so dass der Switch eine falsche Adresstabelle erstellt.

Ringtopologie:

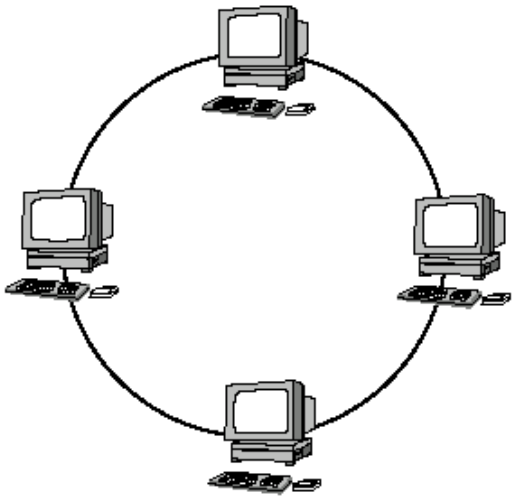


Abb. 1.10: Prinzip der Ringtopologie

Bsp.: Token-Ring von IBM.
Die Nachricht wird von Computer zu Computer weitergereicht. Jeder Teilnehmer kann also das Nachrichtenpaket lesen.
Auswirkungen auf die Sicherheit:

- Der Ausfall eines Computers oder einer Verbindung unterbricht die Kommunikation (*Verfügbarkeit*)
- Eine Nachricht kann im Prinzip auf jedem Computer gelesen werden (*Vertraulichkeit*) und damit auch verfälscht (*Integrität*) oder unterdrückt werden.

Bustopologie:

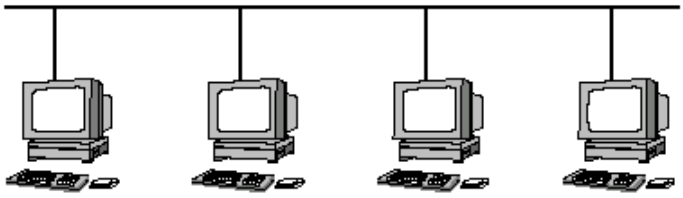


Abb. 1.11: Prinzip der Bustopologie

Bsp.: Ethernet
Alle angeschlossenen Computer können versendete Nachrichten lesen. Beim gleichzeitigen Senden von mehreren Computern treten Kollisionen auf, die durch spezielle Mechanismen (CSMA/CD – Carrier Sense Multiple Access/Collision Detection) behoben werden.
Auswirkungen auf die Sicherheit:

- Der Ausfall eines Computers wirkt sich nicht auf die *Verfügbarkeit* des restl. Netzes aus.
- Versucht ein Computer ständig zu senden, so können die anderen Computer u.U. nicht mehr kommunizieren (*Verfügbarkeit*).
- Jeder angeschlossene Computer kann alle Nachrichten mitlesen (*Vertraulichkeit*).

1.3.2 Vernetzte Netze

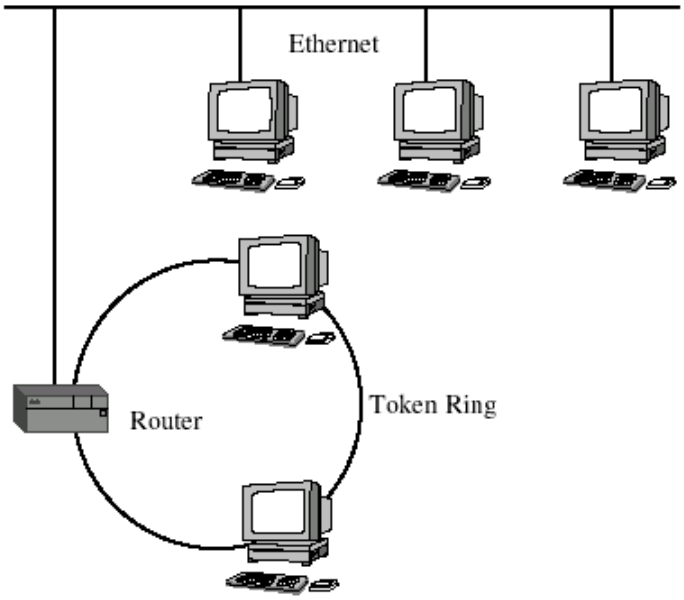


Abb. 1.12: Verbindung von Ethernet u. Token-Ring durch einen Router

Gerätetechnologien zur Verbindung von Netze:

- **Repeater**: Reiner Verstärker (verstärkt neben den Signalen auch die Störungen) zur Verbindung zweier gleichartiger LANs.
- **Bridge**: Im Prinzip wie Repeater, gibt aber nicht alle Signale weiter, sondern nur vollständige und fehlerfreie Datenrahmen. Anhand den Absenderadressen in den Rahmen „lernt" die Bridge mit der Zeit, welche Computer in welchem Netz liegen. Rahmen werden nur dann weitergeleitet, wenn der Zielcomputer in einem anderen Netz liegt. Struktur und Datenrahmen müssen in beiden Netzen gleich sein. Switches realisieren im Ethernet für jeden Computer einelementige Subnetze, die über Bridges mit anderen Subnetzen verbunden sind.
- **Router**: ist ein dedizierter Computer für den Zusammenschluss von Netzen auch mit unterschiedlicher Technologie. Besitzt eine getrennte Schnittstelle (Netzwerkkarte) für jeden Netztyp.

1.3.3 Das Internet-Protokoll

TCP/IP ist die häufigste implementierte Protokollfamilie.

Protokollschichten:

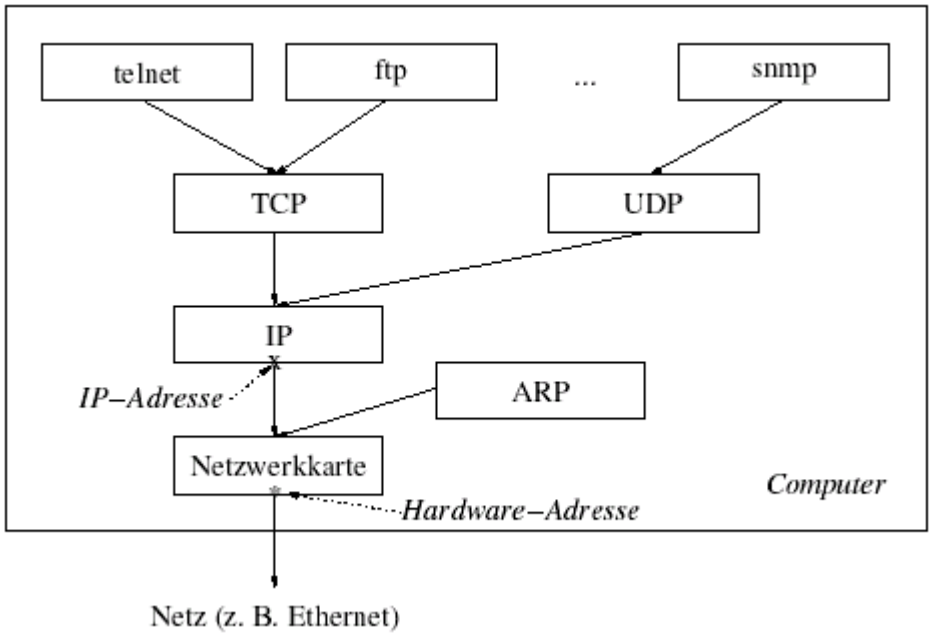


Abb. 1.13: Ablauf der Kommunikation bei TCP/IP

Adressierung:

Jede Netzwerkkarte hat eine eindeutige Adresse, welche von der verwendeten Netzwerktechnologie abhängt (Ethernet -> 6 Byte lang). Im Internet wird von der HW-Adresse durch das *IP-Adressschema* abstrahiert (4 Byte lang). Das Präfix identifiziert das Netz, das Suffix den Computer. Die symbolische Präsentation der IP-Adressen durch Namen (URL's) erfolgt durch das *DNS (Domain Name System)*. DNS-Server cachen einmal angefragte Namensauflösungen, um den Anfrageverkehr zu höheren Ebenen in der DNS-Server Hierarchie zu vermindern. Damit besteht aber auch das Risiko, das der angesprochene DNS-Server eine falsche IP-Adresse für den angefragten Namen gespeichert hat.

Die Auflösung von IP-Adresse auf HW-Adresse der Netzwerkkarte erfolgt durch das ARP (Address Resolution Protocol). Es erfolgt dabei eine Broadcast-Anfrage an alle im Netz

direkt erreichbaren Computer welche HW-Adresse zur angefragten IP-Adresse gehört. Dabei könnte aber auch ein Computer bewusst vorgeben, die angefragte IP-Adresse zu besitzen.

Die Adressierung eines Dienstes oder Anwendung auf einem Computer erfolgt durch die *Port*-Nummer, die an die IP-Adresse mittels Doppelpunkt angehängt wird, z.B. 172.10.12.144:80.

1.3.4 Die Internet-Dienste

Nslookup:

Dient zur Abfrage eines DNS-Server (Rechnername -> IP-Adr. oder umgekehrt). Abbildung Rechnername auf IP-Adr. nicht unbedingt bijektiv, es sind auch mehrere Namen einer IP-Adresse zuordenbar. Nslookup wird unter UNIX zukünftig durch „*dig*“ ersetzt, womit auch noch Informationen über den DNS-Server abgefragt werden können. Damit kann man überprüfen ob die angefragte Namensauflösung etwa aus dem Cache des DNS-Server stammt und somit vielleicht nicht mehr gültig ist.

Ping:

Zum testen, ob ein Computer über das Netz erreichbar ist (es werden IP-Pakete an den entfernten Computer gesendet, welche von dem angesprochenen Computer einfach wieder retourniert werden – Echo). Dabei wird die „turn-around-time“ und die Vollständigkeit analysiert.

Finger:

Damit kann herausgefunden werden, wer an einem entfernten Computer angemeldet ist und weitere Informationen über den angemeldeten User. Da dies als Vorbereitung von Angriffen genutzt werden kann, wird dieser Dienst meistens deaktiviert.

Traceroute:

Dient zur Verfolgung des Weges von IP-Paketen vom eigenen Rechner bis hin zum angesprochenen entfernten Rechner. Damit kann man Informationen über die Struktur der Netzwerkverbindung erhalten. Die Netzwerkstrukturen speziell im Internet sind i.A. aber nicht stabil. Router können auch so konfiguriert werden, dass man mit *traceroute* keine Informationen über den Weg der Pakete erhält. Intranets sind normalerweise ebenfalls so nicht von außen zu analysieren.

Telnet:

Ist ein Terminal-Programm um sich zu einem beliebigen entfernten Computer zu verbinden und auf diesem zu arbeiten, als stünde ein direkt angeschlossenes Terminal zur Verfügung. Da alle Daten unverschlüsselt übertragen werden (auch *username* und *password*), wird heute überwiegend nur mehr *ssh* (Secure Shell) für *remote terminals* benutzt.

File Transfer:

Mit dem Programm *ftp* können Dateien zwischen dem lokalen und dem entfernten Computer übertragen werden. Beim Aufbau der Verbindung werden aber wie bei *telnet* die Daten für *username* und *password* unverschlüsselt übertragen. Es gibt aber auch die Möglichkeit eines „*anonymous*“-Login, die ohne Password funktioniert.
Sicherheitsrisiken:

- Übertragung des Passworts kann abgefangen werden (wie auch bei *telnet*).
- Dritte können auch urheberrechtlich geschützte Daten auf den *ftp*-Server kopieren, wodurch sich der Betreiber des *ftp*-Servers evtl. strafbar macht.

- Der *account* und das *password* könnten auch normale Benutzerkennungen sein und somit wäre auch ein *login* über *telnet* auf dem *ftp*-Server möglich.

World Wide Web:

WWW ist multimediabasiert und an graphischen Benutzeroberflächen orientiert (im Gegensatz zu den oben besprochenen textbasierten Diensten) und es bietet zusätzlich Möglichkeiten des Information Retrieval.
Als Web-Client wird ein Web-Browser verwendet um auf Inhalte von Web-Servern zuzugreifen. Die Kommunikation zw. Client u. Server erfolgt über HTTP, wobei der HTTP-Dienst an Port 80 läuft.
Dokumente auf Web-Server sind in HTML geschrieben, eine SGML(Standard Generalized Markup Language)-Anwendung für die vom W3C die entsprechenden DTD (Document Type Definition) als Regulative für die erlaubten Bestandteile von HTML-Doks festgelegt wurden. Zukünftig sind auch XML-Doks geplant, eine Weiterentwicklung von SGML.
Web-Server und die Dokumente werden über URL (Unified Resource Locator) adressiert, welcher folgendermaßen aufgebaut ist:

Bsp.:

Dienst	Web-Server Name	Dokument Name
http	://www.fernuni-hagen.de	/info.html

Die Kommunikation zwischen Client u. Server erfolgt nach dem „*request – response*“ Prinzip. Die gesamte Kommunikation findet unverschlüsselt statt, außerdem kann weder der Client sicher sein, dass er mit dem gewünschten Server kommuniziert noch umgekehrt (Problem der **Authentizität**).

E-Mail:

Der Versand der Daten wird im SMTP (Simple Mail Transfer Protocol) festgelegt, wobei der SMTP-Dienst an Port 25 läuft. Mit einem E-Mail Client können E-Mails geschrieben, verwaltet, empfangen und versendet werden. Der Client kommuniziert dazu mit einem Mail-Server, der E-Mails vom Client für den eigentlichen Versand entgegennimmt bzw. E-Mails von anderen Absendern empfängt und diese für den Client zum Download bereithält. Dafür wird das POP3 (Post Office Protocol) bzw. das neuere IMAP4 verwendet.
Die E-Mail besteht aus einem „*Envelope*“, in dem alle Empfängeradressen stehen und welches der Mail-Server für den Versand verwendet. Die Nachricht selbst besteht dann aus einem *message header*, in dem nochmals die Absender- u. Empfängeradressen wiederholt werden, und einem *message body* mit dem Nachrichteninhalt. Bei Spams enthält der *message header* meist gefälschte Einträge.
Die Kommunikation zwischen Client und Mail-Server erfolgt unverschlüsselt. (Moderne Mail-Server unterstützen mittlerweile SSL für eine verschlüsselte Verbindung zw. Client u. Server.) Die Nachricht kann also leicht abgehört werden. Jeder beteiligte Mail-Server trägt bei der Weiterleitung im Nachrichtenkopf eine Zeile „*Received*“ inkl. Zeitstempel ein. Damit kann der Weg einer E-Mail rückverfolgt werden. Diese Information kann aber auch der Absender eintragen und somit den Weg der E-Mail verschleiern.
Das E-Mail System ist also so aufgebaut, das der E-Mail Client nicht ständig online sein muss sondern dessen „Eingangspost“ am Mailserver bereitgehalten wird (bei Providern für Privatkunden meist nur bis zur einer bestimmten Speicherplatzgröße).

1.4 Konkrete Gefahren

1.4.1 Viren

Prinzip :

Viren sind Computerprogramme die sich selbst kopieren und somit vermehren können.
Weitere Funktionen können sein:

- Anrichten von Schäden aller Art auf einem Computer
- Versuchen, sich selbst zu tarnen und zu verstecken
- Durch seltsame Ausgaben der Benutzer verunsichern u. erschrecken

Schäden:

Mögliche Schäden und damit Schadensumfang reichen von eigenartigen Ausgaben, über System-Reboots, Löschen von Dateien, schleichendes Löschen einzelner Sektoren der Festplatte bis hin zum Ausspionieren von Logins und Passwörtern und deren Versand per E-Mail um jemanden anderen den unbefugten Zugriff z.B. auf Computer oder auch irgendwelche Shopping-Accounts oder sogar Bankkonten zu ermöglichen.

Virentypen:

- *Bootsektor-Viren (BSV):* Einer der ersten Virentypen, verändern meist den Master Boot Record (MBR). Infiziert wird der Computer durch ein bootfähiges Medium, welches bereits einen infizierten MBR enthält. Dafür kommt eigentlich nur eine CD od. Floppydisk in Frage. Wird nun mit dem infizierten Medium gebootet, so modifiziert der Virus den MBR der Festplatte und wird somit mit jedem Systemstart automatisch wieder gestartet, und zwar vor dem Start des Betriebssystems. Damit sind auch Schutzmechanismen des OS oder von Virenscanner nicht mehr wirksam. Da FD's kaum mehr eingesetzt werden, hat dieser Virustyp kaum mehr Relevanz, obwohl auch bootfähige CDs prinzipiell MBR-Viren enthalten können.
- *Stealth-Viren:* Viren die sich auf irgendeine Art tarnen, um nicht erkannt zu werden. Z.B. liefert der MBR-Virus beim Auslesen des MBR durch einen Virenscanner den kopierten Original-MBR, oder der Beagle-Virus überträgt sich selbst nur verschlüsselt.
- *Dateiviren:* Infiziert vorzugsweise Programmdateien. Wird das infizierte Programm gestartet, so versucht der Virus, sich im Speicher einzulagern und weiter Programme zu infizieren. Der Virus hängt sich dabei meist an das Ende der Programmdatei und verändert somit die Größe der Datei, woran auch eine Infektion erkennbar ist. Eine Entfernung solcher Viren ist nur möglich, indem man das infizierte System mit einem virenfreien Medium startet und alle infizierten Programme neu installiert.
- *Makroviren:* Sind im Prinzip auch Dateiviren, befallen aber Dokumentdateien die mit Programmen bearbeitet werden, welche die Abarbeitung von Makros zulassen wie z.B. M$-Word oder M$-Excel. Diese Programme lassen in ihren Makros auch Autostart-Funktionen zu, d.h. dass beim Öffnen eines Dokuments, das ein Makro enthält, dieses autom. gestartet wird und somit auch dem Virus die Möglichkeit bietet, sich weiter zu verbreiten. Um sich zu tarnen, entfernen solche Makroviren z.B. Menüpunkte im Bearbeitungsprogramm, die das Darstellen des Makros ermöglichen würden oder das Ausführen der Makros deaktivieren würden. Mögliche Schutzmaßnahmen:
  - Einschalten aller Warnungen, die das Anwendungsprogramm vor Makroviren ausgeben könnte.
  - Virenscanner, der auch Makroviren erkennen kann.
  - Verwendung reiner Dokumentviewer für verdächtige Dokument.

Achtung insbesondere vor CDs, die irgendwelchen Computer- od. sonstigen Magazinen beigelegt sind. Diese können natürlich auch prinzipiell alle Arten der oben genannten Viren enthalten.

1.4.2 Würmer

Würmer benötigen keine Wirtprogramme zur Verbreitung sondern verbreiten sich eigenständig. Sie müssen dafür nur einmal vom Benutzer gestartet werden. Der typische Verbreitungsweg ist das E-Mail System. Dazu enthält der Wurm einen eigenen SMTP-Server, die Adressen seiner nächsten Opfer holt sich der Wurm aus der Adressverwaltung des E-Mail Clients. Würmer im Internet sind mittlerweile zahlenmäßig das größere Problem als Viren.

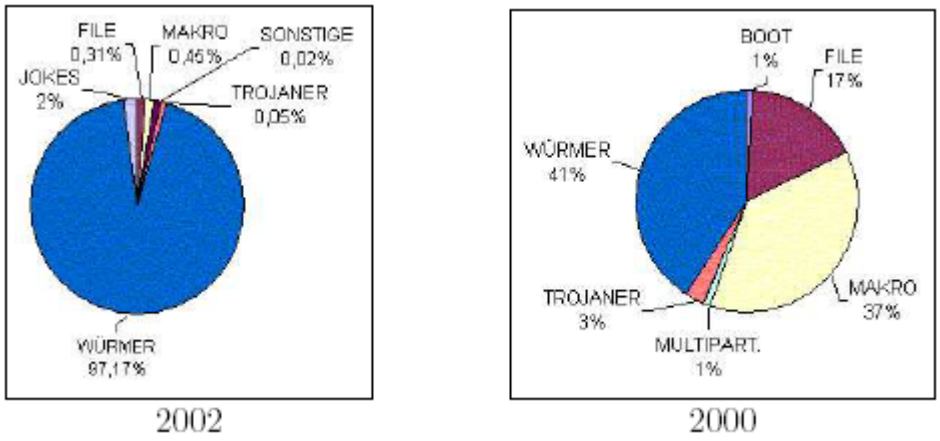


Abb. 1.14: Häufigkeitsverteilung von Viren u. Würmern im Jahr 2002 und 2000 laut BSI. Quelle: www.bsi.de/av/virenstatistik/vaus.htm

1.4.3 Trojanische Pferde

Sind Programme, die neben ihrer eigentlichen Funktion noch weitere, dem Benutzer nicht bekannte Funktionen ausführen. Ein Trojanisches Pferd kann praktisch in jedem Programm stecken. Auch im OS oder der Systemsoftware können unbekannte nicht dokumentierte Funktionen enthalten sein.
Da eine automatische Entdeckung von trojanischen Pferden nicht möglich ist (nicht einmal das Halteproblem ist theoret. entscheidbar), kann der Benutzer nur versuchen, Veränderungen am System zu erkennen. z.B. mit Hilfe einer Datenbank in der für jedes installierte Programm

- das Datum der Änderung
- die Größe der Programmdatei
- und spez. Hasch-Codes der Programmdatei

festgehalten und regelmäßig gegengeprüft werden.

Trojanische Pferde können auch in Dokumentdateien enthalten sein, siehe nur die ganzen zusätzlichen Informationen, die in M$-Word Dateien abgespeichert werden.

1.4.4 Passwortmissbrauch

Mögliche Prüfungen der Authentizität eines Benutzers:

- Biometrische Merkmale
- Kontrolle eines schwer zu fälschenden Gegenstandes (Ausweis)
- Überprüfung eines speziellen Wissens (z.B. Geheimcode)

Gegenüber dem Computer erfolgt derzeit die Authentifizierung des Benutzers hauptsächlich über die Prüfung von spez. Wissen, nämlich dem Passwort. Mögliche Wege, ein fremdes Passwort zu erfahren sind

- *Raten:* dafür werden naheliegende Begriffe verwendet wie Namen der Kinder u.ä.
- *Ausprobieren:* dabei werden aber max. x^n Versuche benötigt wenn das Passwort eine Länge von n Zeichen hat und jedes Zeichen aus einem Vorrat von *x* Zeichen stammen kann.
- *Ausspähen/Abhören:*
  - durch Beobachtung bei der Passworteingabe
  - Abhören unverschlüsselter Übertragung des Passworts
  - Phishing – mittels gefälschter Nachrichten zur Preisgabe des PW zu verleiten
  - mittels gefälschter Nachricht auf eine gefälschte Web-Site verlinken, wo dann der Benutzer sein Passwort eingibt (z.B. DNS-Spoofing od. URL-Hacking).

Passwortgenerierung unter UNIX:

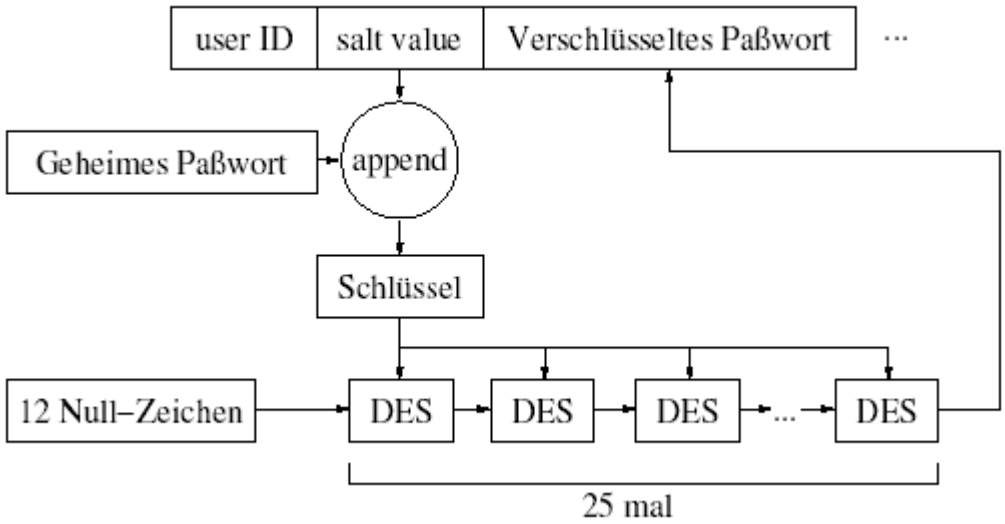


Abb. 1.15: Ablauf beim Speichern eines Passwords unter UNIX.

Der Salt-Value hängt unter anderem von der aktuellen Uhrzeit ab, somit ist sichergestellt, dass für 2 Benutzer mit dem selben Passwort nicht das selbe Verschlüsselungsergebnis erzeugt wird.
Die frei verfügbaren Crack-Programme „*crack*“ und „*john*“ probieren nach diesem Verfahren einfach eine große Menge möglicher u. beliebter Passwörter aus. Die Trefferquote solcher Crack-Programme kann bis zu 25% betragen!

Regeln für die Auswahl von Passwörtern:

- Keine PW verwenden, die in einem erkennbaren Zusammenhang zur eigenen Person bestehen (Namen der Kinder u.ä.)
- Ausreichende Wortlänge (mind. 6 besser 8 Zeichen)
- Keine PW aus einem Wörterbuch. Sonderzeichen in PW einfügen.
- Voreingestellte PW unbedingt ändern.
- Regelmäßiger Wechsel des PW.
- PW niemals notieren.

2. Verschlüsselung und digitale Signaturen

2.1 Einführung

Verschlüsselung – aus einer offenen Nachricht mittels Verschlüsselungsalgorithmus eine verschlüsselte Nachricht erzeugen, die im Idealfall keine Rückschlüsse mehr auf den Inhalt der originalen Nachricht erlaubt.
Entschlüsselung – die umgekehrte Transformation, um aus einer verschlüsselten Nachricht mittels Entschlüsselungsalgorithmus wieder dir Originalnachricht herzustellen.

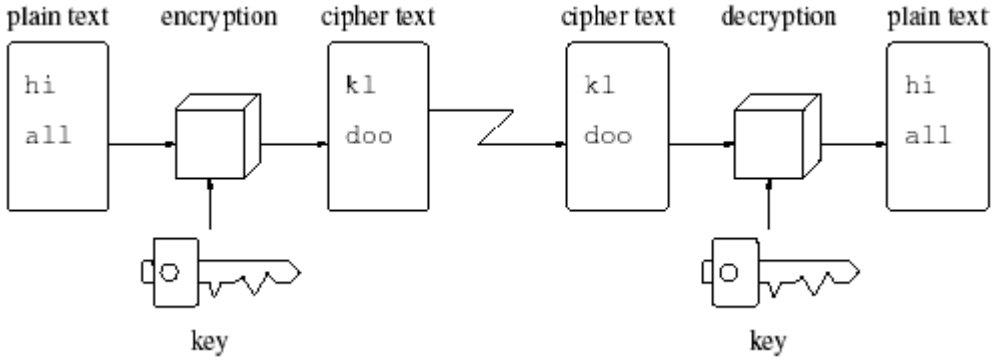


Abb. 2.1: Prinzip der verschlüsselten Übertragung

Bei Übertragung über mehrere Stationen gibt mehrere Einsatzmöglichkeiten:

Leitungsverschlüsselung:

Zwischen zwei Stationen auf dem Übertragungsweg wird die Nachricht verschlüsselt, also der Absender verschlüsselt für den ersten Empfänger, dieser entschlüsselt und verschlüsselt anschließend für den zweiten Empfänger usw.

- *Vorteil:* Immer nur die beiden direkten Kommunikationsnachbarn müssen sich auf einen Schlüssel einigen, daher kann die Methode auf den niedrigen Ebenen des Protokollstacks eingesetzt werden.
- *Nachteil:* Alle Stationen auf dem Übertragungsweg müssen sicher und vertrauenswürdig sein.

Ende-zu-Ende-Verschlüsselung (bevorzugte Methode):

Nur der Absender verschlüsselt die Nachricht, alle weiteren Beteiligten Stationen leiten die verschlüsselte Nachricht weiter und die eigentliche Zielstation entschlüsselt die Nachricht.

- *Vorteil:* keiner der Stationen auf dem Übertragungsweg kann die Nachricht im Klartext lesen.
- *Nachteil:* Absender muss sich mit jedem möglichen Empfänger auf Verschlüsselungsverfahren einigen.

Kriterien zur Klassifizierung von Verschlüsselungsverfahren sind

Verschlüsselungsoperationen:

Grundoperationen (und deren Kombinationen), die ein Verschlüsselungsalgorithmus zur Verschlüsselung verwendet:

- ***Ersetzung (substitution):*** Jedes Zeichen (Zeichengruppe) eines Klartextes wird auf ein best. Zeichen (Zeichengruppe) des verschlüsselten Textes abgebildet. z.B. ersetze jedes A durch B, usw.
- ***Umordnung (transposition):*** Die Zeichen des Klartextes werden neu angeordnet. z.B. Text von hinten nach vorne aufschreiben.

Anzahl der Schlüssel:

- ***Ein Schlüssel:*** Für Verschlüsselung und Entschlüsselung wird der selbe Schlüssel verwendet -> ***symmetrische Verschlüsselung*** od. ***private key*** Verfahren (od. auch secret key).
- ***Mehrere Schlüssel:*** Verschiedene Schlüssel für Verschlüsselung und Entschlüsselung -> ***asymmetrische Verschlüsselung***. I.d.R. kann der Schlüssel zur Verschlüsselung problemlos öffentlich bekannt sein -> ***public key*** Verfahren. Der *private key* zur Entschlüsselung ist nicht oder nur mit sehr hohem Aufwand aus dem *public key* generierbar.

Private key Verfahren i.A. schneller zu berechnen als public key Verfahren. Vorteilhaft ist deshalb eine Kombination der beiden Verfahren – ***hybride Verschlüsselungsverfahren*** – um einen geheimen Schlüssel mit einem public key Verfahren für den Schlüsselaustausch zu verschlüsseln.

Verarbeitung des Klartextes:

Klartext kann normalerweise nicht als Ganzes verschlüsselt werden da die Länge nicht vorab bekannt ist.

- ***Blockverschlüsselung:*** Klartext wird in Blöcke fester Größe eingeteilt (64 od. 128 Bit) und jeder Block wird getrennt verschlüsselt. Ist Länge des Texts nicht ein Vielfaches der Blockgröße, so wird der letzte Block mit einem best. Zeichen bis zur Blockgröße aufgefüllt.
- ***Stromverschlüsselung:*** Jedes ankommende Klartextzeichen wird für sich verschlüsselt -> Online-Algorithmus.
    - *Vorteil*: Kein Auffüllen von Rumpfblöcken, es werden also nur Nutzdaten übertragen.
    - *Nachteil*: Online-Algorithmen können ohne Kenntnis der (Daten-)Zukunft nicht immer optimale Ergebnisse berechnen.

2.2 Hardware- und Betriebsystemsicherheit

Hardware:

Sicherheit eines Computers hängt unter anderem auch von der Sicherheit seiner Umgebung ab, also die Sicherheit

- des Gebäudes selbst
- und der Versorgungsleitungen.

Weiters der darin befindlichen Räume wie

- Büroräume
- Serverräume u.
- Datenarchive

Für den Computer selbst ist auch die Sicherheit des Gehäuses zu betrachten (Versperrbarkeit, Schutz vor Wasser, u.ä.).

Betriebssystem:

Wichtigste Vorgaben sind hierbei das Vorhandensein von

- Benutzerkennungen inkl. Passwort
- Rollenvergabe für verschiedene Benutzergruppen
- Zugriffsrechte für Dateien und Programme
- Schutz transienter Objekte wie z.B. Hauptspeicherbereiche

- kein direkter Zugriff auf Hardware für Anwenderprogramme, die im Userspace laufen. Solche Programme sollten nur im Admin-Mode installierbar sein.

2.3 Private Key Verschlüsselung

2.3.1 Prinzip der Private Key Verschlüsselung

Ist ein symmetrisches Verfahren, bei dem einmalig ein geheimer Schlüssel zwischen den beiden Kommunikationspartnern ausgetauscht werden muss.

2.3.2 Klassische Verschlüsselungsalgorithmen

Cäsar Chiffre:

Einfache Substitutionschiffre wobei jedes Klartextzeichen um 3 Zeichen verschoben wird, also ‚a' wird zu ‚d', usw. Am Ende des Alphabets beginnt die Ersetzung wieder von vorne, also z.B. ‚x' -> ‚a', usw. Falls ‚a'=0, ‚b'=1 usw. dann lautet die Berechnung

$$Chiffre(x) = (x + 3) \bmod 26$$

Die Verschiebedistanz (bei Cäsar = 3) ist der Verschlüsselungsschlüssel k_v. Den Entschlüsselungsschlüssel k_e kann man aus der Zahl der zur Verfügung stehenden Zeichen n und dem Verschlüsselungsschlüssel k_v berechnen:

$$k_e = n - k_v$$

ROT13 ist ein heute noch eingesetzter Cäsar Chiffre der in Newsgroups gerne verwendet wird. k_v ist dabei 13.

Eine allgemeine Formel für die Verschlüsselung von ASCII-Zeichen lautet also

$$Chiffre_k(x) = (x + k) \bmod 256$$

Sicherheit bietet dieses Verfahren allerdings keine, da rel. wenig Versuche (26 bzw. 256) notwendig sind, um alle möglichen Verschiebungen auszuprobieren. Außerdem kann auch über eine Häufigkeitsanalyse der Zeichen unter Kenntnis der verwendeten Sprache auf die Originalzeichen rückschließen (im Deutschen z.B. ist ‚e' der häufigste Buchstabe).

Monoalphabetische Chiffre:

Ist eine Verbesserung des Cäsar Chiffre wobei die Reihenfolge der Ersetzungszeichen nicht mehr vorgeschrieben ist. Der Schlüssel ist dabei die 2. Zeile der Ersetzungstabelle. Z.B.:

```
Klartextzeichen: a b c d e f g h i j k l m n o p q r s t u v w x y z
Chiffrezeichen:  x e o i h u j l m k n f p a r c t g v w d y z q b s
```

Hier kann man ebenfalls wieder mit Häufigkeitsanalysen arbeiten, um den Geheimtext zu entschlüsseln, allerdings muss man dabei alle Buchstaben betrachten.

Polyalphabetische Chiffre:

Eine weiter Verbesserung erreicht man, wenn man für ein Klartextzeichen nicht immer dasselbe Chiffrezeichen zuweist. Klassisches Bsp. ist die ***Vigenère Chiffre***. Es wird zwar wieder für jedes Zeichen eine Verschiebung angewandt, allerdings ist die Verschiebungslänge

von einem Schlüsselwort abhängig, wobei jedes Zeichen des Schlüsselwortes eine best. Verschiebungslänge repräsentiert. Bsp. mit Schlüsselwort „*secret*“:

Klartext	d	a	s	i	s	t	g	e	h	e	i	m
Schlüssel	s	e	c	r	e	t	s	e	c	r	e	t
Länge	18	4	2	17	4	19	18	4	2	17	4	19
Chiffre	v	e	u	z	w	m	y	i	j	v	m	f

Ein Angriff auf diese Verschlüsselung muss zuerst die Länge des Schlüssels ermitteln und anschließend die einzelnen Verschiebungslängen.

Transpositionsverschlüsselung:

Die Klartextzeichen bleiben erhalten, allerdings wird ihre Reihenfolge nach best. Permutationsregeln vertauscht (z.B. (1 3)(2 4) also immer das 1. <-> 3. , 2. <-> 4. Zeichen). Diese Permutation wird dann sukzessive auf alle Zeichengruppen angewandt (also für obiges Bsp. zuerst die 1. vier Zeichen, dann die 2. vier Zeichen, usw.). Es sind im Prinzip beliebige Permutationen durchführbar mit beliebiger Länge. Der Schlüssel ist somit die angewandte Permutation. Als Hilfsmittel für die Verschlüsselung kann eine Matrix verwendet werden die zeilenweise beschrieben und dann spaltenweise ausgelesen wird.
Die im 2. Weltkrieg benutzte Verschlüsselungsmaschine *Enigma* basierte auf diesem Verfahren.

2.3.3 Moderne Verschlüsselungsalgorithmen

Grundoperationen:

Moderne Verfahren benutzen ebenfalls Ersetzungs- und Vertauschungstechniken. Die Ersetzungsfunktion wird meistens durch eine XOR Operation realisiert. Die XOR Funktion hat die vorteilhafte Eigenschaft, dass

$$((a \oplus b) \oplus b) = a \text{ gilt.}$$

Die Realisierung der Vertauschung erfolgt meist durch das HW-unterstützte zirkuläre Verschieben realisiert.

Das Feistel-Verfahren:

(Nach H. Feistel) Die Grundidee ist, die Operationen Ersetzen und Vertauschen mehrfach hintereinander zu benutzen.
Prinzip: Klartextwort der Länge g wird in ein linkes L_0 u. rechtes Teilwort R_0 der Länge $g/2$ aufgeteilt -> aus dem Schlüssel S wird ein erster Teilschlüssel S_0 berechnet, der eine Funktion F steuert, welche auf R_0 angewandt wird -> das Ergebnis R_0' und L_0 werden über XOR verknüpft -> Ergebnis aus XOR und R_0 werden vertauscht. Dieser Vorgang (Runde) wird mehrmals hintereinander ausgeführt, wobei für jede Runde i ein Teilschlüssel S_{i-1} generiert wird. Nach der letzten Runde werden R_n und L_n noch vertauscht.
Die Entschlüsselung erfolgt nach dem selben Prinzip, die Teilschlüssel werden aber in der umgekehrten Reihenfolge benutzt. Die in Abb. 2.2 gezeigten Runden werden also von unten nach oben durchlaufen. Wegen der Tatsache das F eine Funktion ist (jede Eingabe wird eindeutig auf eine Ausgabe abgebildet) und da das XOR nach 2-maliger Anwendung wieder die Eingabe liefert, erhält man am Ende wieder den Klartext.

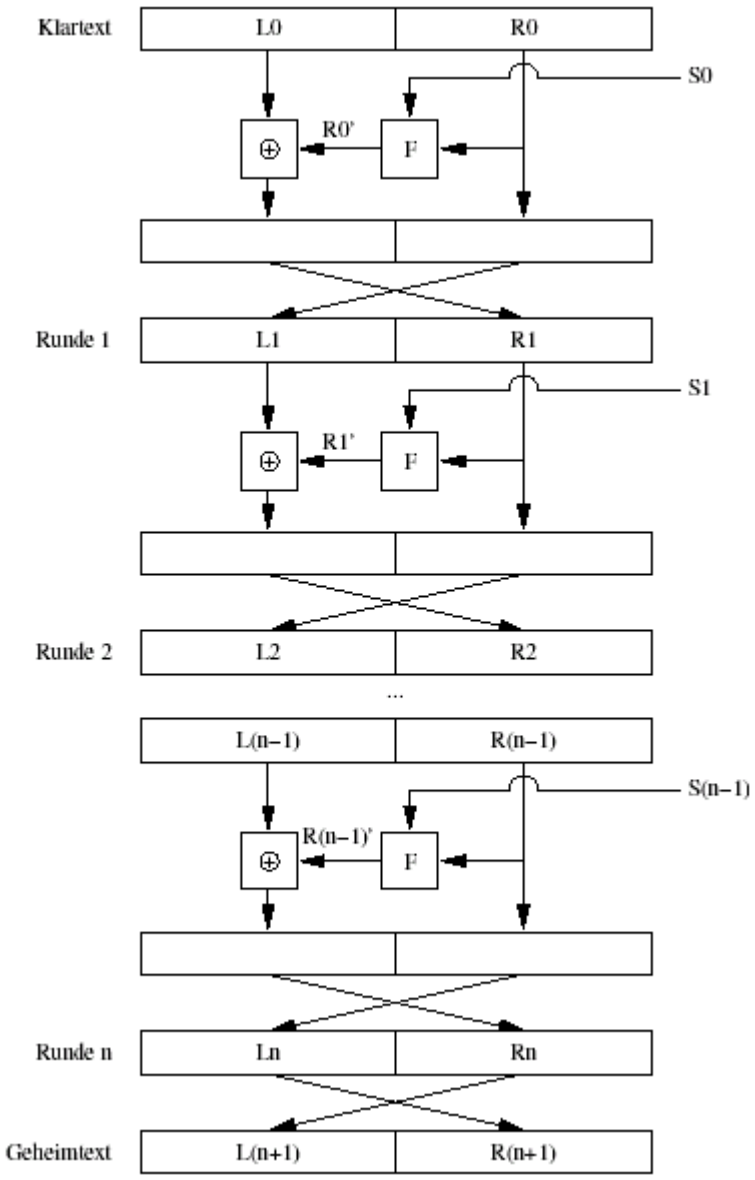


Abb. 2.2: Verschlüsselungsprinzip von Feistel

Die Parameter des Feistel-Verfahrens:

1. Die Funktion F. Diese sollte nicht zu einfach sein um eine erfolgreiche Krypto-Analyse möglichst zu verhindern.
2. Die Art u. Weise, wie die Teilschlüssel berechnet werden.
3. Die Anzahl der Runden.
4. Die Blockgröße g. Große Blöcke erschweren die statistische Analysen.
5. Die Länge des Schlüssels.

DES – Data Encryption Standard:

Ist im wesentlichen ein Feistel-Verfahren mit folgenden Parametern :

Funktion F: Kombination aus XOR und festen Ersetzungsboxen (S-BOX)
Teilschlüssel: Generierung von 48 Bit Teilschlüssel durch Shift und Permutation.
Runden: 16
Blockgröße: 64 Bit
Schlüssellänge: 56 Bit

Schwachpunkt des DES ist die rel. kleine Schlüssellänge von 56 Bit, der bereits 1999 in 22,25 Stunden (mit Parallelrechnern und einer Spezialmaschine „deep crack“) geknackt werden konnte. Eine mögliche Verbesserung ist die Verwendung mehrerer verschiedener Schlüssel um so den Suchraum für einen Angriff zu vergrößern. Heute wird meist *triple DES* (*3DES*) benutzt.

IDEA – International Data Encryption Standard:

Nachfolger von *DES*, aber mit größerer Schlüssellänge. Wird auch für *PGP* eingesetzt.

Funktion F: Kombination aus XOR, Ganzzahladdition mod 2^{16} und Ganzzahlmultiplikation mod $2^{16}+1$.
Teilschlüssel: Extraktion von 52 Teilschlüssel mit je 16 Bit aus Gesamtschlüssel durch Auswahl u. zirkuläre Verschiebung.
Runden: 8
Blockgröße: 64 Bit
Schlüssellänge: 128 Bit

IDEA-Algorithmus wurde von Anfang offen gelegt um Analyse von Schwachstellen zu ermöglichen.

Blowfish:

(von Bruce Schneider) Basierend auf dem Feistel-Verfahren werden in jeder Runde beide Hälften des Wortes verändert. Die Verschlüsselung findet in 2 Phasen statt, wobei zuerst der Teilschlüssel und der Inhalt von 4 S-Boxen berechnet wird und anschließend die eigentliche Verschlüsselung stattfindet.

Funktion F: Kombination aus XOR, Ganzzahladdition mod 2^{32} und S-Boxen.
Teilschlüssel: 18 Teilschlüssel mit je 32 Bit werden ausgehend von Initialwerten durch Verknüpfung mit dem Schlüssel und Anwendung des Algorithmus auf sich selbst generiert.
Runden: 16
Blockgröße: 64 Bit
Schlüssellänge: 32 ... 448 Bit (1...14 Wörter der Länge 32 Bit)

Die variable Schlüssellänge ermöglicht die Auswahl zwischen „sichere Verschlüsselung“ und „schnelle Berechnung“. Die große Schlüssellänge und die aufwändige Initialisierungsphase machen das Ausprobieren aller Schlüssel sehr schwierig. ***Twofish*** ist der Nachfolge-Algorithmus von *Blowfish*.

RC5:

(von Ron Rivest) RC5 ist eine Familie von Algorithmen und beruht nicht auf dem Feistel-Verfahren.

Prinzip: Klartextwort wird in 2 Hälften L_0 und R_0 zerlegt. Zuerst wird zu L_0 der Teilschlüssel S_0 und zu R_0 der Teilschlüssel S_1 addiert (je nach Blocklänge mod. 2^{16}, 2^{32} od. 2^{64}). Dann werden die linke Hälfte und R_0' XOR verknüpft und das Resultat nach links im Kreis verschoben, und zwar um den Wert R_0'. Anschließend wird S_2 zum linken Teilwort addiert und es finden die gleichen Operationen auf dem rechten Teilwort statt. Zur Entschlüsselung müssen die Operation + und <<< durch – und >>> ersetzt werden.

Funktion F: siehe Abb. 2.3
Teilschlüssel: Verknüpfung des Schlüssels mit e = 2,718... und ϕ = 1,618... (goldener Schnitt)

Runden: 0...255
Blockgröße: 32, 64 od. 128 Bit
Schlüssellänge: 0 ... 255 Bit

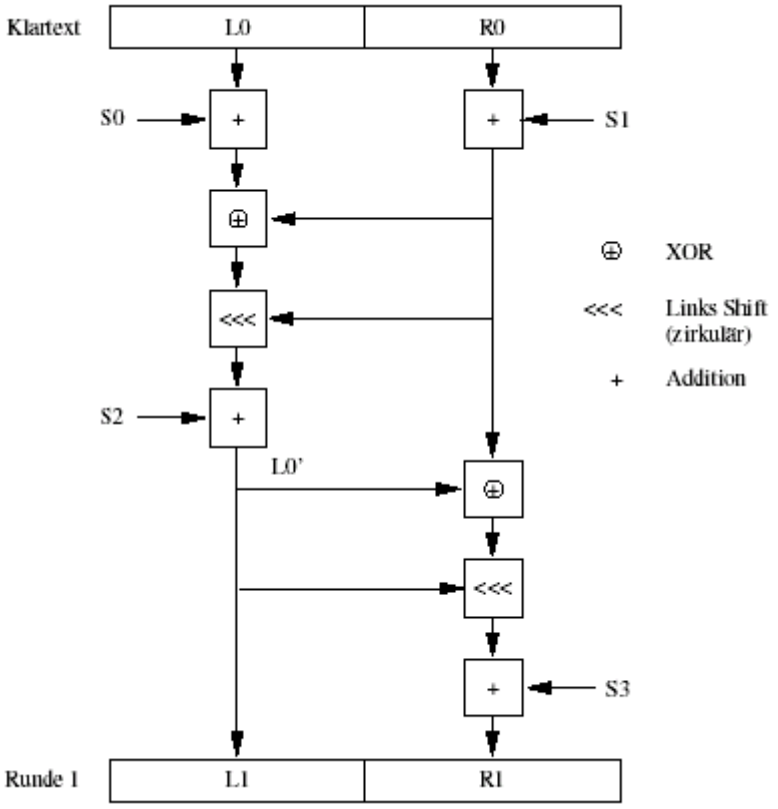


Abb. 2.3 Eine Runde bei RC5

CAST-128:
(von Calisle Adams u. Stafford Tavares) Es ist ein klassischer Feistel-Algorithmus.
Funktion F: Kombination aus einfacher Arithmetik (+, -, ⊕), zirkulären Shifts und fest definierten S-Boxen.
Teilschlüssel: Aus dem Schlüssel mit Hilfe spez. S-Boxen generiert.
Runden: 16
Blockgröße: 64 Bit
Schlüssellänge: 128 Bit

2.3.4 AES – Der Advanced Encryption Standard

siehe auch http://csrc.nist.gov/encryption/aes
Es wurde von NIST (National Institute of Standards and Technology USA) und Kryptoexperten eine Variante des RIJNDAEL-Algorithmus von Daemen u. Rijmen als AES ausgewählt.
Prinzip: Die Blockgröße ist nicht fix und kann entweder 128, 196 oder 256 Bit betragen. Die selben Blockgrößen gelten für den Schlüssel und ist damit mind. 128 Bit. Die Anzahl der Verschlüsselungsrunden ist abhängig von der Schlüssellänge und der Blockgröße und beträgt zwischen 10 u. 14 Runden. Am Anfang wird der Block mit dem Rundenschlüssel XOR verknüpft u. die letzte Runde ist ebenfalls abweichend von den übrigen.
Einen Block kann man sich als Matrix vorstellen, auf der in jeder Runde mehrere Operationen durchgeführt werden:

linear matrix layer: sorgt dafür, dass eine große Diffusion der Werte über mehrere Runden stattfindet.
non linear layer: hier finden nicht lineare Ersetzungen statt, die durch S-Boxen realisiert sind.
key addition layer: hier wird ein Rundenschlüssel zum Block addiert (XOR).

Die Matrix hat 4 Zeilen und eine von der Blockgröße abhängige Anzahl von Spalten. Jedes Matrixelement entspricht einem Byte des Blocks.
Als Erstes findet in jeder Runde eine Ersetzung statt (*ByteSub Transformation*):

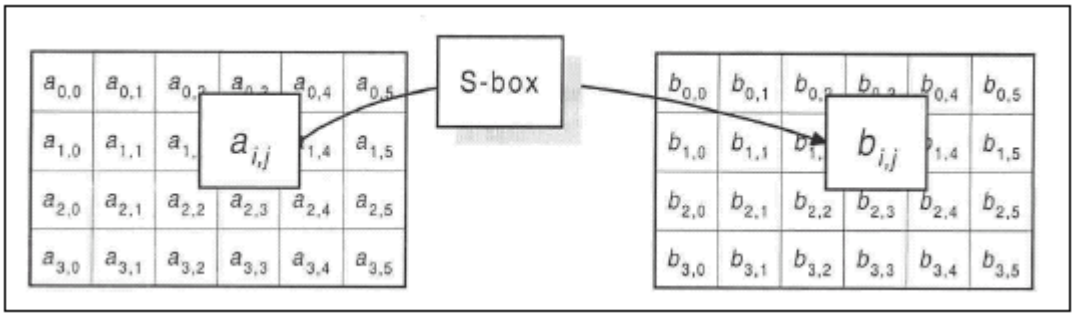


Abb. 2.4: AES byte substitution

Diese Funktion ist umkehrbar, dahinter steht letztlich ein Multiplikation eines Vektors mit einer Matrix und eine Vektoraddition. Im nächsten Schritt wird jede Zeile im Kreis um eine fest definierte Verschiebedistanz verschoben (*ShiftRow Transformation*), welche nur von Schlüssellänge und Blockgröße anhängt.

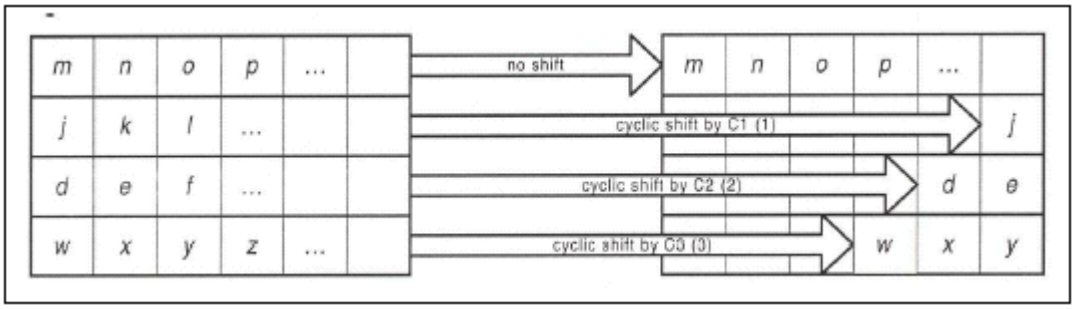


Abb. 2.5: AES shift rows

Im dritten Schritt werden die Spalten der Matrix durcheinander gebracht. Dabei wird jede Spalte mit einer 4x4 Matrix *c(x)* multipliziert. c(x) wird so gewählt, das die Umkehrung dieser Operation möglich ist.

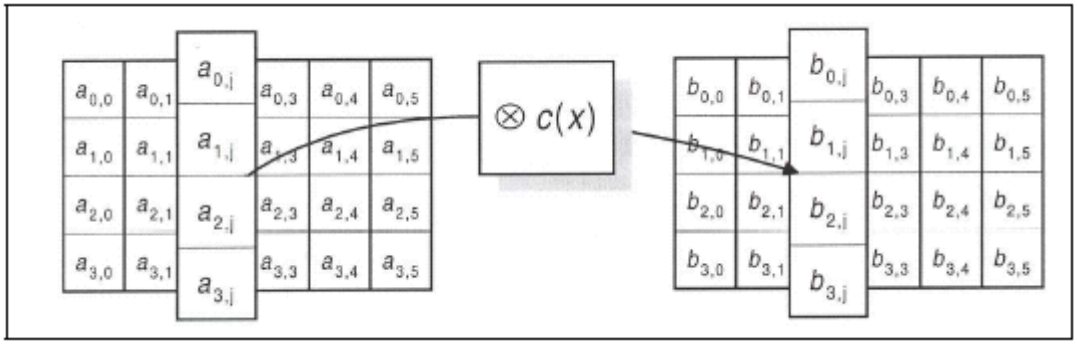


Abb. 2.6: AES mix column

Am Ende einer Runde wird aus dem Schlüssel ein spez. Rundenschlüssel berechnet, der immer die selbe Größe wie ein Block hat. Die Blockbytes werden mit dem Rundenschlüssel XOR verknüpft.

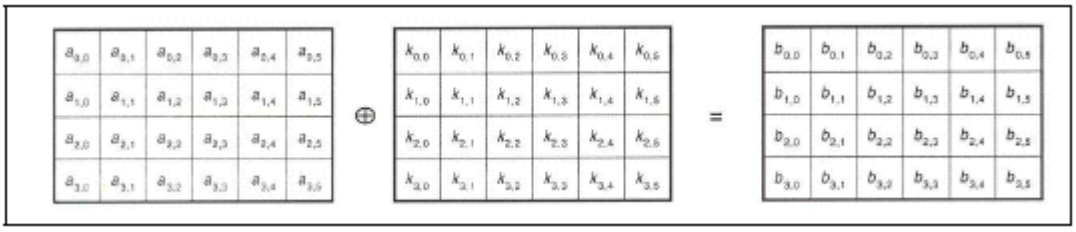


Abb. 2.7: AES add round key

Zur Berechnung des Rundenschlüssels wird der eigentliche Schlüssel expandiert und in einem Array abgelegt. Aus diesem Array werden dann die Rundenschlüssel ausgelesen.

Der Ausgabeblock einer Runde ist der Eingabeblock der nächsten. Da alle Operationen umkehrbar sind ist auch eine Entschlüsselung möglich.

2.3.5 Verschlüsselungsmodi

Jedes bisher beschrieben Verfahren ist deterministisch, also aus dem selben Klartext wird mit demselben Schlüssel immer der gleiche Geheimtextblock erzeugt (***electronic code book ECB***). In diesem Modus kann ein Angreifer ev. Nachrichten wiedererkennen und wiederholt einspielen (***replay Angriff***). Damit könnte ein Aufforderung zu einem automatisierten Handeln versendet werden, deren Auswirkungen sich ev. nur schwer wieder rückgängig machen lassen.
Eine Verbesserung bietet das ***cipher block chaining CBC***, bei dem jeder Klartextblock vor seiner Verschlüsselung mit dem vorhergehenden Geheimtextblock verknüpft wird.

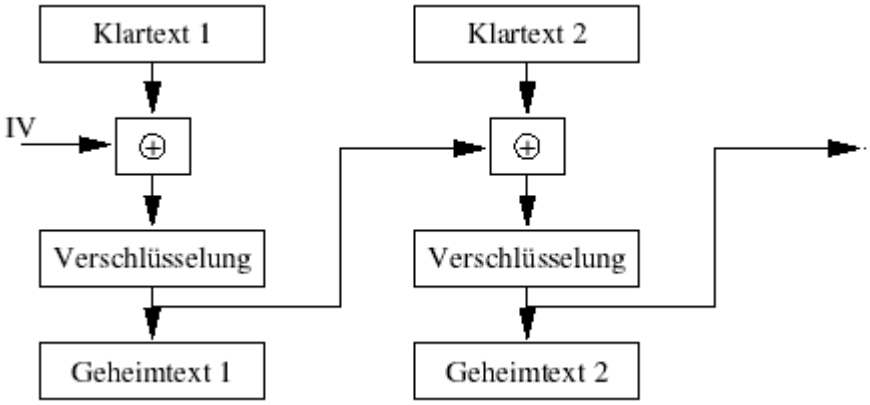


Abb. 2.8: Cipher Block Chaining

IV ... Initialisierungsvektor zur Verknüpfung mit erstem Block.

CBC ist zwar auch deterministisch, da der komplette Klartext wieder immer in denselben Geheimtext verschlüsselt wird, ein statistischer Angriff auf einzelne Blöcke ist aber nicht mehr möglich.
Neben dem CBC gibt es aber noch andere Modi, die Angriffe auf einzelne Blöcke verhindern, wie z.B. Cipher Feedback CFB oder Output Feedback OFB.

2.4 Public Key Verschlüsselung

2.4.1 Prinzip der Public Key Verschlüsselung

Wird auch als asymmetrisches Verfahren bezeichnet Jeder Teilnehmer benötigt zwei zusammengehörige Schlüssel, eine ***öffentlichen Schlüssel*** (***public key***), mit dem Nachrichten an ihn verschlüsselt werden können. Weiters einen ***privaten Schlüssel*** (***private key***), den nur er besitzt und mit dem an ihn gesendete Nachrichten, die mit seinem ***public key*** verschlüsselt wurden, wieder entschlüsselt werden können.

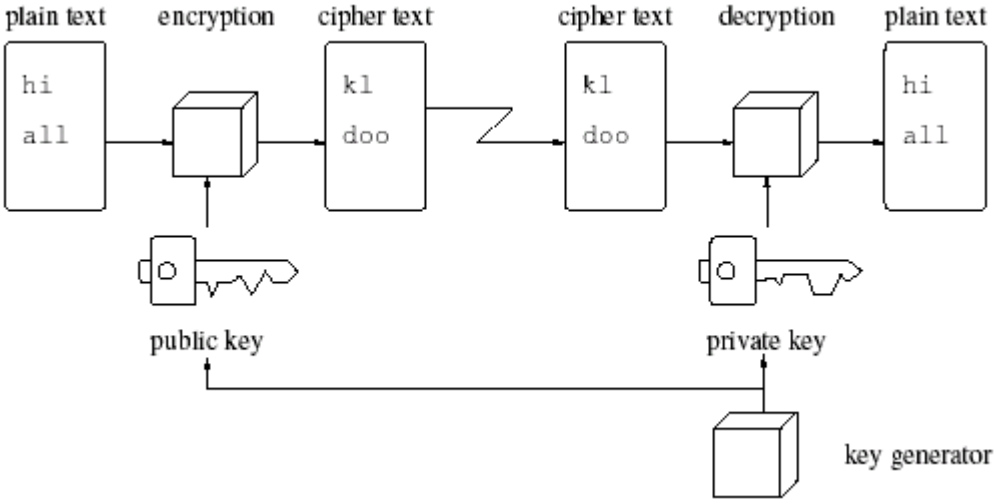


Abb. 2.9: Ablauf der Nachrichtenübertragung mit Public Key Verschlüsselung

Bedingungen:

- Aus dem *public key* kann man den *private key* nicht ableiten bzw. dessen Berechnung ist praktisch nicht in sinnvoller Zeit (Lebensdauer des Schlüssels bzw. der Vertraulichkeit) durchführbar.
- Alleine mithilfe des *public key* kann eine Nachricht nicht entschlüsselt werden.

Bei jedem Public Key Algorithmus gilt für eine Nachricht *m*

$$DeCrypt_d(Crypt_e(m)) = m$$

mit *e* für öffentlichen und *d* für privaten Schlüssel. Wenn die Menge der möglichen Klartexte und Geheimtexte gleich sind dann gilt auch

$$Crypt_e(DeCrypt_d(m)) = m$$

und man kann mit dem Algorithmus auch eine digitale Unterschrift leisten.

2.4.2 Verschlüsselungsalgorithmen

RSA:

(von Rivest, Shamir u. Adleman) RSA kann zur Verschlüsselung und Erzeugung digitaler Signaturen benutzt werden (es muss also obig genannte Bedingung gelten!). Die Sicherheit von RSA beruht darauf, dass es schwer ist, große Zahlen in ihre Primfaktoren zu zerlegen (Faktorisieren).

Prinzip: Für die Erzeugung eines Schlüsselpaares werden 2 große Primzahlen *p* und *q* gewählt und $n = p \cdot q$ berechnet. Für jede Zahl $m \le n$ gilt dann

$$m^{k(p-1)(q-1)+1} \bmod n = m$$

für ein beliebiges *k*.

Nun wählt man eine natürliche Zahl e, die teilerfremd zu *(p-1)(q-1)* ist. Der öffentliche Schlüssel besteht dann aus *e* und *n*. Der geheime Schlüssel *d* wird so berechnet, dass $e \cdot d \bmod (p-1)(q-1) = 1$ gilt, woraus folgt dass $e \cdot d = k(p-1)(q-1)+1$ für ein *k* gilt.

<u>*Verschlüsselung*</u>: Die Nachricht *m* wird mit dem öffentlichen Schlüssel verschlüsselt

$$Crypt_e(m) = m^e \bmod n$$

<u>*Entschlüsselung*</u>: Mit $c = Crypt_e(m)$ und $DeCrypt_d(c) = c^d \bmod n$ folgt insgesamt

$$\begin{aligned} DeCrypt_d(Crypt_e(m)) &= Crypt_e(m)^d \bmod n \\ &= (m^e)^d \bmod n \\ &= m^{e \cdot d} \bmod n \\ &= m^{k(p-1)(q-1)+1} \bmod n \\ &= m \end{aligned}$$

Mit RSA können nur Nachrichten verschlüsselt werden, die kleiner *n* sind. Größere Nachrichten müssen vorher in passende Blöcke zerlegt werden. Wegen der Kommutativität der Multiplikation können Nachricht auch mit dem geheimen Schlüssel verschlüsselt und mit dem öffentlichen Schlüssel wieder entschlüsselt werden (in obiger Ableitung einfach *e* mit *d* vertauschen). Damit ist RSA auch für digitale Unterschriften einsetzbar.
Die Sicherheit von RSA beruht darauf, dass aus dem öffentlichen Schlüssel nicht auf den privaten Schlüssel geschlossen werden kann. Alle bekannten Angriffsverfahren versuchen, n in seine Primfaktoren p und q zu zerlegen (Faktorisierung). Der Aufwand dafür steigt mit der Größe von n. Derzeit sind Schlüssellängen von 1024 Bit bereits die untere Grenze in Punkto Sicherheit.
Da die RSA Verschlüsselung wegen der Multiplikation sehr großer Zahlen sehr zeitaufwändig ist (ca. Faktor 100 langsamer als DES), wird sie hauptsächlich dafür verwendet, den geheimen Schlüssel für die Session (***session key***) eines symmetrischen Verfahrens zu übertragen. Damit ein Angreifer den session key beim Abhören von verschlüsselten Nachrichten nicht herausfinden kann, sollte der session key

- groß genug sein, so dass ein simpler Vergleich mit allen bereits mitgehörten Nachrichten zu aufwändig wäre
- von guten Zufallsgeneratoren mit echten Zufallsquellen zur Initialisierung abhängen.

man in the middle attack:
Ein Angreifer schaltet sich zw. Sender u. Empfänger:

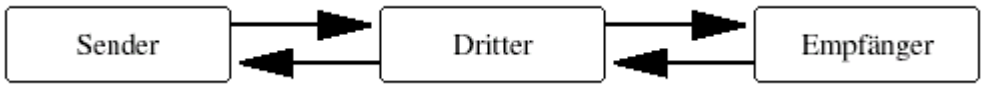


Abb. 2.10: „Man in the middle" Angriff

<u>*Prinzip*</u>:

1. Der Sender schickt eine unverschlüsselte Nachricht inkl. seines *public key* an den Empfänger und bittet um Übersendung des *public key* des Empfängers.
2. Der Dritte fängt Nachricht ab und generiert 2 Schlüsselpaare P_S,G_S für den Sender und P_E,G_E für den Empfänger und sendet beiden den entsprechenden *public key*.
3. Der Sender und der Empfänger halten nun P_S und P_E für den *public key* des anderen und verschlüsseln damit ihre versendeten Nachrichten.
4. Der Dritte kann nun diese Nachrichten entschlüsseln und ev. verändert und wieder verschlüsselt weiterleiten. Er fungiert quasi als „Relaisstation".
5. Sender und Empfänger können mit ihren *private keys* die Nachrichten entschlüsseln und bemerken nichts von den gefälschten *public keys*.

Einen überprüfbaren Zusammenhang zwischen public keys und dazugehörigen Personen od. Organisationen stellen sog. Trust Center zur Verfügung.

El-Gamal:

Die Sicherheit des Verfahrens von El-Gamal beruht auf der Schwierigkeit der Berechnung von diskreten Logarithmen über endlichen Körpern.

Prinzip: Als endlicher Körper wird Z_p, die Menge aller Zahlen, die als Rest bei der Division durch die Primzahl *p* entstehen, verwendet. Also $Z_p = \{0,1,2,..., p-1\}$. Es gibt weiters stets eine Zahl *g* für die gilt $y = g^x \bmod p$ mit $y \in Z_p$. (*g* ist also *primitv mod p*)
Der Sender wählt eine zufällige Zahl k zw. 0 und p-1, die zu (p-1) rel. prim ist ($_{gg}T(k, p\text{-}1) = 1$), und berechnet aus dem public key $y = g^x \bmod p$ des Empfängers und der Nachricht M die Werte

$$a = g^k \bmod p \quad \text{und} \quad b = y^k \cdot M \bmod p$$

und schickt sie an den Empfänger. Der berechnet nun wieder mod p

$$\frac{b}{a^x} = \frac{y^k \cdot M}{a^x} = \frac{g^{(x \cdot k)} \cdot M}{g^{(x \cdot k)}} = M$$

Diffie-Hellmann:

Ähnliches Verfahren wie El-Gamal, mit dem sich 2 Teilnehmer auf einen geheimen Schlüssel verständigen können (z.B. bei SSL).
Prinzip: Einigung auf eine große Primzahl *p* und eine Zahl *g primitiv mod p*, welche beide öffentlich sein dürfen -> Partei A wählt nun zufällige Zahl a_x und schickt $a_y = g^{a_x} \bmod p$ an Partei B -> B geht mit einer Zahl b_x und b_y analog zu A vor -> A berechnet aus dem empfangenen b_y und a_x den Wert $b_y^{a_x} \bmod p$ -> B geht ebenso vor -> A und B haben damit denselben Wert berechnet da die folgende Umformung (immer mod p) gilt:

$$b_y^{a_x} = (g^{b_x})^{a_x} = g^{(a_x \cdot b_x)} = (g^{a_x})^{b_x} = a_y^{b_x}$$

Jemand, der den Kanal abhört, kennt aber nur g, *p,* a_y *u.* b_y und kann den berechneten Wert nicht rekonstruieren.

2.5 Hash-Funktionen

Idee des Hashing ist, aus den Daten selbst die Adresse für deren Speicherung zu berechnen.

2.5.1 Prinzip von Hash-Funktionen

Allgemein ist eine Hash-Funktion eine Abbildung der Elemente einer großen Ursprungsmenge auf Elemente einer kleineren Zielmenge. Damit ist es möglich, dass verschiedene Elemente der Ursprungsmenge auf die selben Elemente der Zielmenge (Hash-Werte) abgebildet werden -> Kollisionen.
Eigenschaften einer guten Hash-Funktion:

- schnell und einfach zu berechnen
- möglichst gute Streuung auf die Elemente der Zielmenge

In der Kryptographie benutzt man Hash-Funktionen, um die Authentizität einer Nachricht zu prüfen. Dazu wird von einer Nachricht ein Wert *H(M)* (*kryptographische Prüfsumme,*

fingerprint od. *message diges*) fester Länge berechnet und mit der Nachricht *M* mitgeschickt. Der Empfänger kann nun von der Nachricht *M'* selbst den Wert *H(M')* berechnen und ihn mit dem mitgeschickten Wert *H(M)* vergleichen. Bei Übereinstimmung ist Authentizität von *M* gegeben. Es darf aber einem Angreifer nicht möglich sein *H(M)* und *M* gleichzeitig zu verändern.

Bedingungen für eine sicheren Einsatz von Hash-Funktion:

Einwegfunktion: Zu einem gegebenen Hash-Wert *h* ist es praktisch unmöglich eine Nachricht *M* zu finden, mit $H(M) = h$.

Schwache Kollisionsresistenz: Zu einer gegebenen Nachricht M_1 ist es praktisch unmöglich eine Nachricht M_2 ungleich M_1 zu finden, mit $H(M_1) = H(M_2)$.

Starke Kollisionsresistenz: Es ist praktisch unmöglich 2 verschieden Nachrichten M_1 und M_2 zu finden, mit $H(M_1) = H(M_2)$.

Damit ergeben sich als konkrete Anforderungen dass

- die Zielmenge ausreichend groß sein muss -> gute Hash-Funktionen haben einen Wertebereich von mind. 128 Bit Breite (also 0 bis 2^{128}-1)
- alle Bits der Nachricht M in die Hash-Berechnung eingehen
- die Hash-Funktion auch dann keine Kollisionen erzeugt, wenn ein Angreifer eine große Menge von Nachrichten erzeugt (siehe auch „Geburtstagsattacke“ bzw. Geburtstagsparadoxon)

2.5.2 Hash-Algorithmen

MD5:

(von Rivest) Verarbeitet Nachrichten bis zu einer max. Länge von ca. 2^{64} Bytes und erzeugt 128 Bit lange Hash-Werte.

Prinzip: Dazu wird die Nachricht in 512 Bit Blöcke unterteilt und der letzte Block gegebenenfalls nach einem bestimmten Schema auf 512 Bit aufgefüllt. Danach wird ein 128Bit Puffer mit fest vorgegebenen Werten initialisiert und darauf eine Komprimierungsfunktion F ausgeführt. F wird dabei durch den Nachrichtenblock gesteuert. Das Resultat von F ist wieder ein 128 Bit Wert, der weitergereicht wird. Am Ende der Kette steht der Hash-Wert der Nachricht.

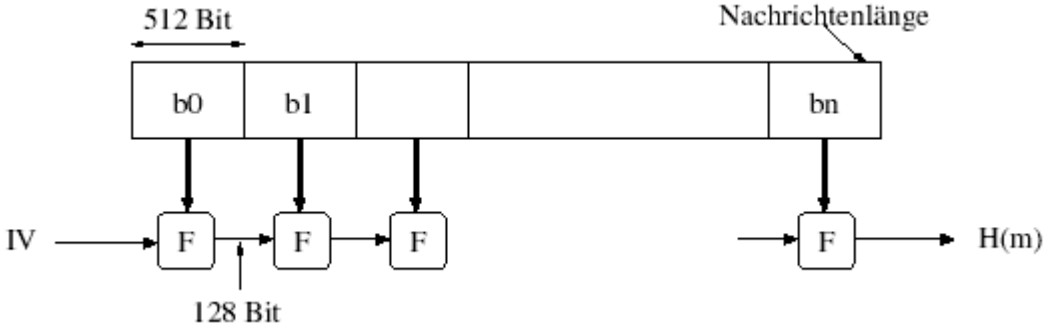


Abb. 2.11: Prinzip des MD5 Algorithmus

Die Funktion F selbst besteht aus 4 Runden mit je 16 Schritten, in denen UND, ODER, NICHT und XOR Operationen ausgeführt werden. An bestimmten Stellen wird ein Wert aus einer festgelegten Tabelle zu den einzelnen Zwischenwerten addiert.

SHA-1:

(von NIST) Ähnlich dem MD5. Nachrichtenlänge bis ca. 2^{64} – erzeugt einen 160 Bit Hash-Wert – 512 Bit Blockunterteilung, Auffüllen ähnlich wie bei MD5 – Initialisierung eines

160Bit Puffers und Berechnung des Hash-Werts analog zu MD5 – Funktion F mir 4 Runden zu je 20 Schritten, Schrittoperationen sind andere als bei MD5.

Da MD5 und SHA-1 öffentlich bekannt sind, ist damit keine echte Verbindlichkeit der Authentizität erreichbar. Dies ist erst möglich, wenn der Hash-Wert verschlüsselt übertragen wird.

2.6 Message Authentication Codes und Digitale Signaturen

2.6.1 Prinzip des Message Authentication Code - MAC

Die Idee ist, dass nicht nur die Nachricht selbst, sondern auch eine geheime Zusatzinformation in die Berechnung des MAC eingeht.

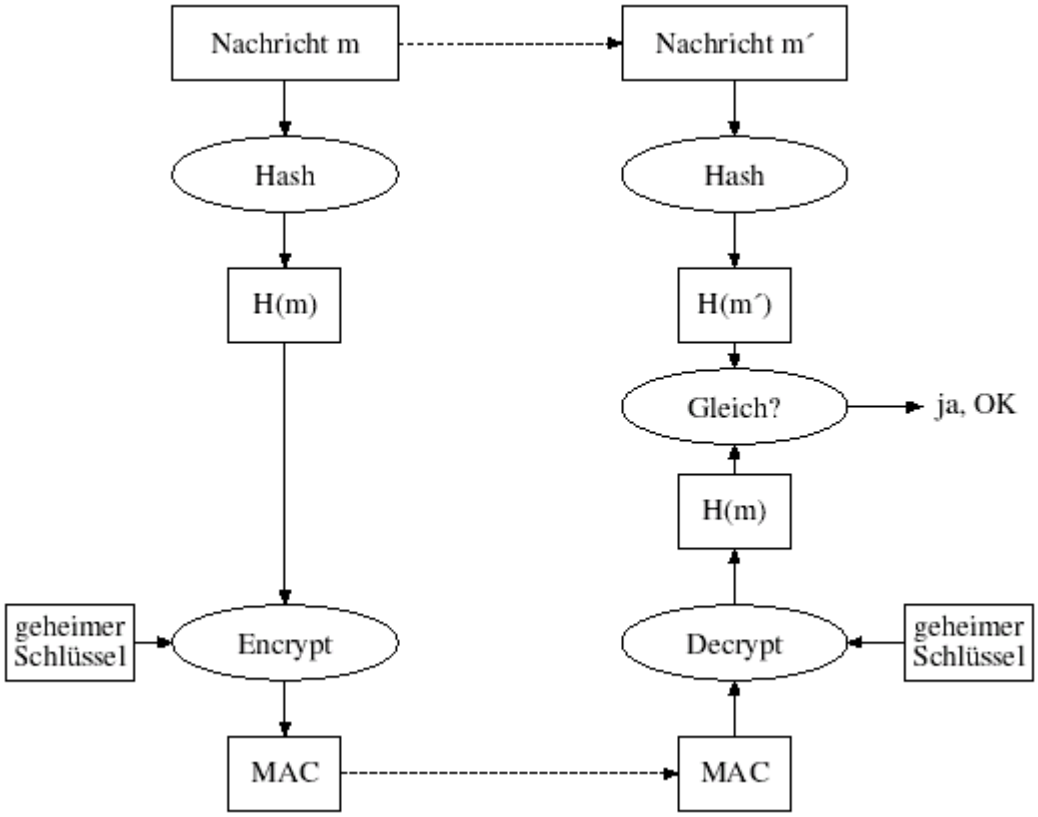


Abb. 2.12 Prinzip der Message Authentication

Prinzip: Der Sender einer Nachricht m berechnet H(m) und verschlüsselt diesen mit einem geheimen Schlüssel zu einem MAC. Dann werden die Nachricht und der MAC an den Empfänger geschickt. Dieser berechnet H(m') und vergleicht diesen mit dem entschlüsselten H(m). Bei Gleichheit ist Authentizität gegeben.
Dieses Protokoll erlaubt es aber dem Sender, zu bestreiten, dass dieser die Nachricht wirklich verschickt hat, da der Empfänger auch den geheimen Schlüssel besitzt und so den MAC selbst erzeugen könnte.

2.6.2 Prinzip Digitaler Signaturen

Es wird bei der Berechnung des MAC kein symmetrischer sondern ein asymmetrischer Algorithmus verwendet. (Unterschied zu MAC aus Abb. 2.12 ist fett gedruckt)

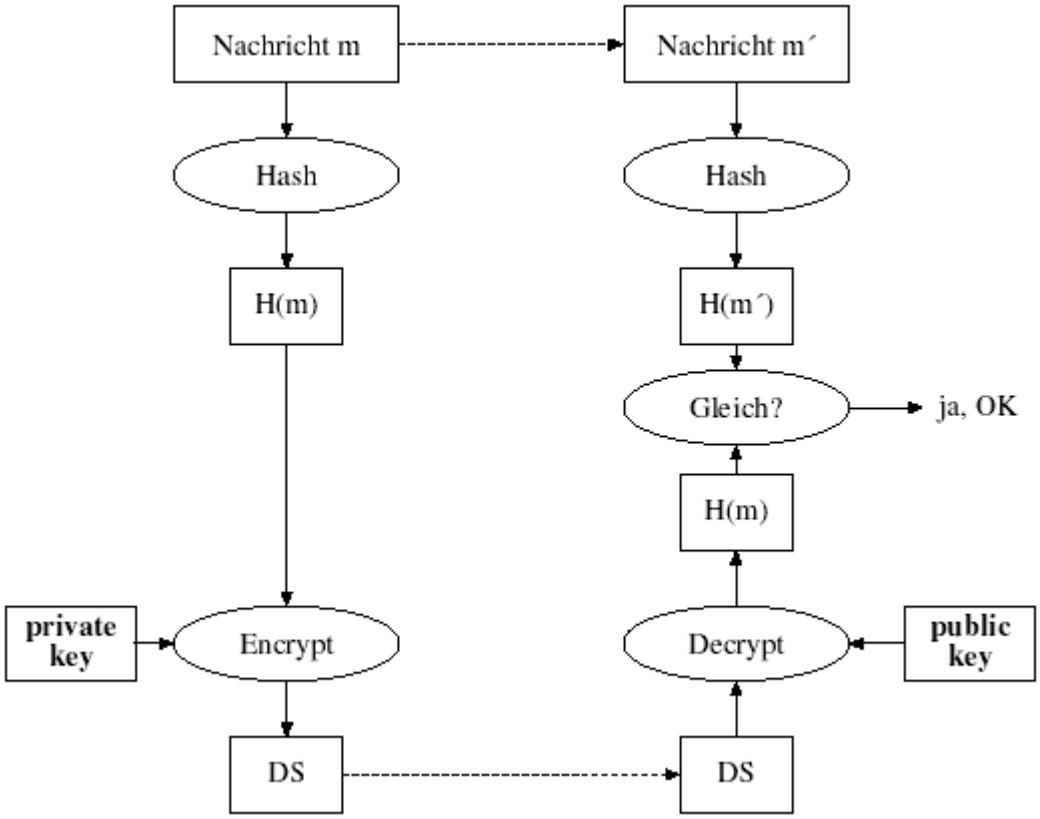


Abb. 2.13 Prinzip der digitalen Signatur

Ein Angreifer kann zwar die Nachricht verändern, aber die passende DS nicht selbst erzeugen. Ebenso kann der Empfänger nicht selbst eine passende DS erzeugen.

2.6.3 Algorithmen für Digitale Signaturen

Zur Berechnung der DS kann entweder RSA oder der in den USA als Standard definierte Digital Signature Algorithm DSA (beruht auf ElGamal), der auch für PGP und GPG verwendet wird, eingesetzt werden.

2.7 Zertifikate und Schlüsselmanagement

2.7.1 Zertifikate

Ein Zertifikat erlaubt die eindeutige Zuordnung eines öffentlichen Schlüssels zur einer Person od. Organisation. Die Zuordnung wird von einer vertrauenswürdigen Instanz, der Zertifizierungsstelle (*Certification Authority CA* oder auch *Trust Center*) hergestellt.
Der Zertifikats-Datensatz enthält u.a. folgende Felder:

- Name des Inhabers
- öffentlicher Schlüssel des Inhabers
- Seriennummer
- Gültigkeitsdauer
- Name der Zertifizierungsstelle

Dazu kommt die DS der Zertifizierungsstelle unter dem Zertifikat. Also ist ein Zertifikat auch nur eine digital signierte Nachricht.

Der Sender schickt nun die Nachricht, die DS und sein Zertifikat an den Empfänger. Dieser überprüft das Zertifikat mit dem *public key* der Zertifizierungsstelle und erhält damit Name und *public key* des Senders. Mit diesem *public key* entschlüsselt er die DS der Nachricht.

Die Überprüfung des public key der Zertifizierungsstelle erfolgt ebenfalls über ein Zertifikat, dass von einer übergeordneten Zertifizierungsstelle ausgestellt ist. Der *public key* der obersten Zertifizierungsstelle eines Landes (in Deutschland z.B. die Regulierungsbehörde für Telekommunikation u. Post – RegTP) veröffentlicht täglich den Hash-Wert ihres *public key* in den Medien. Mit diesem Schlüssel können die Zertifikate der Zertifizierungsstellen überprüft werden u. durch die Veröffentlichung in verschiedenen Medien ist ein Fälschung praktisch nicht möglich bzw. leicht erkennbar.

Anforderungen an eine Zertifizierungsstelle, damit Zertifikate Beweiskraft vor Gericht haben:
- Zuverlässigkeit
- Fachkunde
- Deckungsvorsorge

Geforderte Maßnahmen zur Sicherung gegen Zertifikatsfälschung:
- technische (z.B. sichere Gebäude und Systeme)
- organisatorische (z.B. Vier-Augen-Prinzip)

Auch große Firmen od. Organisationen mit vielen Mitarbeitern betreiben bereits eigene Zertifizierungsstellen. Da diese aber meist keine gesetzlich qualifizierte Zertifikate (entspricht eigenhändiger Unterschrift) für ihre Geschäftstätigkeit brauchen, lassen sie ihre Zertifikate von einer *Bridge-CA* zertifizieren. (siehe auch www.bridge-ca.org)
Um Zertifikate von Zertifizierungsstellen anderer Länder benutzen zu können, gibt es sog. *Cross-Zertifikate*, mit denen sich 2 Zertifizierungsstellen gegenseitig ihre *public key* signieren.

Bei der Erstellung eines Zertifikates wird von der CA die Identität des Registrars festgestellt. Je nach Verlässlichkeit dieser Identitätsfestestellung ergeben sich verschiedene Zertifizierungsklassen:

Klasse 1: Schwächste Identitätsprüfung. Ein Benutzer kann das Zertifikat elektronisch beantragen. Es wird z.B. nur die Existenz der angegebenen e-mail Adresse geprüft.

Klasse 2: Es werden auch weitere Identitätsmerkmale wie Telefonnummer, Adresse mit Hilfe öffentlicher Datenbanken geprüft.

Klasse 3: Der Benutzer oder die Organisation muss persönlich erscheinen oder von einem vertrauenswürdigen Dritten (z.B. Notar) vertreten werden.

Die Kosten der Zertifizierung steigen natürlich mit der Klasse der Zertifizierung. Im E-Commerce sollte man aber auf jeden Fall Zertifikate der höchsten Klasse fordern.

2.7.2 Schlüsselmanagement

Management öffentlicher Schlüssel:

- Schlüssel und Zertifikate haben nur eine begrenzte Gültigkeit
- Schlüsselhinterlegung z.B. für Firmenmitarbeiter, damit im Fall, dass der MA die Firma verlässt, trotzdem noch dessen verschlüsselte Dokumente entschlüsselt werden können. Wichtig ist aber, dass der MA einen separaten Schlüssel zum Signieren von

Dokumenten hat. Dieser Schlüssel darf auf keinen Fall hinterlegt und von jemand anderem verwendet werden -> Gefahr der Identitätsfälschung.
- Achtung: Bestrebungen von Ermittlungsbehörden und Regierungen zur Schlüsselhinterlegung bei staatlichen Stellen zur Verbrechensbekämpfung!
- Falls ein privater Schlüssel verloren geht oder gestohlen wird, kann der *public key* in eine Sperrliste (***certificate revocation list – CRL***) eingetragen werden und verliert somit seine Gültigkeit.

Management privater Schlüssel:

- Speicherung des *private key* in einer sicheren privaten Umgebung (***Personal Security Environment – PSE***).
- z.B. Chipkarte: ist durch einen PIN geschützt und hat einen eigenen Prozessor, somit kann die Verschlüsselung von z.B. des Hash-Wertes vom Prozessor der Chipkarte durchgeführt werden und somit muss der *private key* die Chipkarte gar nicht „verlassen".
- Der *private key* wird symmetrisch verschlüsselt in einer Datei abgelegt, wobei der symmetrische Schlüssel eine *Passwort-Phrase* (od. auch „Mantra" wie bei PGP) sein kann, die sich der Benutzer besser merken kann als den eigentlichen *private key*. Diese Datei kann dann z.B. auf USB-Stick abgelegt werden. -> ACHTUNG: Bei der Entschlüsselung des *private key* steht dann dieser im RAM des Computers, was ein Angriffspunkt wäre, vor allem dann, wenn vom OS dieser RAM-Bereich auch noch temporär auf die Festplatte ausgelagert wird.

2.7.3 Public Key Infrastrukturen – PKI

Derzeit haben sich Zertifikate noch nicht im großen Umgang durchgesetzt sondern es wird im E-Commerce noch zum Großteil mit Passwörtern oder PIN/TAN-Verfahren gearbeitet.

Technische Standards:

- Es muss sichergestellt sein, dass jeder Benutzer die Zertifikate auch einsetzen kann. Dafür braucht es ein einheitliches Datenformat, damit die Anwendungsprogramme die Zertifikate auch lesen können.
- **X.509** ist ein internationaler Standard für die Syntax und Semantik von Zertifikaten.
- Weiters wäre eine Technik wünschenswert, die transparent in den Anwendungsfeldern
  - Zutrittssysteme
  - Computer
  - speziellen Anwendungen gegenüber

  funktioniert, um einen breiten Einsatz zu ermöglichen.

Gesetzliche Anforderungen:

Sind z.B. in Deutschland im Signaturgesetz festgelegt. Viele Organisationen benötigen aber nur einfachere, „nicht Signaturgesetz konforme" Zertifikate. Die Authentisierung mit diesen „einfachen" Zertifikaten gegenüber jemand anderem bedingt aber, dass dieser dieses nicht konforme Zertifikat akzeptiert (also dem Aussteller vertraut) und auf seinem System installiert und gegebenenfalls pflegt. Dieses Verfahren ist aber bei großer Verbreitung nicht mehr praktikabel.

Wirtschaftliche Aspekte:

Für einen weiträumigen Einsatz sind folgende Vorarbeiten notwendig:
- Aufbau und Betrieb vertrauenswürdiger Zertifizierungsstellen
- Entwicklung und Verbreitung von Anwendungsprogrammen, die mit Zertifikaten umgehen können und die untereinander Daten austauschen können.

- Entwicklung und Verbreitung der erforderlichen zusätzlichen Hardware (z.B. Chipkartenleser)

3. Benutzersicherheit im Internet

3.1 Einführung

Thema Sicherheit aus dem Blickwinkel des Web-Benutzers.

3.2 Sichere e-mail im Internet

Anforderungen:

- Nachricht kann auf dem Übertragungsweg nicht mitgelesen werden.
- Sicherheit, dass Nachricht auf dem Übertragungsweg nicht verändert wurde.
- Authentizität des Absenders.

3.2.1 Pretty Good Privacy – PGP

(von Phil Zimmermann) Es enthält

- Algorithmen zur *public key* Verschlüsselung
- Algorithmen zur Erstellung digitaler Signaturen
- Hash-Algorithmen
- versch. *private key* Verschlüsselungsalgorithmen

PGP ist für Privatpersonen Open Source, OpenPGP und GNU Pivacy Guard (GPG) sind komplett Open Source.
Funktionen von PGP bzw. Open PGP:

- Verschlüsseln von Dateien (nur PGP)
- Sicheres Löschen (nur PGP)
- Vertrauliche Kommunikation (Verschlüsselung mir public key des Empfängers)
- Authentische Kommunikation (Signierung mir private key des Senders)
- Erstellen von Schlüsselpaaren
- Management eigener u. fremder Schlüssel

Verwendete Algorithmen:

	PGP 2.6.x	PGP 6.5.x	PGP 8.x	GnuPG
symmetrische Verschlüsselung	IDEA	IDEA CAST-128 3DES	IDEA CAST-128 3DES Twofish AES	 CAST5 3DES Twofish AES Blowfish
Hash-Funktion	SHA-1	SHA-1	SHA-1 MD5 RIPEMD-160	SHA-1 MD5 RIPEMD-160 Tiger
asymmetrische Verschlüsselung	RSA	RSA DSS Diffie-Hellmann	RSA DSS Diffie-Hellmann	RSA DSS ElGamal

PGP ist ein hybrides Verschlüsselungssystem. Die Nachricht wird nach einem symmetrischen Verfahren mit einem *session key* verschlüsselt, der *session key* selbst wird nach einem asymmetrischen Verfahren mir dem *public key* des Empfängers verschlüsselt und an die verschlüsselte Nachricht angehängt. Bei einer Nachricht an mehrere Empfänger wird nur der

session key jeweils mit den *public keys* alle Empfänger verschlüsselt und diese an die Nachricht angehängt, die Nachricht selbst wird nur einmal verschlüsselt. Die Komprimierung der Nachricht vor der Verschlüsselung erfolgt nur zur Speicherplatzeinsparung.

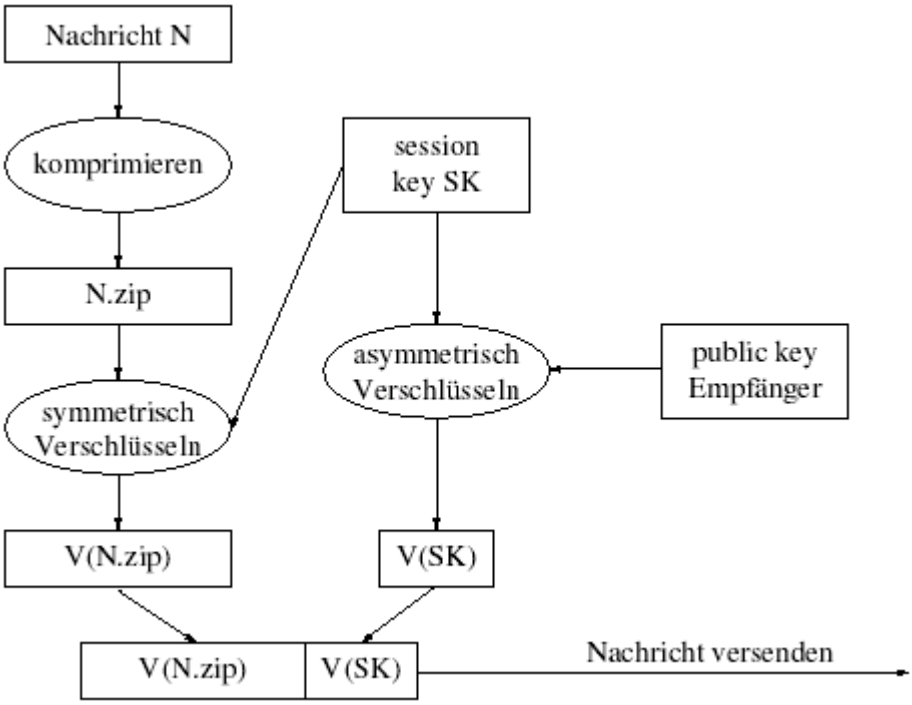


Abb. 3.1: Ablauf bei der Verschlüsselung mit PGP

Bei der digitalen Signierung von Nachrichten wird aus der Nachricht ein Hash-Wert berechnet, der vom Absender mit seinem privaten Schlüssel asymmetrisch verschlüsselt wird und an die Nachricht angehängt wird. Das ganze (Nachricht u. DS) kann jetzt bei Bedarf auch noch verschlüsselt werden, wie oben bereits beschrieben.

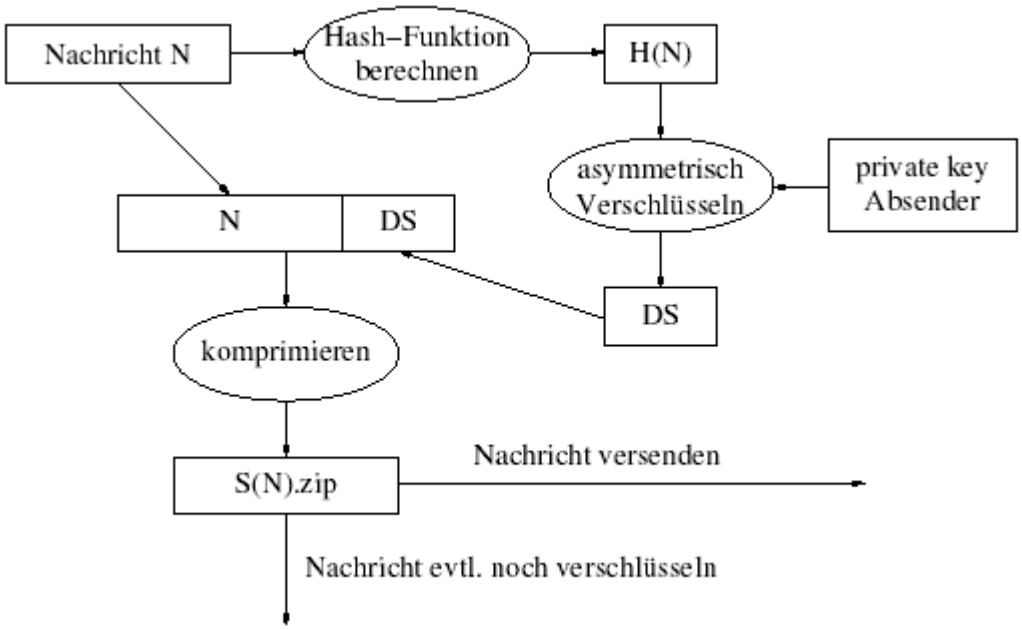


Abb. 3.2: Ablauf beim digitalen Signieren mit PGP

Bezugsquellen für PGP u. GPG:
www.pgp.com oder www.gnupg.org

Nach der Generierung des Schlüsselpaares wird der private Schlüssel noch einmal unter Eingabe einer Passwort-Phrase symmetrisch verschlüsselt und in einer Datei abgespeichert. Der öffentliche Schlüssel kann u.a. auf einem der unter www.pgp.net angeführten Schlüsselserver veröffentlicht werden.
Da es beim Einsatz von PGP/GPG keine zentrale vertrauenswürdige Zertifizierungsinstanz gibt, muss jeder User selbst ein „*web of trust*" aufbauen. Dafür erlaubt PGP verschieden Abstufungen für *owner trust* (Authentizität) und *signature trust* (Vertrauen in jemanden, dass er keine gefälschten öffentlichen Schlüssel signiert):

- ***ultimate trust***: nur gegenüber eigenen Person
- ***always trusted to sign keys***: von diesem Benutzer werden nur korrekte public keys signiert.
- ***usually trsuted to sign other keys***: normalerweise o.k.
- ***usually not trusted to sign other keys***: never trust
- ***unknown user***
- ***unknown trust***

Stellt man nun fest, dass ein vorliegender public key wirklich authentisch ist, signiert man ihn selbst gibt ihn damit dem eigenen System bekannt. Wird der signierte public key dem Partner wieder zur Verfügung gestellt, so kann er diesen mit dieser zusätzlichen Signatur weiter veröffentlichen und damit das Vertrauen in seinen public key erhöhen.

3.2.2 Secure MIME (S/MIME) – Secure Multipurpose Internet Mail Extension

S/MIME ist eine Erweiterung der MIME-Spezifikation, welche selbst eine Erweiterung der SMTP-Spezifikation ist:

RFC 2311–2312	S/MIME; Verschlüsselung, Signaturen, etc.
RFC 2045–2049	MIME; Mehrteilige, Multi-Media Nachrichten
RFC 821–822	SMTP und Inhaltsbeschreibung

Mit MIME wurden die Defizite des SMTP wie

- Nachrichteninhalt nur 7bit ASCII Zeichen
- Möglichkeit der Rückweisung großer Nachrichten
- Umcodierungsprobleme
- Keine Definition für Attachments

behoben, indem neue *header fields* eingeführt wurden die das *content transfer encoding*, den *content type* und die *content description* näher definieren. S/MIME erweitert MIME indem zusätzliche *content types* definiert werden.

MIME Content Transfer Encodings:

MIME schreibt keine feste Codierung vor, sondern definiert im Prinzip nur, wie die Auswahl der Codierung der anderen Seite mitgeteilt wird.

MIME Content Types:

Es ist eine 2-stufige Hierarchie von *content types* vorgesehen, wobei der Parameter *type* beschreibt, um welchen grundsätzlichen Type es sich handelt und *subtype* dann das eigentliche Format konkretisiert.

S/MIME Content Types:

Damit werden weitere content types eingeführt, die speziell auf Signaturen und Zertifikate ausgerichtet sind.

Mit S/MIME kann man eindeutig beschreiben, aus welchen Teilen eine Nachricht besteht (z.B. verschlüsselter Inhalt, Signatur, Zertifikat) und wie diese Teile zu interpretieren sind.

Weiters wird auch der benutzte Algorithmus festgelegt. S/MIME baut im Gegensatz zu PGP auf das formale System der Zertifizierungsstellen auf und wird daher eher im geschäftlichen Bereich eingesetzt. S/MIME-Plugins für freie e-mail-Clients findet man unter www.gnupg.org/aegypten/index.de.html .

3.3 Sicheres „surfen“ im Internet

Für ein sicheres surfen im Internet sind insbesondere folgende Schutzziele zu verfolgen:

- Vertraulichkeit
- Authentizität
- Privatheit (Anonymität) – siehe Kurs 1867

3.3.1 Secure Socket Layer – SSL

SSL erlaubt vertrauliche und authentische Kommunikation.

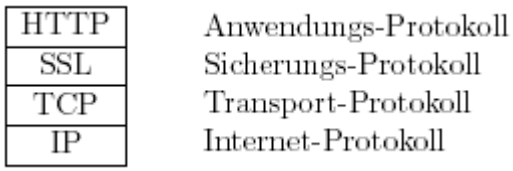


Abb. 3.3: Einordnung von SSL in den Protokollstack

Durch die Einordnung des SSL-Protokolls direkt unter der Anwendungsschicht ergeben sich im wesentlichen 2 Vorteile:

1. es erlaubt die mehr od. weniger dei unveränderte Weiternutzung bestehender Anwendungsprotokolle
2. und SSL kann außer für HTTP auch zur Absicherung anderer Anwendungsprotokolle wie POP3, SMTP od. telnet verwendet werden.

Die SSL-Schicht ist selbst in Schichten aufgeteilt:

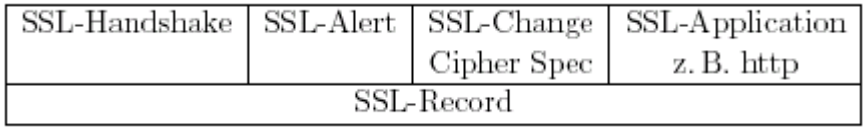


Abb. 3.4: Schichten von SSL

Das SSL-Record Protokoll ist dabei für die gesicherte Übertragung der Anwendungsdaten zuständig. Die Daten können dafür zuerst komprimiert werden, dann wird ein MAC an den Datenblock angehängt. Alles zusammen wird dann symmetrisch verschlüsselt und mit einem SSL header versehen an den transport layer übergeben.
Welcher Algorithmus und welcher geheimer Schlüssel verwendet werden, wird zwischen Web-Client und Web-Server über das SSL Handshake Protokoll ausgehandelt:

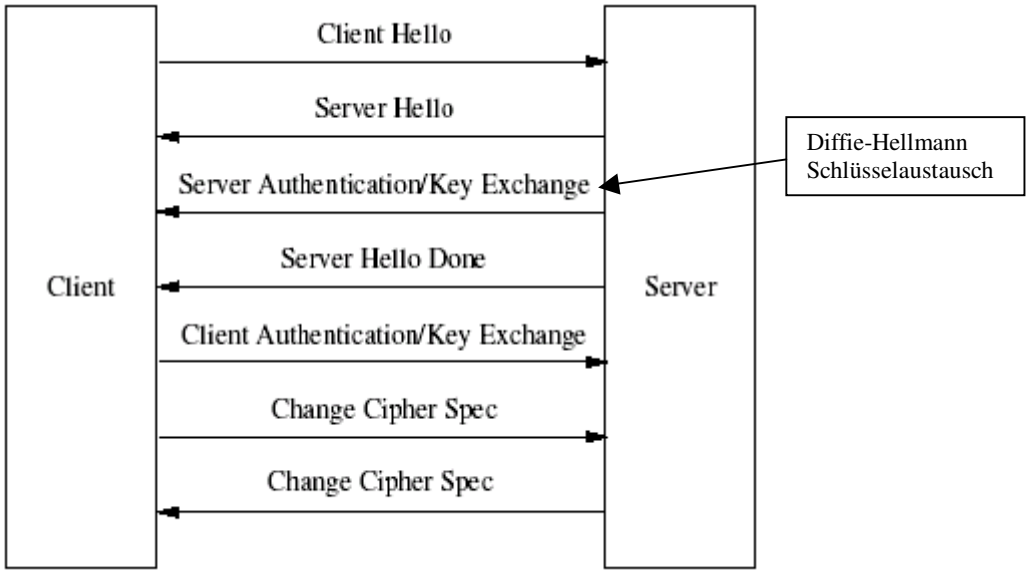


Abb. 3.5: SSL-Handshake Ablauf

Details findet man unter home.netscape.com/eng/ssl3 .
Damit das SSL-Handshake nicht für jede Datei der angesprochenen Web-Site durchgeführt werden muss, enthält SSL (ebenso wie HTTP/1.1) ein Sitzungskonzept.

3.3.2 Sicherheitseinstellungen von Web-Browsern

Verschlüsselung:

Eine SSL Verbindung erkennt man daran, dass die URL mit https://.. beginnt und dass im Browser ein eigenes Icon für den Security State verändert wird (bei Mozilla z.B. ist es das offene oder geschlossene Vorhängeschloss). Klickt man auf diese Icon wenn eine SSL Verbindung besteht, so werden die Zertifikatsinformationen angezeigt.

Die Tatsache, dass öffentliche Schlüssel von Zertifizierungsstellen dem Browser bekannt sein müssen, damit dieser eine sichere Verbindung ohne Rückfrage aufbaut, bringt folgende Probleme mit sich:

- Der Schlüssel muss irgendwo gespeichert sein (entweder im Browsercode direkt od. in einer Datei) und kann somit von einem Angreifer verändert werden.
- Jedes Zertifikat ist nur begrenzt gültig und nach Ablauf muss auch der Browser mit dem neuen Zertifikat wieder updated werden.

Cookies:

Da HTTP ein zustandsloses Protokoll ist, benötigt ein Web-Server für besondere Dienste eine Möglichkeit, die „Surf-Historie“ zumindest temporär zu speichern (z.B. wenn ein User zuerst Waren auswählt und dann auf die Bezahlseite geht) oder einen Benutzer wiederzuerkennen. Dies geht am einfachsten über sog. Cookies, ein Paar aus Variablenname und Wert die auf dem Client-Rechner gespeichert werden. Die Aufforderung an den Client, ein Cookie zu speichern, erfolgt mit einer Set-Cookie Zeile im Header der Server-Antwort.
Für Cookies gelten folgende Einschränkungen:

- ein Client speichert nicht mehr als 300 Cookies
- ein Cookie darf nicht größer als 40kB sein
- Pro DNS-Domain werden nicht mehr als 20 Cookies gespeichert

Details zur Cookie-Spec: http://wp.netscape.com/newsref/std/cookie_spec.html

Gefahren durch Cookies:

- Abgleich der Informationen mit anderen Datenquellen (z.B. Adressbücher) um Datensammlungen für zielgruppenorientiertes Marketing zu erhalten.
- Abhören des Cookies durch einen Angreifer und Verwendung des fremden Cookie um einen fremden User vorzutäuschen.
- Cookies identifizieren den Rechner u. nicht den Benutzer, was bei einem Mehrbenutzerplatz ev. problematisch sein kann.

Also Cookies generell eher restriktiv behandeln und Browser entsprechend konfigurieren.

Das W3C hat eine erste Spezifikation für „platform for privacy preferences – P3P“ entwickelt, die es Web-Servern erlaubt, die verwendeten Praktiken in Bezug auf Privatsphäre eindeutig u. in maschinenlesbarer Form zu beschreiben. Aufgrund dieser Beschreibung und er Web-Browser Konfiguration kann der Browser selbständig entscheiden, welche Daten er übermittelt.

3.4 Zugriff auf entfernte Rechner

Der *telnet*-Dienst oder auch *rsh* ist dafür prinzipiell hervorragend geeignet, allerdings hat er das große Manko, dass er Username und Passwort unverschlüsselt überträgt und somit nur bedingt in z.B. Intranets eingesetzt werden sollte.

3.4.1 Sichere Verbindung mit SSH

Mit SSH kann man sich sicher auf einem entfernten Rechner anmelden und dort dann sicher arbeiten. Die Authentifikation erfolgt dabei mittels public key Verschlüsselungsverfahren, die anschließende Kommunikation wird mit einem session key symmetrisch verschlüsselt. Neben SSH (bzw. SSH1 u. SSH2) gibt es noch eine freie Version OpenSSH, die im OpenBSD-Projekt entwickelt wurde und nur freie Algorithmen verwendet.
Weitere Infos unter www.openssh.org bzw. www.ssh.com .

Ablauf eines Verbindungsaufbaus: Client initiiert Verbindung -> Server antwortet mit ID-String u. Info über SW- u. Protokollversion -> Server sendet *public key*, eine stündlich generierten *server key* u. Zufallszahl (*cookie*) an Client -> Client generiert 256bit *session key*, verschlüsselt ihn mit den Schlüsseln des Servers und schickt ihn an den Server, zusammen mit Info das für Sitzung symmetrische Verschlüsselung gewählt wurde -> Server entschlüsselt *session key* und die folgende Kommunikation erfolgt verschlüsselt -> Client authentisiert sich gegenüber Server (Passwort oder RSA Authentisierung) -> Verbindung aufgebaut.

Für eine RSA Authentifizierung muss der Client zuerst mit *ssh-keygen* ein Schlüsselpaar erzeugen und dann seinen public key auf den Server kopieren. Beim RSA Authentisierungs-Vorgang generiert dann der Server eine *challenge*, verschlüsselt sie mit dem *public key* des Client und sendet sie an den Client. Der Client beweist seine Identität indem er die *challenge* entschlüsseln u. somit lösen kann.

Algorithmen und Verfahren von SSH:

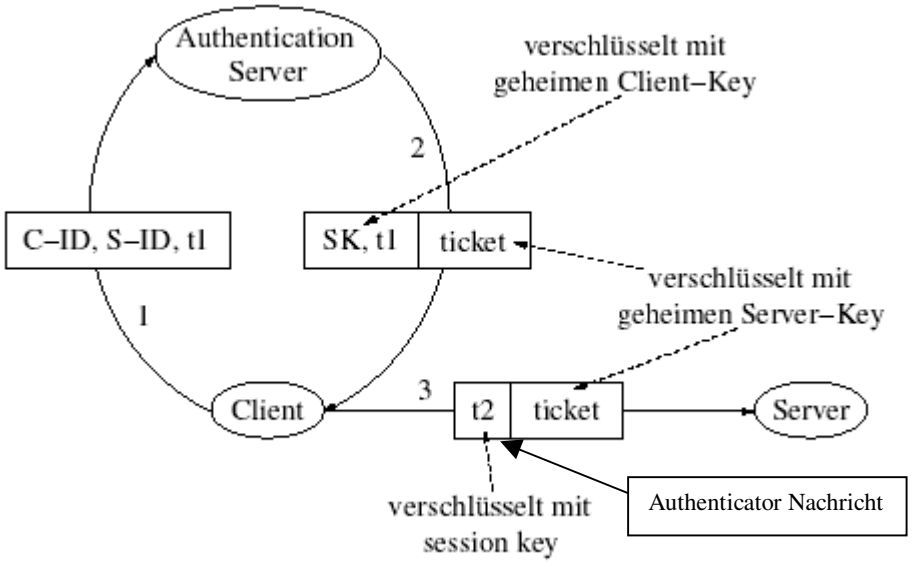
	SSH1	SSH2	OpenSSH
Server Authentisierung	RSA	RSA, DSA	RSA, DSA
Client Authentisierung	Paßwort	Paßwort	Paßwort
	RSA	RSA, DSA	RSA, DSA
symmetrische Verschlüsselung	3DES, DES	3DES, DES	3DES, Cast-128
	IDEA, RC4	IDEA, RC4	AES, Arcfour
	Blowfish	Blowfish	Blowfish

SSH unterstützt auch die Absicherung einiger anderer Anwendungen:
scp: damit können sicher Dateien zwischen 2 Rechner im lokalen Netz kopiert werden.
sftp: ein secure Version des ftp Protokolls für sichere Dateitransfers über das Internet.
X11 connections: sicherer Betrieb von X-Sitzungen
putty: sichere remote shell unter Windows.

Das zentrale Problem der SSH Administrierung ist die einmalige sichere Verteilung der Server-Keys.

3.4.2 Authentifizierung mit Kerberos

Damit können sich Benutzer oder aber auch bestimmte Dienste gegenseitig authentifizieren. Passwörter werden dabei nie im Klartext übertragen, es wird ein symmetrisches Verschlüsselungsverfahren – DES – verwendet. Von einem zentralen Kerberos Authentication Server werden sog. Tickets vergeben, mit denen sich ein Benutzer od. Dienst gegenüber einem anderen ausweisen kann. Dieses Ticket enthält den *session key*, Name des Clients sowie Beginn u. Ende der Gültigkeit. Für die symmetrische Verschlüsselung muss es einen gemeinsamen Verschlüsselungsschlüssel geben, der zunächst auf einem sicheren Kanal ausgetauscht werden muss. Außerdem muss der Authentication Server sehr gut gesichert sein.



C-ID, S-ID ... Client- bzw. Server-ID
t1 ... aktuelle Zeit oder Zufallszahl
t2 ... aktuelle Zeit

Abb. 3.6: Ablauf einer Authentifizierung bei Kerberos.

Mit der aktuellen Zeit t1, t2 soll verhindert werden, dass einen Nachricht abgefangen wird und dann zu einem späteren Zeitpunkt eventuell bereits manipuliert weitergesendet wird.

Da der SK bei jeder Ticket-Anforderung des Client neu erzeugt wird, kann auch niemand die Nachricht zwischen Client u. Server entschlüsseln. Allerdings muss der Client für jeden neuen Server (oder Dienst) ein neues Ticket vom Authentication Server abholen, also muss der User jedesmal sein Passwort eingeben. Das ist unpraktisch u. gefährlich.

Dieses Problem wird durch den ***Ticket-Granting-Server – TGS*** behoben. Der Client muss nur einmal beim Authentication Server ein ***Ticket-Granting-Ticket - TGT*** abholen (und dabei sein Passwort zur Entschlüsselung verwenden) und kann dann damit während der Gültigkeitsdauer (typischerweise 1 Arbeitstag) des *TGT* beim *TGS* seine Tickets abholen.

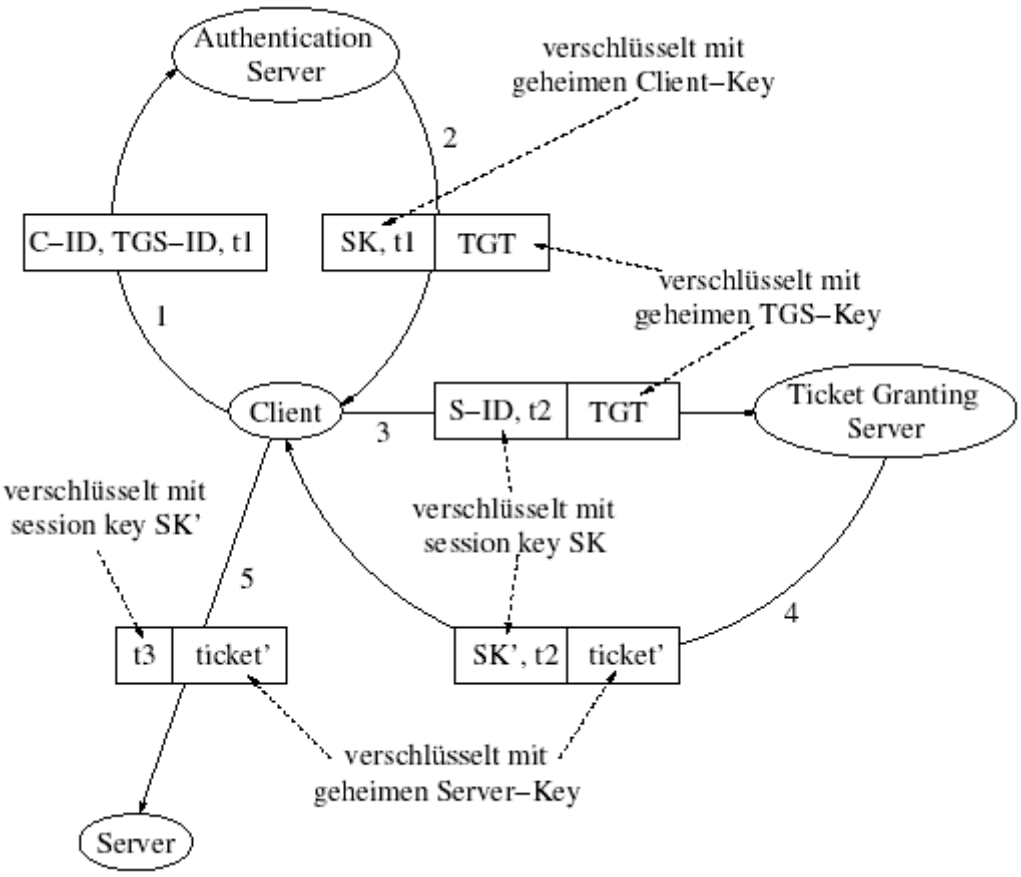


Abb. 3.7: Ablauf der Authentifizierung mit Hilfe des TGS.

Einsatzbereiche:

- Benutzerauthentisierung
- Benutzeranmeldung am Domänen Controller unter Win2000 Server
- SAMBA
- Authentisierung in verteilten Dateisystemen, z.B. Andrew File System - AFS

Weiter Infos unter: www.mit.edu/kerberos/ .

Problem: Bei großen und variierenden Benutzerzahlen wird die Hinterlegung des geheimen Passworts für jeden Benutzer am Kerberos-Server schnell zu einem allzu hohen Admin-Aufwand. In diesem Fall sind Benutzer-Zertifikate die bessere Lösung.

3.5 Schutz des privaten PC's

3.5.1 Virenscanner

Die Viren „I love you", „Code Red", „Sircam" und „Admin" haben in den Jahren 2000/2001 einen geschätzten Schaden von mehreren Milliarden Euro verursacht!

Schutz vor Boot Sektor Viren:

- ***Reihenfolge der Boot Geräte beachten***:
  1. Festplatte
  2. CD-ROM
  3. Diskettenlaufwerk

- ***Immer dieselbe Boot Diskette im Laufwerk lassen***: falls man überhaupt noch eine verwendet
- ***Rettungs Boot Diskette***: Virenfreie Diskette zur Systemrettung

Da Disketten immer seltener verwendet werden, sind Boot Sektor Viren kein wirkliches Thema mehr.

Schutz vor Dateiviren:

- ***Anti Viren Software:*** z.B. *Norton AntiVirus, AntiVir, AVG, McAfee VirusScan* usw. Anforderungen an ein Anti Viren Software:
  - *Erkennung der meisten bekannten Viren*
  - *Mehrere Betriebsmodi*:
    - batch-Betrieb der manuell gestartet wird und alle vom User spezifizierten Laufwerke bzw. Dateien prüft
    - Hintergrund-Betrieb: SW läuft ständig und überwacht jeden Dateizugriff (auch Netzzugriffe).
  - *Automatische Updates über das Internet*: zur Aktualisierung der Viren-Signaturen mit den neuesten Templates. Alternativ kann das auch von einem lokalen Server im Netzwerk geschehen, der immer am aktuellsten Stand gehalten wird.
- ***Makrovirenschutz in Office-Programmen einschalten:*** Dieser Schutz ist allerdings nur rudimentär, die Makrosprache selbst und das ActiveX-Konzept einem Programm erlauben, diese Option ohne Kenntnis des User zu deaktivieren.
- ***Keine e-mail Anhänge öffnen, die ausführbare Programme enthalten:*** ebenso das autom. Öffnen von Anhängen im Mail-Client deaktivieren.

3.5.2 Personal Firewalls

Ist ein Programm auf dem PC, das die Funktion einer Firewall möglichst gut nachbildet. Z.B. *Norton Personal Firewall, Aladdin eSafe Desktop, Zone Labs Zone Alarm*, usw. Sie arbeiten normalerweise als *Paketfilter*, manche auch als *Stateful Inspection Filter* oder *Application Level Gateway*. Eine Firewall sollte im wesentlichen folgende Funktionen erfüllen:

- Angriffe von außen auf den PC unterbinden.
- Schädliche Programme sollen keine Daten vom PC ins Internet schicken können.
- Schädliche Programme werden erkannt, wenn sie auf dem PC installiert werden sollen.

Angriffe von außen:

Diese Angriffe bestehen zunächst einmal einfach aus IP-Paketen an eine bestimmte Portnummer des PC. Die Personal Firewall unterstützt nun den User bei der Verwaltung offener und geschlossener Ports und lehnt alle IP-Pakete von außen an geschlossene Ports ab. Erfolgt der Angriff allerdings auf Ports, der normalerweise Pakete empfangen darf, so nützt die Personal Firewall nur selten.
Weiters kann sie den User bei der Laufwerksfreigabenverwaltung unter Windows unterstützen.

Unerlaubter Datentransfer nach außen:

Die PF kann überwachen, welche lokalen Programme eine Verbindung nach außen aufbauen wollen, und den User im Falle eines solchen Verbindungsaufbaus informieren. Dieser kann den Verbindungsaufbau zulassen oder nicht. Die Schwachpunkte dabei sind,

- dass der „durchschnittliche“ User nicht in der Lage ist anhand des angezeigten Programmnamens (trojanische Pferde) zu entscheiden, ob die Verbindung zu erlauben ist oder nicht.

- dass programmtechnisch die PF zwischen Anwendungsprogramm u. der Netzwerkschicht des OS sitzt. Unter Windows kann die PF aber nicht verhindern, dass lokale Programme an der PF vorbei direkt mit dem OS kommunizieren und so eine Verbindung aufbauen. Unter WinNT/2000/XP haben Programme mit Administratorrechten das selbe Zugriffsrecht auf die Netzwerkschicht wie die PF!
- Konfigurationsdaten stehen in einer Datei. Hat z.B. ein Trojaner Zugriff auf diese Datei, so kann er sich selbst die Erlaubnis für Verbindungen nach außen erteilen.
- e-mail Würmer können nicht mehr kontrolliert werden, wenn der Mail-Client ohne Rückfrage Verbindungen ins Internet aufbauen darf.
- Ungewöhnliche Nutzungen von erlaubten Protokollen wie HTTP können nicht erkannt werden. z.B. könnte mit einer URL http:/xxx.com/index.html?Passwort=asdfg ein User-Passwort an den Web-Server xxx.com übertragen werden. Alles hinter dem Fragezeichen können vertrauliche Infos sein, die der Webserver nicht auswertet sondern nur speichert.

3.5.3 Sichere Windows Konfiguration

Windows XP erlaubt es, Rechte für einzelne Benutzer zu vergeben. Dafür sind folgende Punkte notwendig:

1. Das System muss verschiedene Benutzer (**Subjekte**) kennen und unterscheiden können. (Auch Programme sind Subjekte, da sie auch Aktionen ausführen).
2. Das System muss Rechte verwalten und überprüfen können. Rechte beziehen sich auf Ressourcen (**Objekte**) im System, auf denen Aktionen ausgeführt werden können.
3. Das System muss bestimmte Aktionen kennen und ermöglichen. Bsp. sind Lesen, Schreiben und Ausführen.

Damit nicht für jeden Benutzer die Rechte jedesmal explizit definiert werden müssen, gibt es das Konzept der Benutzergruppen, für die diese Rechte definiert werden. Benutzer werden dann der entsprechenden Gruppe zugeteilt.
Die eigentlichen Rechte werden in XP in **Access Control Lists – ACL** verwaltet. Im Access Control Entry – ACE der ACL stehen für einen Benutzer die Zugriffsrechte auf das Objekt. Verbote sind dabei vorrangig, Rechte sind vererbbar (z.B. für alle Unterverzeichnisse).

Benutzer anlegen:

User können mit dem Befehl `net user` oder mit einem graphischen Verwaltungstool angelegt werden. Zu beachten ist, dass jene Benutzer, die während der Installation des OS angelegt wurden, automatisch der Gruppe Administrator zugeordnet werden.

ACL definieren:

Mit Hilfe des Kommandos `calcs` oder mit dem Kontextmenü Eigenschaften zu Ordnern und Dateien kann man die Zugriffsrechte auf Objekte einstellen. Damit das OS die Zugriffsrechte tatsächlich prüfen kann, müssen die ACL's im Dateisystem gespeichert sein. Dafür muss aber NTFS als Dateisystem gewählt werden.

Windows Explorer:

Dieser ist standardmäßig in Bezug auf Sicherheit ungünstig konfiguriert. Folgende Konfigurationsänderungen sollten durchgeführt werden:

- Namen immer komplett anzeigen
- Detail Liste anzeigen
- kein Autostart von eingelegten Medien

Zustand überwachen:

Ab Win2000 wird der Microsoft Baseline Security Analyser – MBSA angeboten. Dieses Programm untersucht Schwachstellen wie:

- Fehlende Sicherheitsupdates für das OS od. MS-Programme
- Zu schwache oder fehlende Passwörter für Benutzerkennungen
- Fehlende oder deaktivierte Firewall
- Fehlender oder deaktivierter Virenscanner
- Anmeldemöglichkeiten über das Netz
- Freigabe von Laufwerken
- Ausgewählte Dateisysteme

3.6 Aktive Inhalte

Aktive Inhalte sind Programme die in Web-Sites enthalten sind und zusammen mit deren Inhalt auf den Client Rechner geladen und dort ausgeführt werden.

3.6.1 JavaScript

Ist eine kompakte, objekt-basierte Sprache zur Erstellung von Programmen im WWW. Moderne Web-Browser enthalten einen Interpreter für JavaScript. Das JavaScript muss nicht komplett im HTML-Code der Web-Site enthalten sein, es kann auch nur der Link auf das Script enthalten sein.
Eine JavaScript-Programm hat Zugriff auf verschieden Daten des Web-Browsers, wie

- die gespeicherten Cookies
- oder die History-Liste.

Die Sprachmittel erlauben allerdings keinen Aufbau von Netzverbindungen zu anderen Computern, es kennt aber Funktionsaufrufe. Sie können auch HTML-Seiten laden. Ein JavaScript wird meist als Reaktion auf ein Ereignis (z.B. Mausklick) gestartet.

Risiken – Gefahren für die Vertraulichkeit:

Ein JavaScript-Programm kann Daten nach außen weiter geben, indem es eine URL anfordert, die auf ein CGI-Script (laufen auf Web-Server) zeigt, an das es dann als Parameter die vertraulichen Daten übergibt. Für die Parameterübergabe gibt es 2 Möglichkeiten:

1. Die Parameter werden an die URL des CGI-Scripts angehängt, z.B.
   `<A HREF=www.xy.z/cgi-bin/script?name=ihr+name&action=machwas>` .
2. Die Parameter werden separat mit HTTP Methode POST übertragen. Dabei werden die Parameter für den Benutzer unsichtbar übertragen.

Die *Same Origin Policy* beschränkt die Möglichkeiten von JavaScripts indem sie Scripts, die nicht aus der selben Quelle wie die HTML-Seite auf der sie stehen stammen, keinen Zugriff vom Interpreter auf die Browserdaten erhalten. Der Benutzer sollte JavaScript im Browser möglichst deaktivieren oder entsprechend einschränken.

3.6.2 Java

Java ist eine objektorientierte Programmiersprache. Ein Java Compiler erzeugt einen **Java Byte Code**, der dann von einem Interpreter, der **Java Virtual Machine – JVM** ausgeführt wird. Der Interpreter abstrahiert also von einer konkreten HW-Plattform bzw. OS. In HTML Seiten könne **Java Applets** integriert werden, die dann vom integrierten Interpreter des Web-Browser ausgeführt werden. Dies Applets laufen aber im Gegensatz zu einem normalen Programm in einer besondern, eingeschränkten Laufzeitumgebung, der **Java Sandbox**. Java Applets ist es damit nicht gestattet

- Dateien des lokalen Systems zu lesen oder zu manipulieren
- andere Programme auf dem lokalen System zu starten
- Netzverbindungen zu andern Computern aufzubauen, außer zu dem Computer von dem das Applet geladen wurde

Möchte das Java Programm (wie auch beim JavaScript) weitergehende Rechte beim Benutzer anfordern, so muss es signiert sein.

Risiken:

Obwohl der Java Compiler bei der Übersetzung die Einhaltung bestimmter Vorgaben prüft, könnte doch mit einem modifizierten Compiler sicherheitskritischer Java Byte Code generiert werden. Deswegen führt die JVM auf mehreren Stufen Überprüfungen durch:

1. ***Byte Code Verifier***: Überprüft vor der Ausführung, ob eventuell Stacküberläufe im Byte Code enthalten sind, oder ob z.B. Funktionen mit den richtigen Parametern aufgerufen werden.
2. ***Java Class Loader***: Soll verhindern, das ein Angreifer Code versucht, wichtige Basisklassen des Java Systems zu manipulieren.
3. ***Security Manager***: Zugriffe auf sicherheitskritische Ressourcen aus einem Java Programm oder Applet müssen von ihm erlaubt werden. Dazu wird die Herkunft des Programms geprüft. Lokale Programme haben mehr Rechte als Applets aus dem Internet.

3.6.3 ActiveX

Mit ActiveX kann ein Benutzer beliebige Programme aus dem Internet einfach und transparent lokal ausführen. Das auf einer HTML-Seite befindliche ActiveX Control wird in einem ActivX Container ausgeführt. Die Funktion des ActivX Containers übernimmt z.B. der Internet Explorer.

Risiken:

Ein ActivX Control kann auf alle Funktionen des OS zugreifen (Dateien lesen, Netzverbindungen aufbauen, Daten senden, usw.). Das Sicherheitskonzept von M$ sieht zwar vor, das ActivX Controls zertifiziert werden können. Damit ist aber nur die Herkunft sichergestellt, nicht aber, welche Funktion das ActivX Control ausführen wird. Aus Sicherheitsgründen sollten deshalb ActivX Controls gänzlich deaktiviert werden.

3.6.4 Sonstige aktive Inhalte

Auch in den eigentlich als Seiten-Beschreibungssprachen gedachten Dateiformaten **Portable Document Format – PDF** und **PostScript** können aktive Inhalte stehen. In beiden Sprachen sind Zugriffe auf Dateien des lokalen Systems möglich, ebenso können andere Programme gestartet werden. Ein Angreifer muss aber im vorab wissen, wie die Dateien heißen, wo sie stehen und welche Programme auf dem lokalen System installiert sind. Die Gefahren durch aktive Inhalte in PDF- oder PostScript-Dateien sind deshalb eher gering.

4. Anbietersicherheit im Internet

4.1 Einführung

Thema Sicherheit aus dem Blickwinkel des Verantwortlichen für Rechner im Internet.

4.2 Sichere Web-Server

4.2.1 Betriebssystemunabhängige Aspekte

Minimales System:

Es sollte ein minimales System aufgebaut werden, also nur Programme und Dienste installieren, die auch unbedingt notwendig sind. Nicht notwendige Programme sind z.B.

- GUI
- Programme zur SW-Entwicklung
- Treiber für Geräte, die nicht installiert sind.

Richtige Konfiguration:

Die Standardinstallation eines OS ist meist nicht geeignet, es muss also eine entsprechende Konfiguration des OS manuell durchgeführt werden (Stichwort offen Ports).

- Es sollte auch nur jene Benutzerkennungen eingerichtet werden, die wirklich benötigt werden. Irgendwelche vom OS während der Installation autom. erzeugte Benutzer sollten wieder gelöscht werden.
- Für die einzelnen Dienste sollten eigene Benutzer eingerichtet werden mit möglichst eingeschränkten Rechten. Damit hat ein Angreifer bei einem erfolgreichen Angriff (z.B. buffer overflow) auf einen Dienst nur beschränkte Rechte und der mögliche Schaden kann somit minimiert werden.

Aktuelle Software:

Alle installierten Programme und Dienste immer auf dem aktuellsten Stand halten und gegebenenfalls Patches einspielen, die entdeckte Schwachstellen beheben. Z.B. Mailing-Listen des CERT (www.cert.org) oder des DFN-CERT (www.cert.dfn.de) abonnieren.

Selbst „hacken“:

Mit typischen Hacker-Tools wie z.B. *Scanner* (z.B. *COPS* od. *saint*) das eigene System „angreifen“, und zwar regelmäßig.

Integritätstest:

Das System wird regelmäßig auf Veränderungen untersucht. Gefunden Änderungen die man nicht erklären kann, deuten auf einen Angriff hin. Integritätstest-Programme unterstützen diese Analysen, indem vom Systemzustand ein Snapshot aufgenommen wird, d.h. zu jeder Datei ein kryptographische Prüfsumme (z.B. MD5) berechnet wird und diese auf einem externen Datenträger abgespeichert wird (nicht auf HD, da dann die Möglichkeit besteht, dass diese Prüfsummen vom Angreifer verändert werden). Bei einer späteren Analyse werden dann die aktuellen Prüfsummen mit den archivierten verglichen und so Veränderungen erkannt.

Log-Dateien kontrollieren:

Durch regelmäßige Kontrolle der Log-Dateien erkennt man das „normale“ Verhalten der Benutzer. Außerdem können so erfolglose Angriffe erkannt werden und präventive Gegenmaßnahmen gestartet werden.

Regelmäßiges Backup:

Neben dem Backup muss auch geprüft werden, ob die Sicherung im Ernstfall dazu taugt, das System zu rekonstruieren.

Keine vertraulichen Daten im Web-Bereich:

Alle vertraulichen Daten werden an anderen Stellen im Dateisystem gespeichert. Der Web-Server wird so konfiguriert, dass er keine symbolischen Links verfolgt.

Sichere Administration entfernter Web-Server:

Entweder über eine eigene Modem-Verbindung (ist aber langsam) oder aber über einen sicheren Kanal (z.B. *ssh*) über das Internet.

4.2.2 Betriebssystemabhängige Aspekte

Windows NT, Windows 2000 und Windows XP:

Bei der Installation des Internet Information Server – IIS ist darauf zu achten,

- dass nicht auch der Mail-Server, der News-Server oder der M$-Index-Server installiert werden
- keine Beispieldateien installiert werden
- der IIS so konfiguriert wird, dass alle nicht benötigten Dateiverknüpfungen deaktiviert werden
- der Zugriff auf die Registry des Servers über Netzverbindungen gesperrt wird
- die mit WinXP mitgelieferte Firewall aktiviert und entsprechend konfiguriert wird.

Aktuelle Hinweise auf Security Updates oder sonstige Sicherheitshinweise findet man unter www.microsoft.com/security .

UNIX:

Zu beachten ist hier

- bei der Installation bereits ein sinnvolle Partitionierung vorzunehmen
  - Trennung von beschreibbaren und nicht beschreibbaren Verzeichnissen
  - und von System- und Userverzeichnissen.
  - ev. Root-Verzeichnis des Server-Prozesses in eigene Partition legen
- nicht benötigte Dienste in der *inetd*-Konfigurationsdatei deaktivieren (*inetd* wartet auf eingehende IP-Packete und startet je nach adressierter Portnummer den entsprechenden Prozess). *xinetd* ist ein verbesserter Nachfolger von *inetd*.

4.2.3 Konfiguration des Web-Server-Prozesses

Anhand des *apache* Web-Servers wird dargestellt, worauf bei einer sicheren Installation zu achten ist. Die Konfiguration von *apache* erfolgt über Konfigurationsdateien.

- Nach einer Änderung einer Konfigurationsdatei, muss der Server neu gestartet werden, damit die Änderungen übernommen werden.
- Die Konfigurationsdateien sollten dem Admin gehören und nur von diesem verändert od. gelöscht werden können.
- Der Server-Prozess sollte unter einer Benutzerkennung mit entsprechend eingeschränkten Rechten laufen. Dies kann in `httpd.conf` eingestellt werden. (Server startet immer unter *root*, kann sich aber dann selbst einem anderen *user* zuordnen.)

- Die Einträge *ServerRoot* und *DocumentRoot* legen fest, wo im Dateibaum das Root-Verzeichnis und das Dokument-Verzeichnis des Web-Servers liegen.
- Mit *DirectoryIndex* wird festgelegt, welche Datei der Web-Server liefert, wenn die URL nur aus einem Verzeichnisnamen besteht. (z.B. soll index.html geliefert werden). Fehlt dieser Eintrag oder die spezifizierte Datei `index.html`, so liefert der Server eine Liste der im angesprochenen Verzeichnis liegenden Dateien. Dies ist ein beträchtliches Sicherheitsrisiko, da damit ein Angreifer dann direkt Dateien lesen kann, auch dann, wenn dafür gar kein Link auf der Seite eingetragen ist.
- Zugriffsrechte auf einzelne Verzeichnisse können mit dem Parameter *Directory* gesetzt werden, und zwar abhängig, von wo aus (IP-Adresse) der Zugriff erfolgt bzw. auch von welchem Benutzer.
- Standardmäßig ist in *apache* keine SSL-Unterstützung implementiert. Es gibt aber SSL-Erweiterungsmodule wie *mod_ssl* bzw. ein eigenes *apache_ssl* Projekt (www.apache-ssl.org). Für einen sinnvollen Einsatz von SSL benötigt man ein Zertifikat, für erste Tests kann man sich aber auch mit openssl selbst eines erstellen.

4.3 Computer Forensik

Bei einem erfolgreichen Angriff muss man versuchen

1. den Angriff überhaupt zu erkennen (Aufgabe der IDS -> Kurs 1867),
2. Beweise zu sichern, um den Angreifer zur Verantwortung ziehen zu können,
3. den Angriff zu analysieren, um die Ursachen zu ergründen,
4. die Angriffsspuren zu beseitigen, ebenso wie die zugrunde liegenden Schwachstellen des Systems.

4.3.1 Beweise sichern

Dafür muss der Zustand des Systems exakt, persistent und unveränderbar gesichert werden (dies steht primär in Konflikt mit dem Ziel, das System so schnell wie möglich wieder regulär zu betreiben). Zum Systemzustand gehören:

- Inhalt des Hauptspeichers
- Inhalt der Festplatten
- Liste der angemeldeten Benutzer
- Liste der laufenden Prozesse
- geöffnete Netzverbindungen und
- aktuelle Systemzeit.

Sichern nicht persistenter Informationen:

Dilemma: um den Zustand des Systems nicht zu verändern, muss man Kommandos zur Sicherung verwenden, die eventuell durch den Angriff bereits manipuliert sein können. Man muss also dafür sorgen, dass man einen Satz vertrauenswürdiger Kommandos zur Verfügung hat, die der Angreifer nicht manipulieren konnte *(z.B. auf Diskette, CD-ROM o.ä. -> Programme müssen statisch gebunden sein -> durch Verwendung externer Medien wird aber auch Systemzustand verändert! -> und bei Sicherung des Systems müssten ebenfalls die Programme von den externen Medien verwendet worden sein).*

Unter *UNIX/Linux* stehen folgende Kommandos für das Sammeln von Informationen über den Systemzustand zur Verfügung:

- ***Hauptspeicher:*** Allgemeine Infos im Verzeichnis `/proc/meminfo`, Infos über den virtuellen Speicher einzelner Prozesse in `/proc/PID/mem` .

- ***Festplatte:*** Allg. Infos über Dateisystem und mount-points einzelner Geräte erhält man mit `df`, Infos über die Festplatte und deren Partitionierung über `fdisk -l` .
- ***Benutzer:*** who zeigt an, welche Benutzer aktuell angemeldet sind.
- ***Prozesse:*** Die laufenden Prozesse werden mit `ps -elf` angezeigt. Weiter Infos zu jedem Prozess findet man in `/proc/PID` .
- ***Netzverbindungen:*** Infos über die Konfiguration der Netzwerk-HW erhält man mit `ifconfig` und `arp`, mit `netstat` über die aktiven Netzwerkverbindungen und mit `lsof`, welche Dateien od. Ports von welchen Prozessen geöffnet sind.
- ***Systemzeit:*** mit `date` .

Unter *Windows* benötigt man spezielle Hilfsprogramme wie z.B. jene, die frei unter http://www.sysinternals.com/ erhältlich sind (z.B. *PsTools*):

- ***Benutzer:*** Mit `psloggedon` (aus *PsTools*) erhält man eine Liste der lokal angemeldeten Benutzer.
- ***Prozesse:*** Anzeige der Prozesse einmal mit dem *TaskManager* oder mit `pslist` (aus *PsTools*), wobei damit auch weitergehende Prozesseigenschaften angezeigt werden.
- ***Netzverbindungen:*** Hier gibt es ebenfalls die Kommandos `arp` und `netstat`. Die Windows-Variante von ifconfig heißt `ipconfig`. Mit `fport /p` werden die geöffneten Ports angezeigt. Mit dem Programm `tcpview` (von Sysinternal) wird auch angezeigt, welcher Prozess auf dem eigenen Rechner ein Endpunkt einer Verbindung ist.

Die Zustandsinformationen müssen dauerhaft gesichert werden. Entweder werden die Ausgaben der einzelnen Programme in eine Datei auf einem externen Medium *(ändert allerdings wieder Systemzustand)* umgeleitet, oder man verwendet z.B. unter *Linux* den Befehl `script`, der eine neue Shell startet und alle Aktionen darin protokolliert. Alternativ können die Zustandsinfos über eine eventuell vorhandene Netzverbindung gesichert werden. Dazu muss aber der Netzwerkteil noch vertrauenswürdig sein.

In jedem Fall muss man alle durchgeführten Schritte in einem schriftlichen Protokoll dokumentieren.

Sichern persistenter Informationen:

Entweder baut man die betroffene Festplatte aus, oder man startet das System von einer CD o.ä. mit einem garantiert integeren OS und erstellt dann ein Image der gesamten Festplatte oder einzelner betroffener Partitionen. Unter *UNIX/Linux* kann man dafür das Programm `dd` verwenden, unter *Windows* gibt es dafür spezielle meist kommerzielle Programme. Von allen gesicherten Images muss eine kryptographische Prüfsumme erstellt und protokolliert werden, um eine spätere Manipulation der Images erkennen zu können.

4.3.2 Angriff analysieren

Dateisystem untersuchen:

Mit einer Kopie der gesicherten Festplattendaten sucht man nach Dateien, die

1. in irgendeiner Form versteckt wurden
2. zu den Systemdateien gehören und nicht mehr im ursprünglichen Zustand sind, oder
3. vor kurzem erst geändert wurden.

Ad 1.) Neben den Dateien mit gesetztem *Hidden*-Attribut (in Unix mit Punkt am Dateianfang) sind vor allem ausführbare Dateien in ungewöhnlichen Verzeichnissen bzw. ausführbare Dateien mit ungewöhnlichem Dateitypbezeichner (z.B. ausfürbare `datei.jpg`) verdächtig. Unter *UNIX* findet man letztere mit dem Kommando `file .`

Ad 2.) Änderungen an Systemdateien können zum Beispiel über den Vergleich der kryptographischen Prüfsumme mit jener einer Systemsicherung vor dem Angriff erkannt werden. Das IDS *tripwire* macht dies automatisiert. Die Konfigurationsdateien `/etc/passwd` und `/etc/shadow` (unter *Linux*) sollten daraufhin kontrolliert werden, ob neue Benutzer angelegt wurden oder Privilegien bestehender Benutzer verändert wurden.

Ad 3.) Die zuletzt geänderten Dateien erkennt man an den Metainformationen der Dateien über den Zeitpunkt der letzten Änderung bzw. letzten Zugriffs, falls die Dateisysteme diese Infos unterstützen (z.B. *NTFS, ext2, ext3 u. ReiserFS* tun es).

Verdächtige Dateien können dann mit dem Kommando `strings` nach Zeichenketten untersucht werden, die auf eine Manipulation hinweisen. Auch bereits gelöschte Dateien (falls dieser Platz auf der Festplatte noch nicht überschrieben wurde) können so rekonstruiert werden, wenn man das Kommando auf die gesicherten HD-Images anwendet.

Protokolldateien analysieren:

Ein Angriff könnte auch Spuren in den Log-Dateien hinterlassen, dies setzt aber voraus, dass das Logging aktiviert war und der Angreifer keine Zeit hatte, die Log-Dateien zu manipulieren. Bei der Analyse sind auf jeden Fall Einträge über

- erfolglose Anmeldeversuche,
- erfolgreiche Anmeldeversuche kurz vor dem vermuteten Angriffbeginn,
- neu gestartete oder neu installierte Dienste,
- Infos, wann das System von wem heruntergefahren und/oder neu gestartet wurde und
- Änderungen an der HW-Konfiguration

interessant.
Spezielle Software zur Unterstützung der Forensik ist z.B.

- **EnCase**: Windows Software www.guidancesoftware.com
- **The Sleuth Kit**: Sammlung von UNIX-Tools www.sleuthkit.org/sleuthkit
- **Autopsy**: GUI zu Sleuth Kit www.sleuthkit.org/autopsy
- **WinHex:** zum Auffinden u. Rekonstruieren gelöschter Dateien

Gegenmaßnahmen:

Identifizierung der Schwachstelle, die zum Einbruch geführt hat. Mögliche Schwachstellen sind:

- *Ausnutzung einer bekannten Schwachstelle(Programmfehler) in einem Dienst*: SW-Upgrade des Dienstes oder Dienst vorübergehend sperren.
- *Kompromittierung einer lokalen Benutzerkennung inkl. des Passworts*: Vergabe eines neuen und sicheren Passworts.
- *Kompromittierung bestimmter Software*: durch Viren, Würmer, Trojaner. Benutzerverhalten bez. dem Umgang mit neuen Programmen od. e-mail Anhängen ändern.

4.3.3 System aktualisieren:

Dadurch sollen Angriffsspuren entfernt und das System wieder in einen vertrauenswürdigen und benutzbaren Zustand versetzt werden. Also

- Rückgängig machen aller Veränderungen durch den Angreifer
- Verhinderung zukünftiger Angriffe dieser Art.

Unter Windows ist dafür meist eine Neuinstallation des Systems notwendig eventuell durch Einspielen eines alten Images. Auf jeden Fall müssen danach alle aktuellen Sicherheits-Updates für die Systemsoftware eingespielt werden. Danach sollte wieder ein Sicherungs-Image der neuen Installation gezogen werden.

4.4 Firewall

Die Aufgabe einer Firewall ist zu verhindern, dass

- dass nicht autorisierte Benutzer aus dem Internet auf Ressourcen des privaten Netzes zugreifen und
- vertrauliche Daten aus dem privaten Netz nach außen dringen.

Dazu wird in einer Firewall definiert,

- welche Außenstehende auf welche Ressourcen des privaten Netzes zugreifen dürfen
- und welche internen Benutzer od. System auf welche Dienste des Internets zugreifen dürfen.

Dafür muss die Firewall die einzige Verbindung zw. internem Netz und Internet sein.

Mit einer Firewall können auch Adressen umgesetzt werden (Network Address Translation NAT), wenn z.B. im internen Netz keine offiziell registrierten IP-Adressen verwendet werden.

Eine Firewall alleine garantiert allerdings noch keine Sicherheit, da immer noch an der Firewall vorbei z.B. über ein Modem vertrauliche Daten nach außen gesendet werden können oder aber auch mit e-mails, falls dieser Dienst erlaubt ist. Ebenso kann bei verschlüsselten Daten die Firewall den Kommunikationsinhalt nicht prüfen.

4.4.1 Firewall-Architekturen

Packet filtering router:

Ist i.d.R. ein speziell eingerichteter Router, der für jedes IP-Paket entscheidet, ob es passieren darf oder nicht. Dafür wird anhand der Informationen aus dem IP-Paket-Header (z.B. IP-Adresse od. Port-Nr. des Senders/Empfängers) aus einer Tabelle mit Regeln eine Regel ausgewählt. Damit kann die Firewall bestimmte Dienste oder Adressen komplett sperren. Damit die Filterung schnell arbeitet, wird die Regeltabelle sequentiell abgearbeitet und die erste passende Regel angewandt. Deshalb sollten die spezielleren Regeln am Anfang der Liste stehen und dann erst die allgemeineren.

Nachteile:

- Mit der Anzahl der Filterregeln sinkt damit auch der Durchsatz des Routers.
- Ein Paketfilter erstellt i.d.R. keine Protokolle, Angriffe um den Filter zu überlisten sind also nur schwer erkennbar.
- Es können keine benutzerbezogenen Regeln erstellt werden, da im IP-Header keine Informationen über Benutzer enthalten sind.
- Die Filterentscheidung erfolgt auf Basis einzelner IP-Pakete, es kann also z.B. für einer ftp-Sitzung nicht einstellen, dass nur bestimmte Verzeichnisse erlaubt sind.

- Eingehende Bestätigungspakete (SYN-ACK, ACK) können nicht geprüft werden, ob sie tatsächlich Antworten auf einen Verbindungs-Request sind oder von einem Angreifer generiert wurden.
- Es kann auch keine Adressumsetzung durchgeführt werden, da der Verbindungszustand vom Packet-Filter nicht gespeichert wird.

Vorteile:

- Transparent für den Anwender.
- Keine zusätzliche SW auf den internen Rechnern.
- Moderne Router besitzen bereits eingebaute Packet-Filter.

Stateful inspection filter:

ist ein erweiterter Packet-Filter mit einem lokalen Gedächtnis über den aktuellen Verbindungszustand.

Application level gateway:

arbeitet auf der Ebene der Anwendungsprotokolle. Ein *Proxy*-Server läuft auf dem Gateway und fungiert als Stellvertreter zwischen Client und dem Server.

- Er baut die Verbindung nach innen auf und ist der Kommunikationspartner für Benutzer von außen.
- Er übernimmt auch Identifikation/Authentifikation der Benutzer die eine Verbindung aufbauen wollen.
- Der *Proxy* kann somit für eine Filterung der Kommunikation auf Anwendungsebene eingesetzt werden.
- Ein HTTP-*Proxy* cached auch Web-Sites und reduziert damit die Netzlast.
- Ein HTTP-*Proxy* anonymisiert den internen Benutzer gegenüber dem Internet.

Für jeden Dienst wird ein eigener *Proxy* eingerichtet, womit die Kommunikation nicht nur gefiltert sondern auch protokolliert werden kann. Dafür muss aber das IP-Routing auf dem Gateway deaktiviert werden.
Single homed: eine Netzwerkkarte, nur sinnvoll in Verbindung mit einem packet filter router.
Dual homed: zwei Netzwerkkarten.

Screened subnet:

Bezeichnet ein eigenes Teilnetz, das zwischen dem internen Netz und dem Internet liegt und an beiden Enden durch einen Filter geschützt ist: **Demilitarized Zone – DMZ**.

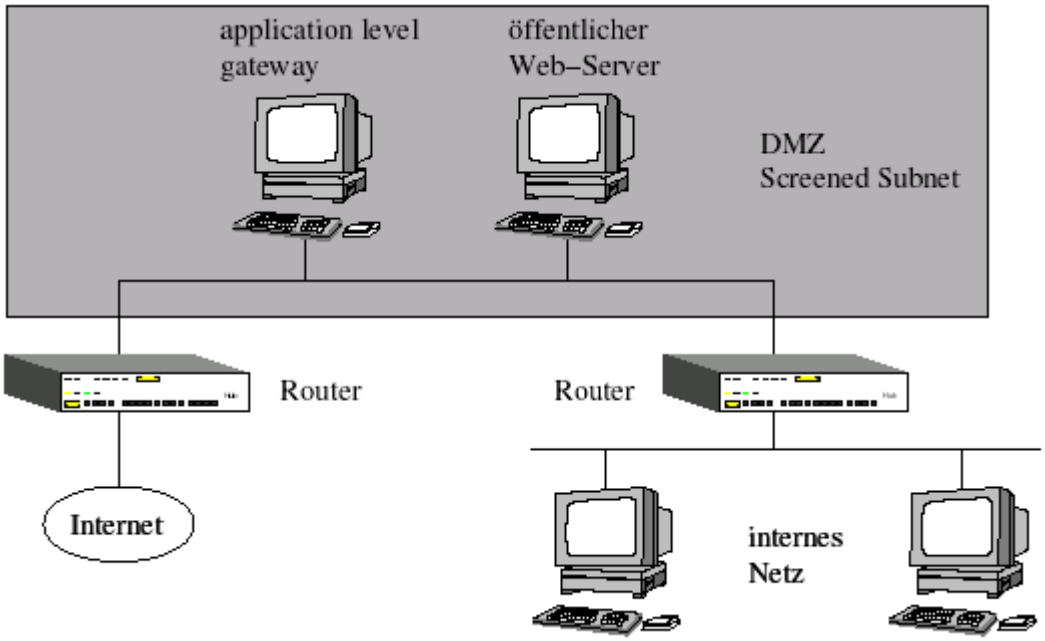


Abb. 4.1: Screened Subnet bzw. Demilitarized Zone DMZ mit single homed Gateway

Die Router sind so konfiguriert, dass alle Paket über die Rechner der *DMZ* laufen müssen.
Vorteile:

- Es erfolgt eine Filterung auf Paketebene und auf Anwendungsebene
- Die Architektur ist recht gut skalierbar (Verteilung des *Application Level Gateway* auf mehrer Rechner).
- Es können auch neue Dienste installiert werden, für die noch kein *Proxy* zur Verfügung steht. Die Router werden dann so konfiguriert, dass Pakete dieser Dienste direkt durch die *DMZ* durchgereicht werden.
- Mit einem *dual homed Application Level Gateway* kann das *DMZ* nochmals aufgesplittet werden und damit die Sicherheit erhöht werden. (Web-Server ist nicht mehr so leicht angreifbar)

4.4.2 Firewall-Konfiguration

Alles was nicht explizit verboten ist, ist erlaubt:

Es wird im wesentliche eine Liste der verbotenen Verbindungen definiert, alle anderen Verbindungen sind erlaubt.
Vorteil: Angenehm für den Benutzer
Nachteil:

- Erlaubt Freiheiten, die von Angreifer ausgenutzt werden können.
- Fehler in der Konfiguration fallen erst bei einem erfolgreichen Angriff auf, der bereits Schaden verursacht hat.

Alles was nicht explizit erlaubt ist, ist verboten:

Es wird eine Liste der erlaubten Verbindungen definiert, alle anderen Verbindungen sind verboten.
Vorteil:

- Hält neue Angriffsformen wirksam ab.
- Fehler in der Konfiguration werden schnell erkannt (Benutzer kann eine benötigte Verbindung nicht nutzen)

Nachteil: Muss ständig an Benutzerwünsche angepasst werden.

Folgende **Dienste** sollten wegen **zu hohem Risiko** komplett gesperrt werden:

- ***rlogin, rsh und rexec:*** Übertragen Passwörter in Klartext und können so konfiguriert werden, dass die Authentifikation umgangen wird. Ports 512 bis 514.
- ***NIS, NFS und RPC:*** NIS u. NFS erlauben eine einfache Konfiguration eines LAN und basieren auf RPC's – Port 111. Angriffe auf diesem Port können dazu führen, dass Passwörter ausspioniert werden oder Dateien gelesen und verändert werden. Diese Dienste sollten nur innerhalb des lokalen Netzes benutzt werden.
- ***X Windows:*** Erlaubt, dass Programm und dessen E/A auf verschiedenen Rechnern läuft. Der Server für die E/A könnte missbraucht werden, um Tastatureingaben abzufangen.

Folgend **Dienste** sind mit einem **geringeren Risiko** behaftet und sollten deshalb von der Firewall entsprechend kontrolliert werden:

- ***http:*** sollten über einen Proxy laufen.
- ***SMTP:*** sollte von Firewall nur zu den Mail-Servern durchgelassen werden.
- ***telnet:*** nur über Proxy und nur zu ausgewählten Rechnern zulassen.
- ***NNTP:*** Internet News Dienst nur zu den News-Servern zulassen.
- ***DNS:*** Firewall sollte eigenen DNS-Server bereitstellen, damit interne Rechnernamen und IP-Adressen nicht von außen einsehbar sind.

4.4.3 Firewall-Betrieb

Sicherheitsmaßnahmen müssen regelmäßig auf ihre Einhaltung überprüft werden (Sicherheit ist ein ständiger Prozess).

- ***Sicherer Admin-Zugang:*** Entweder sicherer physischer Zugang zu den Geräten selbst oder über ein eigenes sicheres Admin-Netz (z.B. mit Hilfe von *ssh*).
- ***Sicheres Speichern von KonfigDaten u. Log Files:*** z.B. KonfigDaten auf CD-ROM und LogFiles auf eigenem Logserver od. eigener Partition auf der Festplatte.
- ***Protokollierung:*** zur Erkennung von Angriffsversuchen u. zur Beweissicherung.
- ***Firewall unter Ausnahmezuständen testen:***
  - wie reagiert Firewall auf Systemabsturz,
  - sind beim Ausfall Verbindungen zum Internet noch möglich
- ***Kenntnis des „normalen" Datenverkehrs:*** Analyse des normalen Datenverkehrs u. der Log Files.

Intrusion Detection Systeme – IDS:

Sollen zeitnah und zuverlässig erkennen, ob ein Angriff vorliegt. Ereignisse, die einen Angriff kennzeichnen, müssen also im System bekannt sein.
IDS benutzen zwei Verfahren:

- ***Anomalie Erkennung:*** Das IDS kennt das „normale" Benutzerverhalten und vergleicht damit die aktuellen Protokolle. Festgestellte Abweichungen werden gemeldet.
- ***Signatur Analyse:*** IDS vergleicht die Protokolldaten mit der Signatur typischer Angriffe. (z.B. beginnen viele Angriffe mit einem Port-Scan).

Reaktion auf Zwischenfälle:

1. **Keine Panik**!
2. **Überlegen ob Sofortmaßnahmen notwendig sind:** Findet Angriff noch statt oder nicht. Einzuleitende Maßnahme hängt davon ab, wie gravierend der Angriff ist und reicht von

- man lässt alles laufen wie bisher bis zu
- man trennt das gesamte Intranet vom Internet.

3. **Beweise sichern:** Backup der betroffenen System erstellen.
4. **Problem analysieren:** Ursache, die erfolgreichen Angriff ermöglichte, ermitteln.
5. **Maßnahmen zur Problemlösung ergreifen:** (hoffentlich) gefundene Lücke im System schließen.

Falls ein Angriff erfolgreich war, sollte man davon ausgehen, dass das System soweit kompromittiert wurde, dass ein weiterer Angriff möglich ist -> System neu aufsetzen!

4.5 Organisatorische Sicherheitsmaßnahmen

4.5.1 Der IT-Sicherheitsprozeß

Vorgeschlagener Sicherheitsprozess des BSI – Bundesamt für Sicherheit in der Informationstechnik (www.bsi.de).

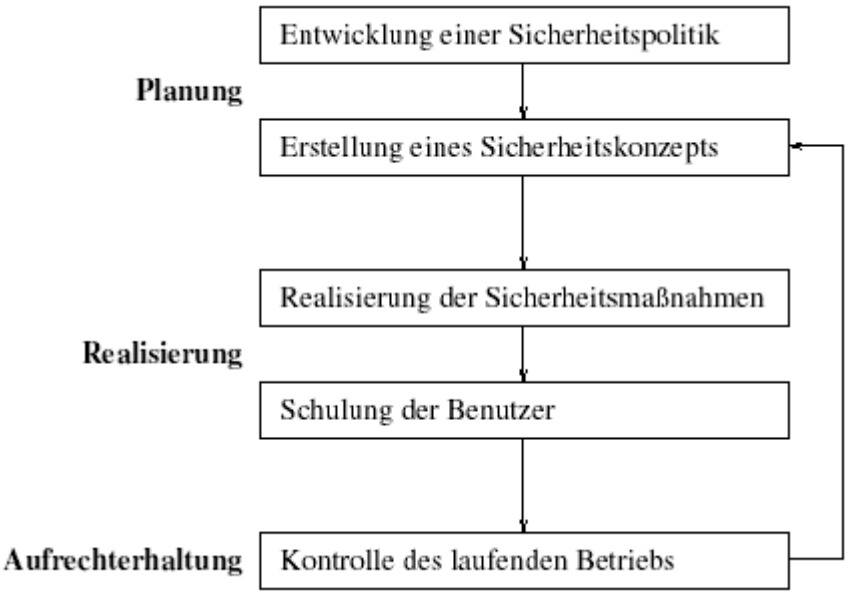


Abb. 4.2: IT-Sicherheitsprozeß des BSI

Entwicklung einer IT-Sicherheitspolitik:

Es werden die allgemeinen Ziele definiert, die bez. der IT-Sicherheit erreicht werden soll. Dabei wird definiert

- was geschützt werden soll,
- welches Sicherheitsniveau für welche Objekte angestrebt wird

Erstellung eines IT-Sicherheitskonzeptes:

Es werden die Risiken analysiert, welche die oben definierten Sicherheitsziele gefährden können. Die Bewertung der Risiken erfolgt nach den Kriterien

1. *Eintrittswahrscheinlichkeit*
2. *Ausmaß des potentiellen Schadens*

Die Ergebnisse der Bewertung können in einer probability impact Matrix visualisiert werden.

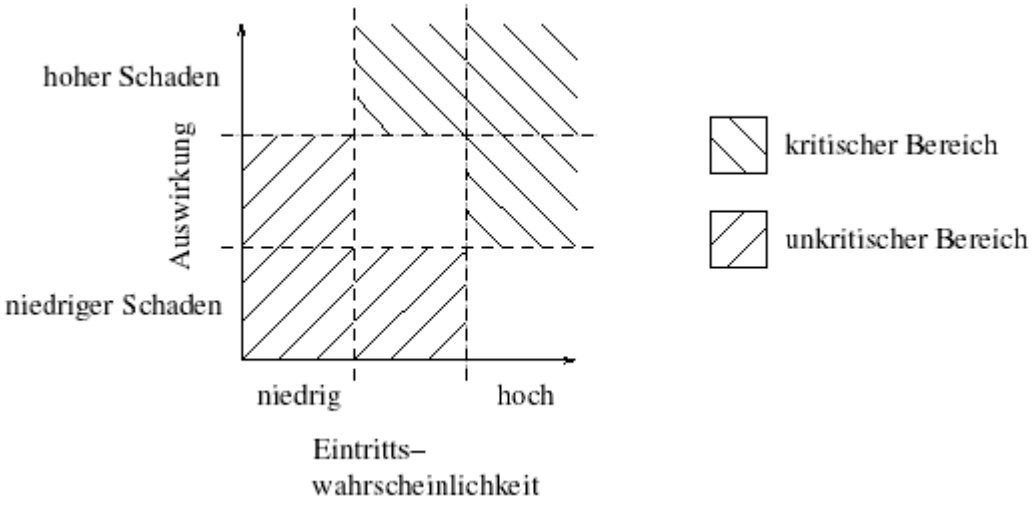


Abb. 4.3: probability impact Matrix

Außerdem bestimmen gesetzliche Regelungen wie z.B.

- KonTraG – Gesetz zur Kontrolle und Transparenz im Unternehmensbereich (Risikovorsorge für AGs)
- und die diversen Datenschutzgesetze und –richtlinien

welches Sicherheitsniveau mindestens eingehalten werden muss. Darüber hinaus ist immer der Aufwand des Sicherheitskonzept mit dem maximalen Schaden, der dadurch abgewendet werden kann, zu vergleichen.

Realisierung der Sicherheitsmaßnahmen:

Es wird ein Plan zur Realisierung der Maßnahmen erstellt, der zumindest aus folgenden Punkten besteht:

- Arbeitspläne
- Verantwortlichkeiten
- Kosten
- Termine

Schulung der Benutzer:

Ein Schulungs- und Sensibilisierungsprogramm soll die Mitarbeiter

- über die Sicherheitspolitik und –maßnahmen ***informieren***,
- durch weitergehende Erläuterungen ***motivieren***,
- mit den Maßnahmen ***vertraut m***achen
- und auf die ***Konsequenzen aufmerksam machen***, wenn gegen die Sicherheitspolitik verstoßen wird.

Diese Maßnahmen sind zeitnah zur Realisierung der Sicherheitsmaßnahmen durchzuführen.

Kontrolle des laufenden Betriebs:

Prüfen,

- ob die Sicherheitsmaßnahmen eingesetzt werden,
- ob sie korrekt eingehalten werden,
- ob sie den Anforderungen im laufenden Betrieb genügen, und
- ob sie bei Änderungen nach wie vor relevant sind oder angepasst werden müssen.

4.5.2 Eine IT-Sicherheitsorganisation

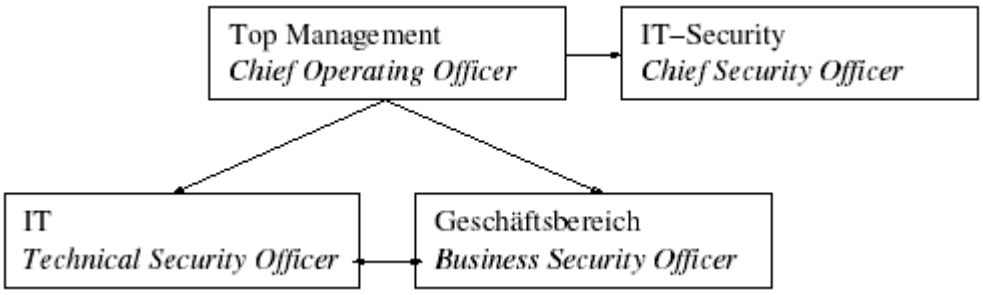


Abb. 4.4: Mögliche Organisationsstruktur für IT-Security

Top Management:

Muss sich zum Thema IT-Security bekennen, die Bedeutung akzeptieren und alle Teile der Organisation dafür sensibilisieren. Nur das Top-Management kann Veränderungen der Organisationsstruktur und der Arbeitsabläufe anstoßen.

IT-Security:

Fachliche Unterstützung des Management.

- Ausarbeitung der organisationsweiten Sicherheitspolitik.
- Miteinbeziehung neuer Entwicklungen, die Einfluss auf die Sicherheitspolitik haben. (Gesetze, Standards, Forschungsergebnisse u.ä.m.)
- Entscheidung über die erforderlichen Schulungs- u. Sensibilisierungsmaßnahmen.

Geschäftsbereiche:

Ein Geschäftsbereich kennt am besten den eigenen Schutzbedarf. Deshalb sollte eine Person des Bereichs als *Business Security Officer* benannt werden, der mit dem IT-Bereich zusammenarbeitet.

IT:

Der Technical Security Officer übernimmt die Verantwortung, dass die vom Geschäftsbereich vorgegebenen Schutzziele durch passende technische Schutzmaßnahmen auch umgesetzt werden. Allgemein hat die IT die Aufgabe, die IT-Systeme für den Geschäftsbereich zu betreiben und weiterzuentwickeln, so dass der Geschäftsbereich seine Geschäfte machen kann.

Anhang A – Prüfungsfragen Kurs 1866

Zusammengestellt aus Gedächtnisprotokollen von Studienkollegen aus dem Zeitraum Mai 2003 bis Oktober 2004. Die Gedächtnisprotokolle sind auf den Seiten des Fachschaftsrat erhältlich.

1. Welche Schutzziele gibt es? *[4x Keller] [2x Kern-Isberner]*

2. Warum sticht Verfügbarkeit heraus? Warum ist sie ein Schutzziel? *[4x Keller]*

3. Was sind konkrete Bedrohungen im Internet und welche Auswirkungen können sie haben? *[1x Kern-Isberner]*

4. Welche Netztopologien gibt es und welchen Einfluss haben sie auf die Sicherheit? *[1x Kern-Isberner]*

5. Verschlüsselungsverfahren – symmetrisch/asymmetrisch? Prinzip? Funktion? Vor- und Nachteile? *[4x Keller] [2x Kern-Isberner]*

6. Worauf basieren symmetrische Verfahren? *[1x Kern-Isberner]*

7. Was ist nötig, damit RSA funktioniert? *[1x Keller]*

8. Wie lässt sich RSA Algorithmus mathematisch herleiten? *[2x Kern-Isberner]*

9. Kann man den private key aus dem public key berechnen? *[2x Kern-Isberner]*

10. Wie sieht denn im allgemeinen das Feistelschema aus? *[1x Kern-Isberner]*

11. Wie kann man Vertraulichkeit erreichen? *[1x Keller]*

12. Warum muss überhaupt verschlüsselt werden, dass ist doch für den Nachrichtenaustausch gar nicht notwendig? *[1x Keller]*

13. Wie kann man sicher Kommunizieren/Daten übertragen? *[1x Kern-Isberner]*

14. Was ist dabei zu beachten, wenn ich ihnen symmetrisch verschlüsselt eine Nachricht zusenden möchte? *[1x Keller]*

15. Wie schützt man Integrität? *[2x Keller]*

16. Wie kann man bei einer Binärdatei erkennen, ob sie verändert wurde? *[1x Keller]*

17. Hash-Funktionen, Anforderungen an Hash? *[2x Keller]*

18. Woher weiß ich, dass der öffentliche Schlüssel wirklich von dem stammt, von dem ich es annehme? oder Wieso vertraue ich einem Public Key? *[2x Keller]*

19. Wie funktioniert die Authentifizierung im public key Verfahren? *[1x Kern-Isberner]*

20. Wie können wir sicherstellen, dass eine Email an mich wirklich von ihnen kommt? *[1x Keller]*

21. Wie wird ein Zertifikat erstellt? Was ist das überhaupt? *[1x Keller]*

22. Wer stellt Zertifikate aus? Wie erhalten sie denn so ein Zertifikat? *[2x Keller]*

23. Woher kommt die Sicherheit über die Identität eines Zertifikates? *[1x Keller]*

24. Woher wissen sie denn, dass das Zertifikat der Zertifizierungsstelle korrekt ist? *[1x Keller]*

25. Wenn sich jetzt ein Privatmann einen PC und ein Modem kauft und ins Internet möchte. Was würden sie ihm raten? oder Wie kann ich meinen privaten PC schützen? *[2x Keller][1x Kern-Isberner]*

26. Woher bekommt man aktueller Informationen zur IT-Sicherheit? *[1x Kern-Isberner]*

27. Und was ist, wenn ich auch Email empfangen möchte, deren Inhalte ich nicht kenne? Wie sieht es mit Schadensprogrammen aus? *[1x Keller]*

28. Wie schütze ich denn meine Anonymität im Internet? *[1x Keller]*

29. Was macht die Personal Firewall? *[1x Keller]*

30. Was macht eine Firewall? Welche Arten gibt es? *[2x Kern-Isberner]*

31. Firewall – Architekturen, Funktionsweise? *[1x Keller]*

32. Sie sind Sicherheitsbeauftragter in einem Unternehmen. Wie gehen sie ihre Aufgabe an? *[1x Kern-Isberner]*