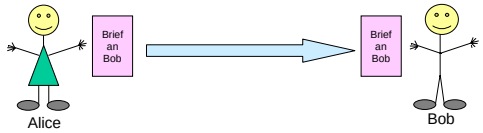
9. Transaktionssicherheit

Inhalt:

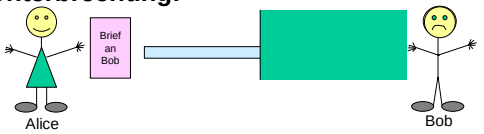
- Kommunikation und ihre Risiken
- Anforderungen an eine sichere Kommunikation
- Risiken und Realität
- Verschlüsselungsverfahren
- Signaturen und Zertifikate
- Zusammenfassung
- Literatur und Referenzen

Kommunikation und ihre Risiken

- **„normale“ Kommunikation:**



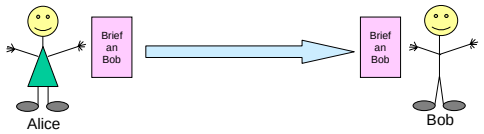
- **Sicherheitsprobleme:**
  - **Unterbrechung:**



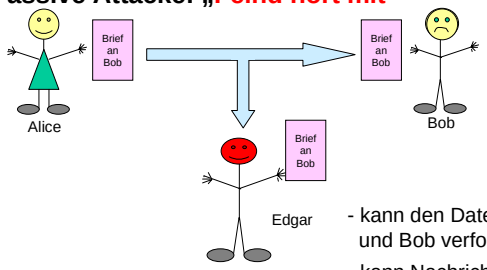
Bob erfährt nicht, daß Alice ihm eine Nachricht geschickt hat.

Kommunikation und ihre Risiken

- **„normale“ Kommunikation:**



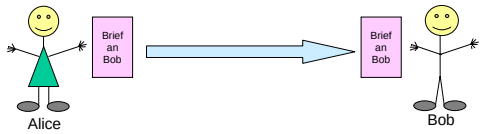
- **Sicherheitsprobleme:**
  - **Passive Attacke: „Feind hört mit“**



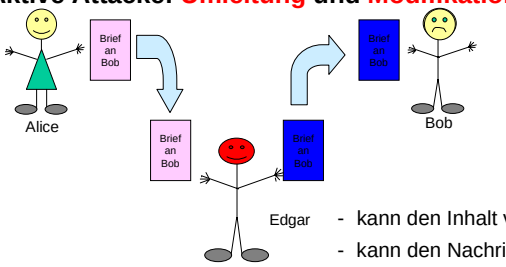
- kann den Datenverkehr zwischen Alice und Bob verfolgen
- kann Nachrichten lesen

Kommunikation und ihre Risiken

- **„normale“ Kommunikation:**



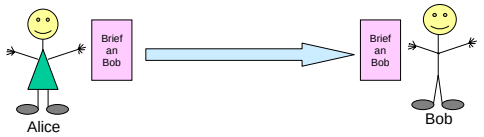
- **Sicherheitsprobleme:**
  - **Aktive Attacke: Umleitung und Modifikation**



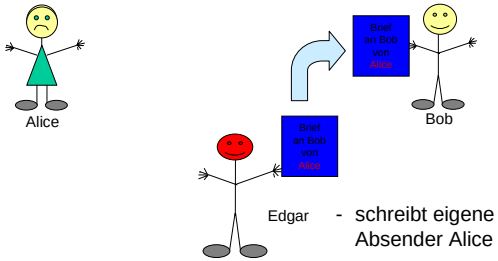
- kann den Inhalt von Nachrichten verändern
- kann den Nachrichtenfluß verändern (verzögern, umordnen)

Kommunikation und ihre Risiken

- **„normale“ Kommunikation:**



- **Sicherheitsprobleme:**
  - **Aktive Attacke: Maskerade**



Edgar - schreibt eigene Nachrichten mit Absender Alice

Anforderungen an sichere Kommunikation:

- **Schutz gegen passive Attacken durch**
  - **Zugriffskontrolle** für Dateien und Übertragungskanäle
  - **Verschlüsselung:**
    keine unbefugte Person darf Information aus Nachricht oder Datei erhalten (bzw. nur zu Kosten, die bei weitem den Wert der Information übersteigen).

- **Schutz gegen bzw. Erkennung von aktiven Attacken durch**
  - **Authentifizierung**:
    nachprüfbare Verbindung zwischen Autor/Absender und Dokument, bzw. Nachprüfung, ob bei Erstellung des Dokuments ein nur dem (angeblichen) Absender bekanntes Geheimnis verwendet wurde
  - **Integritätssicherung**:
    Garantie, daß Daten und Datenfluß nicht verändert wurden (d.h. korrekter Absender, Inhalt, Reihenfolge und Zeitstempel), bzw. Schaffung der Möglichkeit, die Korrektheit dieser Daten zu überprüfen.

Risiken und Realität

♥ Grundphilosophie des akademischen Internet:

Gutwilligkeit der Benutzer + Offenheit des Systems

Gefahren des Internet:

Hacker: an detaillierten Systemkenntnissen interessiert, können Sicherheitslücken entdecken, freimütige Wissensmitteilung, keine Datenzerstörung

Cracker: verletzen die Systemintegrität fremder Rechner, suchen unauthorisierten Zugang, zerstören wichtige Daten, verursachen Probleme im Arbeitsablauf des angegriffenen Rechners

Die Grenze zwischen beiden ist oft fließend.

Risiken und Realität: Beispiele

USA - 1997: Einrichtungen der Verteidigungsbehörden wurden voraussichtlich 250.000mal im letzten Jahr angegriffen; betroffene Abteilungen: Waffen- und Supercomputer-Forschung, Logistik, Finanzen, Beschaffung, ... (Bericht Government Accounting Office)

Dezember 1996: ein Cracker erlangt Kontrolle über eine Site der Luftwaffe der Vereinigten Staaten und ersetzte Verteidigungsstatistiken durch Pornographie

März 1997: schwedische Cracker dringen in Notruf-System in Florida ein, elf Bezirke betroffen

Risiken und Realität: Beispiele

Februar 1998: wichtige Hosts des Pentagons wurden von israelischem Teenager Ehud Tenenbaum geknackt

April 1998: eine Gruppe namens "Masters of Downloading" knackte das Defense Information System Network (DISN) und stahl Software, welche zur Satelliten-Kontrolle eingesetzt wird

September 1997: Website von Coca-Cole wird lahmgelegt

Dezember 1997: Yahoo geknackt

Juli 1997: Angriff auf StarWave, Kreditkarten-nummern von NBA-Kunden wurden abgefangen

Mai 1997: Carlos Filipe Salgado jr. besorgte sich durch ein "Packet Sniffer"-Programm 100.000 Kreditkartennummern

Risiken und Realität: Farmer-Studie

Stan Farmer benutzte 1996 SATAN (Tool zur Aufspürung von Sicherheitslücken)

2200 zufällig ausgewählte Internet-Hosts von Banken, Kreditvereinigungen und Behörden auf Anfälligkeit überprüft

1700 (65%) der Seiten erwiesen sich als anfällig gegenüber Angriffsmethoden, wie sie allgemein von Crackern benutzt werden.

Viele getestete Ziele verfügten über Firewalls und andere Sicherheitsmaßnahmen.

Risiken und Realität: Ernst&Young Studie

Ernst&Young LLP/Information Week Information-Security-Studie

befragt wurden 1998 über 4.000 EDV-Manager

- mehr als 35 % benutzen keine Tools zum Aufdecken von Eindringlingen
- mehr als 50% setzen keine Tools zur Überwachung von Internet-Verbindungen ein
- mehr als 60% haben keine schriftlichen Richtlinien über das Verhalten nach einem Sicherheitsvorfall

Risiken und Realität: Viren

Viren: eine nicht selbständige, aber sich selbst reproduzierende Programmroutine zur vom Anwender nicht kontrollierbaren Manipulation im Systembereich, an anderen Programmen oder deren Umgebung (Boot-Viren, File-Viren, Stealth-Viren (Tarnkappenviren), Polymorphe Viren, Makroviren)

Wurm: selbständiges, selbstreproduzierendes Programm, welches sich von Host zu Host ausbreitet

Trojanisches Pferd: Programme, welche neben ihrer eigentlichen (dem Anwender bekannten) Funktion noch weitere Funktionen ausführen, von denen der Anwender nichts weiß und deren Ausführung er im Normalfall auch nicht bemerkt

Risiken und Realität: Werkzeuge

Scanner: mit einem Scanner kann ein Angreifer einen Ziel-Host nach vermeintlich fehlerhaften Diensten abtasten

- Welche Dienste laufen derzeit?
- Unter welcher User-ID laufen diese Dienste?
- Werden anonyme Logins unterstützt?
- Erfordern gewisse Netzwerkdienste eine Authentifizierung

Passwort-Knacker: umgeht Passwort-Sicherheitsmaßnahmen durch Aufdeckung der Passwörter (bisherige Erfolgrate: 30%)

basieren auf:

- Wörterbuch-Dateien, Wortlistensammlungen
- Verschlüsselung mit Hilfe von DES (Data Encryption Standard)
- gängigen Transformationsregeln

Risiken und Realität: Werkzeuge

Sniffer: sind Programme oder Geräte, welche Netzwerk-Datenpakete abfangen

- können Passwörter abfangen
- können vertrauliche oder proprietäre Informationen abfangen
- können benutzt werden, Sicherheitsmaßnahmen angrenzender
- Netzwerke zu durchbrechen

Risiken und Realität

IP-Spoofing: Technik der Fälschung von Daten auf einem Netzwerk durch Vortäuschung einer falschen Absenderadresse

Denial of Service (DoS): Ziel einer DoS-Attacke ist die Abtrennung eines oder mehrerer Hosts vom Netz

E-Mail-Bomben: eine Serie von Nachrichten, welche die Mailbox überschwemmen (Variante: List-Linking)

Hoax: Falschmeldungen, beispielsweise über vermeindliche Computer-Viren

Spionage: hier speziell Industriespionage mit elektronischen Mitteln

Cyberkrieg: Problem der immer stärker wachsenden Abhängigkeit unserer Gesellschaft von elektronischen Datensysteme und der starken Verwundbarkeit dieser Systeme mit relativ einfachen Mitteln für terroristische Angriffe

Risiken und Realität: Informationsquellen

Computer Emergency Response Team (CERT): 24-Stunden Notfalldienst
Webseite mit Sicherheitsinformationen
http://www.cert.org/nav/alerts.html

Jahresbericht

Computer Incident Advisory Capability (CIAC) des US-Department of energy: CIAC-Virus-Datenbank, CIAC-Sicherheitsbulletins, CIAC-Sicherheitsdokumente http://ciac.llnl.gov/

Mailing-Listen

Usenet-Newsgruppen

Risiko: Was tun?

Das Internet ist unsicher!

- Wie können Transaktionen auf sichere Art und Weise abgewickelt werden?
- Wie kommuniziert man sicher?

Lösung: Verschlüsselung!

Verschlüsselung Grundproblem

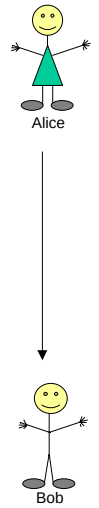
sicherer Standort
Filiale A (Alice) sendet
die Nachricht
Zahle DM 100 an C — Klartextnachricht

↓

"Zahle DM 100 an C" — ungeschützte Klartextnachricht

↓

sicherer Standort
Zentrale (Bob) empfängt
die Nachricht
Zahle DM 100 an C — Klartextnachricht



Verschlüsselung Grundproblem

Zahle DM 100 an C — Klartextnachricht

↓

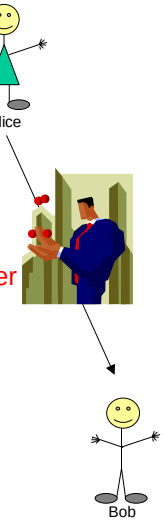
"Zahle DM 100 an C" — ungeschützte Klartextnachricht

↓

"Zahle DM 1000 an H" — Henry der Fälscher manipuliert die übertragenen Daten

↓

sicherer Standort
Zentrale (Bob) empfängt
die Nachricht
Zahle DM 1000 an H — Klartextnachricht — Was tun?



Verschlüsselung: Grundproblem

Alice schreibt die Nachricht Zahle DM 100 an C (Klartextnachricht)
und verschlüsselt diese Chiffriergerät %6§.!ku/§ah%=&
sicherer Standort sendet die chiffrierte Nachricht

↓

"%6§.!ku/§ah%=&" ungeschützte aber chiffrierte Nachricht

↓

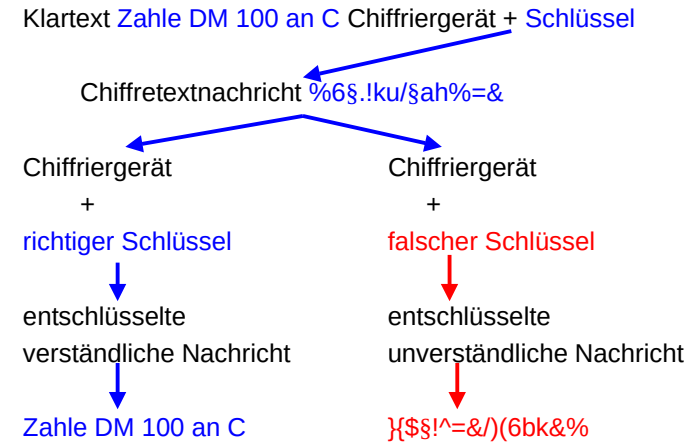
"%6§.!ku/§ah%=&" Tom der Schnüffler liest %6§.!ku/§ah%=&
aber versteht ???

↓

sicherer Standort
Chiffriergerät Bob empfängt die Nachricht
Zahle DM 100 an C (Klartextnachricht)

Verschlüsselungsverfahren bieten eine Möglichkeit zur Erhöhung der Sicherheit.

Verschlüsselung: Grundproblem



♠ Zur Entschlüsselung benötigt man: das **richtige** Chiffriergerät
und den **richtigen** Schlüssel

Ziele der Transaktionssicherheit

- wirtschaftliche Anschaffungskosten und benutzerfreundliche Sicherheitssysteme
- einfache Kommunikation mit vielen Rechnern
- direkter Internet-Zugang
- anbieten von Produkten an außenstehende über das Internet
- strenge Geheimhaltung
- strenge Authentifizierung von Nachrichten

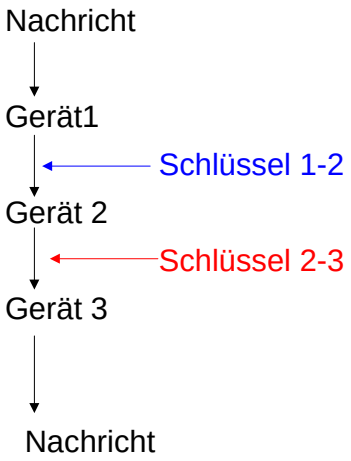
Leitungsverschlüsselung: Grundidee

Jedes Netzkoppelement (etwa i) empfängt die Nachricht mit allen Informationen, welche in den verschiedenen Headern enthalten sind, verschlüsselt die kompletten Daten inklusive der Zieladresse und leitet sie an das nächste Netzkoppelelement (i+1) auf dem Weg zum Ziel weiter.

Der dabei benutzte Schlüssel wird nur für die Leitung zwischen den Netzkoppelelementen i und i+1 benutzt.

Element i+1 entschlüsselt die Nachricht und verschlüsselt diese wieder mit dem Schlüssel für die Leitung von i+1 nach i+2. u.s.w.

Leitungsverschlüsselung: Grundidee



Leitungsverschlüsselung: Vorteile / Nachteile

Vorteile:

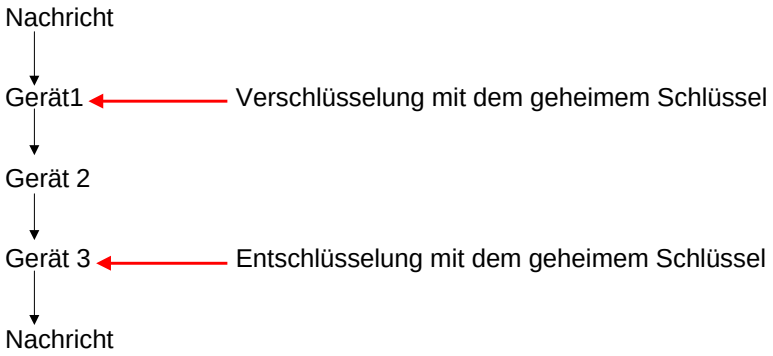
- Schlüsselverwaltung relativ einfach
- der Aufbau redundanter Wege ist leicht möglich
- wird keine sinnvolle Information übertragen, so können zufällig erzeugte Daten ausgetauscht werden

Nachteile:

- jede Leitung, welche zum Transport der Daten benutzt werden kann muß verschlüsselt sein
- am Verschlüsselungsprozeß sind oft sehr viele Geräte beteiligt

Ende-zu-Ende-Verschlüsselung

man realisiert ein Verschlüsselungsverfahren auf einer zusätzlichen Schicht zwischen Internetschicht und der Transportschicht, oder gänzlich auf der Anwendungsschicht



Ende-zu-Ende-Verschlüsselung

Vorteile:

Daten liegen nur in den beiden äußeren Geräten unverschlüsselt vor

Schlüsselaustausch nur zwischen 2 Partnern

Nachteil:

die Adressen und andere zur Weiterleitung benötigte Informationen liegen unverschlüsselt vor

Verschlüsselungsverfahren: Klassifikation

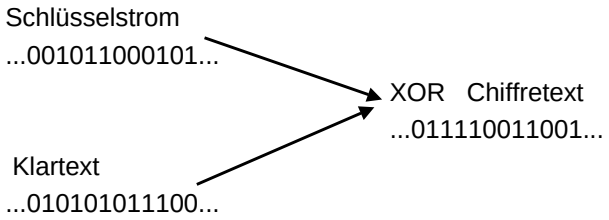
Symmetrische Verschlüsselungsverfahren / Private-Key-Verfahren

- bei diesen Verfahren wird zur Verschlüsselung und zur Entschlüsselung der gleiche Schlüssel benutzt
- der Schlüssel muss geheim gehalten werden und zwischen beiden Partnern auf sicherem Wege ausgetauscht werden

Asymmetrische Verschlüsselungsverfahren / Public-Key-Verfahren

- hierbei benutzt man zwei unterschiedliche Schlüssel, welche über ein mathematisches Verfahren voneinander abhängen
- ein Schlüssel ist in der Regel öffentlich der andere privat
- beide können zur Ver- bzw. Entschlüsselung benutzt werden

Symmetrische Verschlüsselung: z.B. Stromchiffrierung



Problem: bei sich periodisch wiederholendem Schlüsselstrom angreifbar

moderne Anwendung: Schlüsselstrom wird durch Prozedur erzeugt

als sicher geltende Methode: One-Time-Pad (Einmalblock)
Schlüsselstrom echt zufällig und nicht wiederverwendet

Beispiel einer Symmetrischen Methode für bitweise Verschlüsselung (Stromchiffrierung, z.B. RC4):

Verschlüsselung:

- Gegeben sei ein Klartext, die zu verschlüsselnde Nachricht.
- Erzeuge (Pseudo-)Zufallsfolge derselben Länge wie der Klartext.
- Bilde das bitweise XOR der beiden Bitfolgen.

```
Klartext:      01110011010111010001110101110111111011....
Zufallsfolge:  10110101100010010100001101101110101010101....
Chiffretext:   11000110110101000101111000011001011110....
Zufallsfolge:  10110101100010010100001101101110101010101....
Klartext:      01110011010111010001110101110111111011....
```

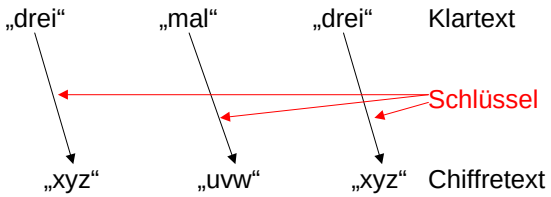
Entschlüsselung:

- Bilde erneut das bitweise XOR mit der Zufallsfolge.

Der **geheime Schlüssel** dieses Verfahrens ist die Information zur Erzeugung der (Pseudo-)Zufallsfolge.

Symmetr. Verschlüsselungsverfahren: Blockchiffrierung

Datenblöcke fester Länge werden mit einem Schlüssel fester Länge zu Chiffretextblöcken mit fester Länge verschlüsselt



Problem: derselbe Klartextblock
+ derselbe Schlüssel ➔ derselbe Chiffretext
➔ cut-and-past-Angriff möglich!

Chiffriermodus: grundsätzliche Art der Vorgehensweise, um dieses und andere Probleme zu vermeiden

Blockchiffrierung: Electronic-Codebook-Modus (ECB)

Vorgehen: Verschlüsselung wird blockweise auf den Klartext angewendet
eventuell Auffüllung eines Blocks (Padding) nötig, da nur vollständige Blöcke verschlüsselt werden können

Sicherheit:

- Muster im Klartext werden nicht verborgen.
- Die Eingabe für die Blockchiffrierung wird nicht randomisiert, sondern ist mit dem Klartext identisch.
+ Mit dem gleichen Schlüssel kann mehr als eine Nachricht chiffriert werden.
- Der Klartext kann leicht manipuliert werden; man kann Blöcke entfernen, wiederholen oder vertauschen

Blockchiffrierung: Electronic-Codebook-Modus

Effizienz:

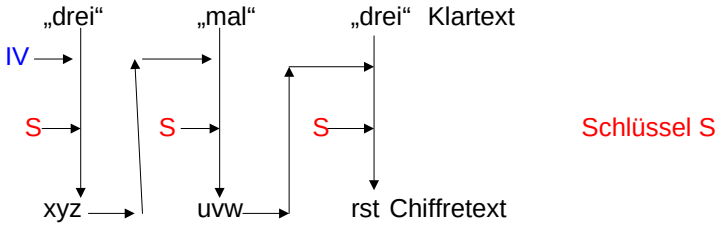
+ Die Geschwindigkeit entspricht der der Blockchiffrierung.
- Durch das Auffüllen wird der Chiffretext um bis zu einen Block länger als der Klartext.
- Es sind keine Vorausberechnungen möglich.
+ Die Verarbeitung ist parallelisierbar.

Ausfallsicherheit:

- Ein Fehler im Chiffretext betrifft einen kompletten Klartextblock.
- Synchronisierungsfehler können nicht behoben werden.

Blockchiffrierung: Cipher Block Chaining (CBC)

- Jeder Klartextblock wird vor der eigentlichen Verschlüsselung bitweise mit einem Chiffretextblock XOR-verknüpft.
- Dabei werden die Klartextdaten nicht sofort verschlüsselt, sondern zuerst mit einem zufällig erscheinenden Chiffretext verknüpft.



- Man beginnt dabei mit einem zufällig gewählten Block (Initialisierungsvektor IV) von Initialisierungsbits.
- Die nachfolgenden Blöcke werden durch Chaining miteinander verknüpft.

Blockchiffrierung: Cipher Block Chaining (CBC)

Sicherheit:

+ Muster im Klartext werden durch XOR-Verknüpfung mit dem vorhergehenden Chiffretextblock verborgen.
+ Die Eingabe für die Blockchiffrierung wird durch XOR-Verknüpfung mit dem vorhergehenden Chiffretextblock randomisiert.
+ Mit dem gleichen Schlüssel kann mehr als eine Nachricht chiffriert werden.
+/- Manipulation des Klartextes ist etwas umständlich; Blöcke können z.B. am Ende der Nachricht entfernt werden; Wiederholung ermöglicht einige kontrollierte Änderungen.

Blockchiffrierung: Cipher Block Chaining (CBC)

Effizienz:

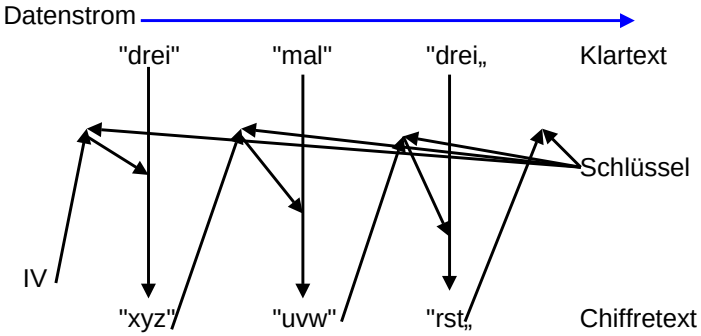
+ Die Geschwindigkeit entspricht der der Blockchiffrierung.
- Der Chiffretext wird bis zu einem Block länger als der Klartext (IV nicht mitgezählt).
- Es sind keine Vorausberechnungen möglich.
+/- Verschlüsselung ist nicht parallelisierbar; Entschlüsselung ist parallelisierbar und erlaubt wahlfreien Zugriff.

Ausfallsicherheit:

- Ein Fehler im Chiffretext betrifft einen kompletten Klartextblock sowie das entsprechende Bit im nächsten Block.
- Synchronisierungsfehler können nicht behoben werden.

Blockchiffrierung Cipher-Feedback-Modus (CFB)

- ähnelt CBC-Modus - auch hier "Rückkopplung„ des Chiffretextblocks
- Klartext wird nicht sofort verschlüsselt
- zuerst wird aus dem vorangehenden Block, bzw. dem Initialisierungsvektor (IV), ein ständig wechselnder neuer Schlüssel erzeugt



Blockchiffrierung Cipher-Feedback-Modus (CFB)

Sicherheit:

+ Muster im Klartext werden verborgen.

+ Die Eingabe für die Blockchiffrierung wird randomisiert.

+ Mit dem gleichen Schlüssel kann mehr als eine Nachricht chiffriert werden, falls unterschiedliche IV‘s benutzt werden.

+/- Manipulation des Klartextes ist etwas umständlich; Blöcke können z.B. am Ende der Nachricht entfernt werden; Wiederholung ermöglicht einige kontrollierte Änderungen

Blockchiffrierung Cipher-Feedback-Modus (CFB)

Effizienz:

+ Der Chiffretext ist genauso lang wie der Klartext (IV nicht mitgezählt).

+/- Verschlüsselung ist nicht parallelisierbar, Entschlüsselung ist parallelisierbar und erlaubt wahlfreien Zugriff.

- Einige Vorausberechnungen sind möglich, bevor ein Block ankommt. Der vorherige Chiffretextblock kann verschlüsselt werden.

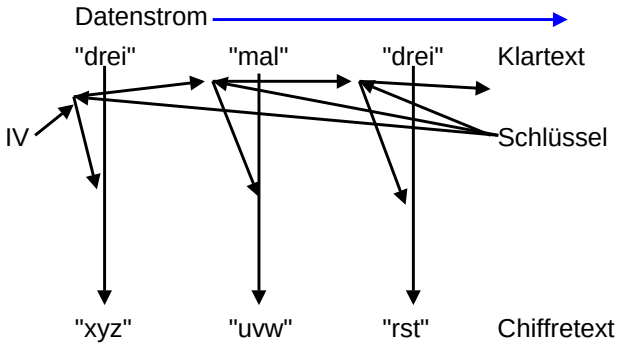
Ausfallsicherheit:

- Ein Chiffretextfehler wirkt sich auf das entsprechende Bit des Klartexts und den nächsten vollständigen Block aus.

+ Synchronisierungsfehler voller Blocklänge können behoben werden.

Blockchiffrierung Output-Feedback-Modus (OFB)

- ähnlich wie CFB-Modus, aber einfacher
- erzeugt die wechselnden Schlüssel nur mit der Blockchiffrierung
- Schlüsselstrom hängt nicht vom Datenstrom ab



Blockchiffrierung Output-Feedback-Modus (OFB)

Sicherheit:

+ Muster im Klartext werden verborgen.
+ Die Eingabe für die Blockchiffrierung wird randomisiert.
+ Mit dem gleichen Schlüssel kann mehr als eine Nachricht chiffriert werden, falls unterschiedliche IVs benutzt werden.
- Manipulation des Klartextes ist sehr einfach; Jede Änderung des Chiffretextes beeinflusst direkt den Klartext.

Blockchiffrierung Output-Feedback-Modus (OFB)

Effizienz:

+ Die Geschwindigkeit entspricht der der Blockchiffrierung.
+ Der Chiffretext ist genauso lang wie der Klartext (IV nicht mitgezählt)
+ Vorausberechnungen sind möglich, bevor eine Nachricht ankommt.
-/+ OFB-Verarbeitung ist nicht parallelisierbar.

Ausfallsicherheit:

- Ein Chiffretextfehler wirkt sich nur auf das entsprechende Bit des Klartexts aus.
- Synchronisierungsfehler können nicht behoben werden.

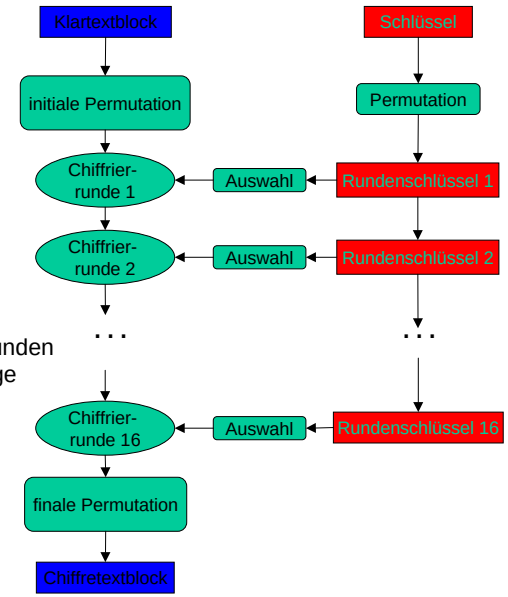
Standardmethoden für symmetrische Blockchiffrierung:

- **DES** = **D**ata **E**ncryption **S**tandard
  - fortlaufende Verschlüsselung von 64 Bit Blöcken mit 56 Bit Schlüssel
  - entwickelt von IBM gemeinsam mit der NSA (National Security Agency)
  - seit 1977 empfohlen als „ausreichend sicherer" Verschlüsselungsstandard für die USA (außer für "streng geheime" Information) (nicht mehr sicher, über einen Nachfolger wird zur Zeit beraten)
- **Triple-DES**:
  - Mehrfachverschlüsselung mit DES, Schlüssellänge 112 oder 168 Bit
- **IDEA** = **I**nternational **D**ata **E**ncryption **A**lgorithm
  - entwickelt an der ETH Zürich 1990 - 1992
  - Blocklänge 64 Bit, Schlüssellänge 128 Bit
- **CAST** :
  - entwickelt in Kanada, Schlüssellänge 64 Bit oder 128 Bit
  - stark verwandt mit DES, aber erheblich sicherer.

IDEA, **Triple-DES** und **CAST** werden in **PGP** verwendet (**P**retty **G**ood **P**rivacy), u.a. dadurch sehr weit verbreitet (http://www.pgpi.com).

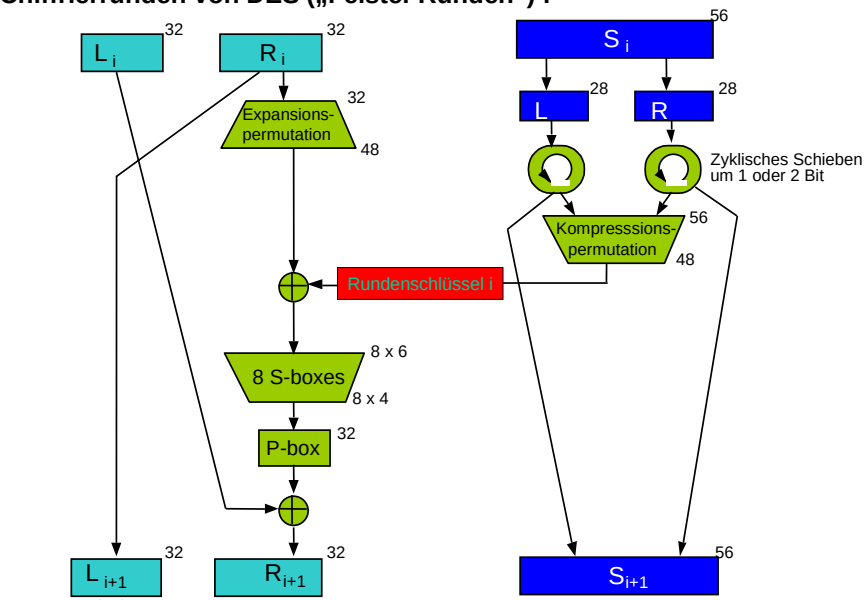
Details von DES

- **Grobstruktur:**

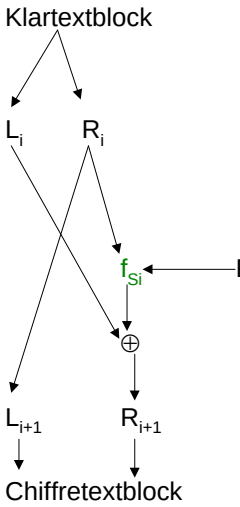


- Entschlüsselung durch Anwendung der Chiffrierrunden in umgekehrter Reihenfolge

Chiffrierrunden von DES („Feistel-Runden“) :



DES (Feistel-Netzwerk)



Li und Ri sind Halbblöcke der Eingabe

$L_{i+1} = R_i$

$R_{i+1} = L_i + f_{Si}(R_i)$

Wer f_{Si} kennt kann dechiffrieren:

$L_i = L_i + f_{Si}(R_i) + f_{Si}(R_i)$

$= R_{i+1} + f_{Si}(R_i);$

$R_i = L_{i+1}$

Bemerkungen zu DES:

- Expansionspermutation und P-Box bewirken **Diffusion**seffekt (auch ***Lawineneffekt***):
  Jedes Bit des Schlüssels und des Klartexts beeinflußt möglichst viele Bits des Chiffretexts.
- Durch S-Boxes immun gegen differentielle Kryptanalyse.
  (**Konfusion**seffekt, **CAST** verwendet für jede Anwendung spezielle S-Boxes.)
- **Sehr gut geeignet für Implementierung in Hardware (Smartcards!)**

Sicherheit von DES:

- Nur intelligente „Brute Force“-Angriffe möglich ⇒ sehr rechenaufwendig,
  aber mit genügend Hardware-Aufwand relativ schnell ausführbar - **wenige Stunden**!
  (DES-cracker der Electronic Frontier Foundation - http://www.eff.org).
- Gefährlichste Angriffe durch
  - direkten Zugang zu Schlüsseln
  - gezielte Veränderung des Verfahrensablaufs durch externe Manipulation
  - genaue Zeitmessungen, Stromverbrauchsmessungen bei Smartcards

⇒ **DES sollte man heute nicht mehr einsetzen,**
statt dessen Triple-DES, CAST oder IDEA

Asymmetrische Verschlüsselung: „Public-Key“ Kryptosysteme

- Jede Person hat ihren eigenen **öffentlichen Schlüssel P** zum Verschlüsseln von Nachrichten / Dateien (verfügbar in öffentlichem „Schlüsselbuch“).
- Jede Person hat ihren eigenen **geheimen Schlüssel S** zum Entschlüsseln.
- Sei M eine Nachricht
  C=P(M) mit P verschlüsselte Nachricht
  M'=S(C) mit S entschlüsselte Nachricht

Anforderungen:

(i) M'=S(P(M))=M für jede Nachricht M.

(ii) S aus P abzuleiten ist genauso schwierig wie P(M) ohne Kenntnis von S zu entschlüsseln.

(iii) S und P lassen sich leicht erzeugen.

(iv) Verschlüsselung und Entschlüsselung sind nicht zu aufwendig.

Standardverfahren: **Diffie-Hellman, RSA, ElGamal**

(ebenfalls eingesetzt in PGP)

Asymmetrische Verfahren (Public-Key-Verfahren)

Algorithmus von Diffie und Hellman (1976)

Kommunikationspartner A und B wollen vertrauliche Nachrichten austauschen.

Hierzu muss ein geheimer Sitzungsschlüssel S vereinbart werden, mit welchem die Nachrichten unter Benutzung eines symmetrischen Verfahrens verschlüsselt werden sollen.

Zum Austausch des Sitzungsschlüssels S wird das nachfolgende Protokoll vereinbart:

Asymmetrische Verfahren: Diffie und Hellman

(1) A und B vereinbaren eine große Primzahl n und eine Zahl g mit $1 < g < n$ (n und g nicht geheim).

(2) A wählt eine geheime große ganze Zahl x und berechnet:

$$Z_A = g^x \bmod n.$$

(3) B wählt eine geheime große ganze Zahl y und berechnet:

$$Z_B = g^y \bmod n.$$

(4) A und B tauschen die Ergebnisse Z_A und Z_B aus.

(5) beide berechnen den Sitzungsschlüssel S:

A Berechnet: $S = (Z_B)^x \bmod n = g^{y*x} \bmod n$

B Berechnet: $S = (Z_A)^y \bmod n = g^{x*y} \bmod n$

Dieser Sitzungsschlüssel kann nun für das beabsichtigte symmetrischen Verschlüsselungsverfahren benutzt werden.

Asymmetrische Verfahren: RSA [Rivest, Shamir, Adleman 1978]

Grundidee:

(1) A wählt zwei geheimzuhaltende große Primzahlen p und q, von deutlich unterschiedlicher Größe, und berechnet $n = p*q$.

(2) A wählt einen öffentlichen Schlüssel $e > 1$ so,

dass e und $(p-1)*(q-1)$ teilerfremd sind

(3) A berechnet seinen geheimen Schlüssel d durch:

$$d = e^{-1} \pmod{(p-1)*(q-1)}.$$

(4) Chiffrierung: Will B nun an A einen verschlüsselten Text T versenden, so verwendet er den öffentlichen Schlüssel e von A und berechnet:

$$G = T^e \bmod n$$

(5) Dechiffrierung: Um die Nachricht lesbar zu machen benutzt A ihren geheimen Schlüssel d:

$$T = G^d \bmod n$$

RSA-Kryptosystem Vorgehen:

- öffentlicher Schlüssel P: Paar ganzer Zahlen (N,e)
  (N ist Produkt zweier zufällig gewählter Primzahlen p,q)
- geheimer Schlüssel S: Paar ganzer Zahlen (N,d) *(nur d ist geheim)*
- e und d haben höchstens so viele Bits wie N (evtl. erheblich weniger!).
- Standardlängen von N: 512, 1024 oder 2048 Bit
  (im Prinzip beliebige Länge möglich)

RSA Verschlüsselung:

1) Teile Nachricht M in Folge $m_1 \ldots m_k$ von Zahlen (Bitblöcken) kleiner als N.
2) Berechne $c_i := P(m_i) = m_i^e \bmod N$.

RSA Entschlüsselung:

1) Teile Nachricht C in Folge $c_1 \ldots c_k$ von Zahlen kleiner als N.
2) Berechne $m_i := S(c_i) = c_i^d \bmod N$.

RSA-Kryptosystem

- **Korrektheit** des Verfahrens basiert auf Sätzen der Zahlentheorie.
- **Sicherheit** des Verfahrens beruht auf Schwierigkeit der Faktorisierung von N (d.h. der Bestimmung von p und q bei alleiniger Kenntnis von N).

Einzelheiten siehe Literatur bzw. H. Schmeck: Algorithms for Internet-Applications Vorlesung wird auch im WS 01/02 angeboten

RSA-Kryptosystem

- Problem: Erzeugung großer Primzahlen
  probabilistischer Ansatz: Test von Miller-Rabin
- RSA- ist in den USA patentiert.
- RSA ist De-Facto-Standard in weiten Teilen der Welt.
- Es existieren verschiedene Hardware-Implementierungen von RSA
- Geschwindigkeit: wesentlich langsamer als DES

RSA-Kryptosystem

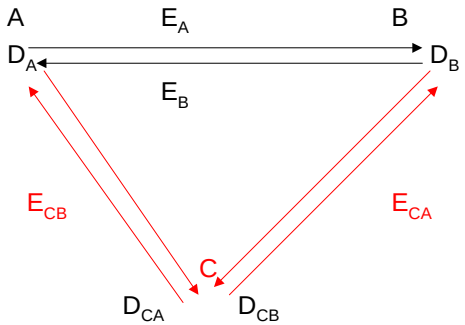
- **Sicherheit von RSA:** Es wurde bisher nicht mathematisch bewiesen, daß n faktorisiert werden muss, um T aus G und e zu berechnen.

verschiedene Risiken existieren:

- ♦ gleiche Primzahlen in verschiedenen Moduln n
- ♦ Angriff mit ausgewähltem Geheimtext
- ♦ Angriff gegen kleine Werte von e
- ♦ Angriffe bei gemeinsamen Moduln
- ♦ neue Methoden bei der Faktorisierung großer Zahlen
- ♣ Diese ersten vier Risiken lassen sich durch geeignete Implementierung ausschließen.

Asymmetrische Verfahren (Public-Key-Verfahren)

Man-in-the-Middle-Angriff



Angreifer C unterbricht die Verbindungng, fängt beide an und ersetzt diese durch zwei von ihm erzeugte Schlüssel E_{CA} und E_{CB}.

Benutzt A E_{CB} und B E_{CA}, so kann C mitlesen.

D_A, D_B private Schlüssel von A bzw. B
E_A öffentlicher Schlüssel von A
E_B öffentlicher Schlüssel von B

Asymmetrische Verfahren (Public-Key-Verfahren)

➔ sicherer Schlüsselaustausch wichtig!

persönlich oder vertrauenswürdige Person

bzw. Institution (Zertifizierung)

Anwendung des RSA-Algorithmus:

Verschlüsselung von Nachrichten:

Vorteile:

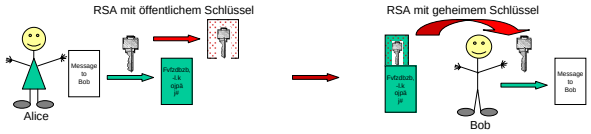
- Asymmetrie vermeidet das Problem des sicheren Austauschs von Schlüsseln (allerdings bleibt das Problem der sicheren Aufbewahrung des geheimen Schlüssels)
- Public-key System erlaubt jedem den Versand verschlüsselter Nachrichten ohne vorherige Kommunikation mit dem Empfänger.
- Sichere Kommunikation zwischen k Partnern erfordert nur k Schlüsselpaare *(verglichen mit k(k-1)/2 für symmetrische Verfahren).*

Nachteile:

- hoher Berechnungsaufwand für Ver- und Entschlüsselung
- geringe Sicherheit, falls der Angreifer die Menge möglicher Klartexte kennt
  (z.B: Kaufe Siemens.
  Verkaufe Daimler.
  Halte Deutsche Bank.
  Angreifer könnte alle möglichen Ratschläge/Klartexte selber verschlüsseln.)

Weitere Anwendung:
Sicherer Schlüsselaustausch für symmetrische Verfahren:

- Alice und Bob
  - wählen für jeden Austausch von Nachrichten einen neuen **Sitzungsschlüssel**,
  - verschlüsseln ihre Nachrichten mit einem ausreichend sicheren symmetrischen Verfahren,
  - verschlüsseln den Sitzungsschlüssel mit dem öffentlichen Schlüssel des Empfängers mit einem asymmetrischen Verfahren (z.B. RSA) und senden die verschlüsselte Nachricht zusammen mit dem verschlüsselten Sitzungsschlüssel *(dies entspricht der Verwendung eines versiegelten Umschlags).*
  - Nur mit dem geheimen Schlüssel des Empfängers kann der Sitzungsschlüssel wiedergewonnen und damit die Nachricht entschlüsselt werden.



⇒
- Für kurze Sitzungsschlüssel sind die hohen Kosten von RSA tolerierbar.
- Jeder Schlüssel des symmetrischen Verfahrens wird nur einmal verwendet.
- **Aber: Aktiver Angriff durch Auswechseln der Nachricht möglich!**

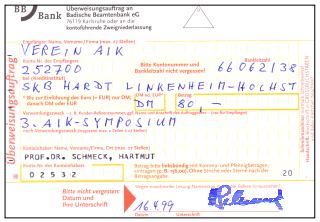
Problem: **Gültigkeit des öffentlichen Schlüssels und des Inhalts der Nachricht**.

Digitale Unterschriften

Beispiel:

- Alice sendet Bob einen (elektronischen) Scheck über $ 5,000.
- Bob ändert den Betrag in $ 50,000 und legt den Scheck der Bank vor.
- Wie kann Alice der Bank ermöglichen, die Gültigkeit des Schecks zu überprüfen?

Übliche Wirkung einer Unterschrift:



- Ohne Unterschrift ist ein Dokument wertlos.
- Die Unterschrift
  - bestätigt die Gültigkeit des Dokuments,
  - verbindet den Unterzeichner mit dem Dokument.
- Jede (sichtbare) Änderung des Dokuments macht es ungültig.

⇒ **Digitale Unterschrift muß diese Funktionen erfüllen.**

Übliches Verfahren für digitale Signatur:

- Berechne Hashwert (***message digest***) D=f(M) mit **Einweg-Hashfunktion** f.
- Signiere D durch Verschlüsselung mit geheimem Schlüssel S, d.h. berechne **σ** =S(f(M)).
- Empfänger überprüft erhaltenes Dokument M' durch Berechnung von f(M') und Vergleich mit P(**σ**)=f(M) (d.h. unter Verwendung des öffentlichen Schlüssels P).

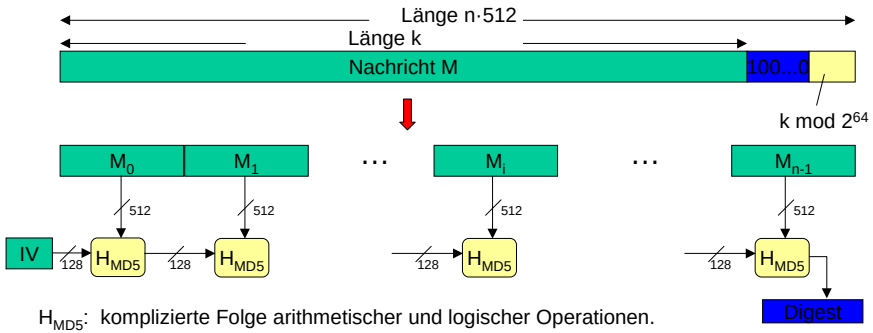
Einweg-Hashfunktionen

- Hashfunktionen:
  Abbildung einer Menge von (Schlüsseln von) Datensätzen auf Tabellenadressen (s. Dateiorganisation, Suchverfahren)
- Anforderungen:
  - Beliebige (zufällige) Mengen von Schlüsseln sollen möglichst gleichmäßig über die Tabelle verteilt werden.
  - Benachbarte Schlüssel sollen möglichst nicht auf benachbarte Adressen abgebildet werden, d.h. kleine Änderungen im Schlüssel sollen große Änderungen in der Adresse bewirken.
- Zusätzliche Anforderung bei Einweg-Hashfunktionen:
  Es soll unmöglich sein, aus einer Tabellenadresse zu berechnen, welche Schlüssel durch die (bekannte) Hashfunktion auf diese Adresse abgebildet werden.
- Schwächere Anforderung:
  Die Kosten der Berechnung der inversen Hashfunktion müssen den Wert der dadurch gewonnenen Information erheblich übersteigen.

Häufig verwendete Einweg- Hashfunktion:

- **MD5**: *Message Digest Function*

1. Erweitere M durch eine Folge 10...0 bis auf eine Länge kongruent 448 (mod 512).
2. Füge die Länge hinzu (weitere 64 Bit).
3. Berechne den 128-Bit Hashwert wie folgt:



H_{MD5}: komplizierte Folge arithmetischer und logischer Operationen.

IV : Initialisierungsvektor

- **Alternativen: Secure Hash Algorithm - SHA** oder **RIPE-MD160,** erzeugen 160 Bit

Gültigkeit öffentlicher Schlüssel

- Empfänger einer digitalen Signatur muß den gültigen öffentlichen Schlüssel des Absenders kennen, um die Signatur überprüfen zu können.
- Gültigkeit wird zugesichert durch ein **Zertifikat** ausgestellt durch
  - eine vertrauenswürdige Person (PGP - **Web of Trust**)
  - eine Zertifizierungsstelle (gemäß **Signaturgesetz**)
- Ausstellung von Zertifikaten und digitalen Signaturen in Deutschland rechtsverbindlich geregelt durch das Deutsche Signaturgesetz (1.8.1997), es wird zur Zeit an Europäische Vereinbarungen angepasst.

Auszug:

§ 2 (3):

Ein Zertifikat im Sinne dieses Gesetzes ist eine mit einer digitalen Signatur versehene digitale Bescheinigung über die Zuordnung eines öffentlichen Signaturschlüssels zu einer natürlichen Person (Signaturschlüssel-Zertifikat) oder eine gesonderte digitale Bescheinigung, die unter eindeutiger Bezugnahme auf ein Signaturschlüssel-Zertifikat weitere Angaben enthält (Attribut-Zertifikat).

Inhalt von Zertifikaten:

§ 7 (1):

Das Signaturschlüssel-Zertifikat muß folgende Angaben enthalten:

1. *den Namen des Signaturschlüssel-Inhabers, der im Falle einer Verwechslungsmöglichkeit mit einem Zusatz zu versehen ist, oder ein dem Signaturschlüssel-Inhaber zugeordnetes unverwechselbares Pseudonym, das als solches kenntlich sein muß,*
2. *den zugeordneten öffentlichen Signaturschlüssel,*
3. *die Bezeichnung der Algorithmen, mit denen der öffentliche Schlüssel des Signaturschlüssel-Inhabers sowie der öffentliche Schlüssel der Zertifizierungsstelle benutzt werden kann,*
4. *die laufende Nummer des Zertifikates,*
5. *Beginn und Ende der Gültigkeit des Zertifikates,*
6. *den Namen der Zertifizierungsstelle und*
7. *Angaben, ob die Nutzung des Signaturschlüssels auf bestimmte Anwendungen nach Art und Umfang beschränkt ist.*

Dies entspricht den Vorschriften des internationalen X.509 - Standards für Zertifikate.

Das Zertifikat wird von der Zertifizierungsstelle digital signiert.

Austausch von Nachrichten nach Signaturverordnung

(1) A erzeugt einen geheimen Schlüssel DA und einen öffentlichen Schlüssel EA

(2) A sucht sich eine Zertifizierungsstelle (ZS) seines Vertrauens und übergibt dieser Stelle seinen öffentlichen Schlüssel.

ZS überprüft, ob der Schlüssel mit einem zugelassenen Verfahren erzeugt wurde.

Wahlweise kann ZS auch das Schlüsselpaar erzeugen (darf geheimen Schlüssel nicht speichern).

(3) ZS überprüft die Identität von A und klärt A über die richtige Nutzung und die Risiken bei der Erzeugung digitaler Signaturen auf.

(4)ZS erzeugt Zertifikat (Angaben siehe unten) für den öffentlichen Schlüssel von A.

Austausch von Nachrichten nach Signaturverordnung

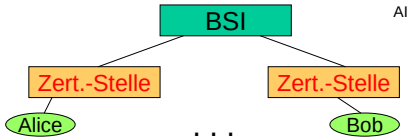
(5) ZS erzeugt eine Signatur von diesem Zertifikat.

Der zugehörige öffentliche Schlüssel wird zusammen mit dem von A (EA) in einem Verzeichnis abgelegt. Auf dieses hat B Zugriff. A erhält ebenfalls den öffentlichen Schlüssel der ZS.

(6) A erzeugt eine digitale Signatur für ein zu versendendes Dokument und verschickt Dokument zusammen mit der Signatur und seinem Zertifikat an B.

(7) B prüft die digitale Signatur des Dokuments. Dazu kann der öffentliche Schlüssel der ZS benutzt werden

Zertifizierungshierarchie:



- **Root Certification Authority**:
  Bundesamt für Sicherheit in der Informationstechnik
  Der öffentliche Schlüssel des BSI muß für jeden vertrauenswürdig verfügbar sein.
- Root CA genehmigt und zertifiziert **untergeordnete Zertifizierungsstellen**, d.h. der öffentliche Schlüssel einer Zertifizierungsstelle wird durch ein Zertifikat der Root CA abgesichert.
- Zertifizierungsstelle stellt für "**Endanwender**" (Alice und Bob) Zertifikate aus.
- Bob kann aus dem Zertifikat für den öffentlichen Schlüssel von Alice
  - den öffentlichen Schlüssel entnehmen,
  - die Gültigkeit des öffentlichen Schlüssels anhand der digitalen Signatur der Zertifizierungsstelle überprüfen,
  - die Gültigkeit des dafür erforderlichen öffentlichen Schlüssels der Zertifizierungsstelle wiederum anhand des dazugehörigen Zertifikats überprüfen, das die Root-CA signiert hat.
  - Die Gültigkeit des öffentlichen Schlüssels der Root-CA muß auf anderem Wege überprüft werden.

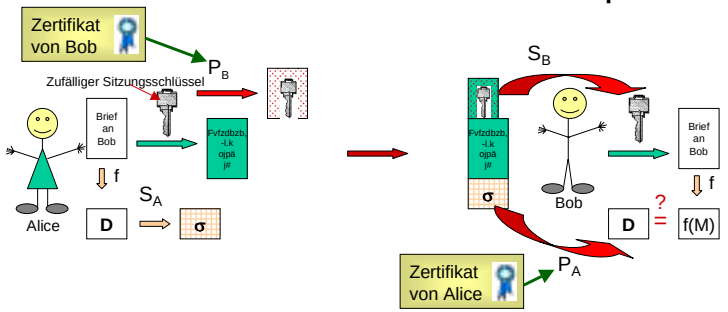
Andere Arten der Zertifizierung:

- Keine einzige Root-CA, sondern mehrere, die sich gegenseitig zertifizieren (**Wald von "Zertifizierungsbäumen"**).
  Dies entspricht der global vorhandenen Situation, in der es mehrere nationale Zertifizierungshierarchien gibt.
- Im **Web-of-Trust** des PGP-Systems:
  - an der Kommunikation beteiligte Personen stellen sich gegenseitig Zertifikate für ihre öffentlichen Schlüssel aus.
  - Vertrauenswürdigkeit eines öffentlichen Schlüssels hängt davon ab, wie viele Zertifikate vorliegen und für wie vertrauenswürdig diese Zertifikate eingeschätzt werden.

Vorteile:
- Benötigt keinerlei staatlich kontrollierte Zertifizierungsstellen.
- Vollständig selbstorganisierend.

Nachteile:
- Risiko der Verwendung ungültiger Zertifikate
- keine rechtliche Absicherung bei der Verwendung von Signaturen auf der der Basis des Web-of-Trust

Einsatz der Bausteine für sicheres Kommunikationsprotokoll



Unter der Annahme zuverlässig zertifizierter öffentlicher Schlüssel P_A und P_B liefert dieses Protokoll

- **Authentizität**: erfordert die Verwendung der geheimen Schlüssel S_A und S_B
- **Integrität**: digitale Signatur erfordert die Verwendung des geheimen Schlüssels S_A
- **Vertraulichkeit**: Sitzungsschlüssel k_S wird nur durch Verwendung des geheimen Schlüssels S_B verfügbar.
- **Verbindlichkeit**: Alice kann die Verwendung ihres geheimen Schlüssels nicht abstreiten

Standardprotokolle für sichere Kommunikation

- **Secure Socket Layer (SSL)**
  - entwickelt von Netscape (1994+...)
  - erweitert worden zu "**Transport Layer Security**" **(TLS)** *(siehe ftp://ftp.isi.edu/in-notes/rfc2246.txt).*
  - Ermöglicht sichere Kommunikation über unsichere Kanäle (oberhalb von TCP/IP) durch
    - Authentisierung (Client-Server)
    - digitale Signaturen
    - Verschlüsselung der übertragenen Daten mit Sitzungsschlüsseln.
  - Verwendet die übliche Kombination asymmetrischer und symmetrischer kryptographischer Verfahren (RSA, Diffie-Hellmann, ,,,; RC2, RC4,...;MD5, SHA).
  - Client/ Server verständigen sich zu Beginn über das stärkste gemeinsam verfügbare Verfahren.
  - Liefert keinerlei zusätzliche Sicherheit auf der Ebene der Anwendungsprogramme.

Secure Electronic Transactions - SET

Von IBM, VISA und Mastercard entwickeltes Protokoll für den sicheren Austausch von Informationen bei der Abwicklung von Geschäftstransaktionen unter Verwendung von Kreditkarten.

SET gewährleistet:

- **Vertraulichkeit der Informationen** bezüglich
  - des Zahlungsereignisses (Kommunikation mit der Bank)
  - des Kaufauftrags (Kommunikation mit dem Händler)

  durch Verschlüsselung der Nachrichten (Verwendung von Sitzungsschlüsseln)
- **Integrität** durch Verwendung digitaler Signaturen
- **Authentifizierung** von Kunde und Händler:
  - Händler: *Verfügt der Kunde tatsächlich über ein gültiges Kreditkartenkonto mit ausreichender Deckung?*
  - Kunde: *Ist der Händler ein vertrauenswürdiger Partner des Kreditkartenunternehmens?*
- SET erreicht dies durch aufwendiges Protokoll unter Verwendung von digitalen Signaturen und Zertifikaten von Kunden, Händlern und Banken.
- SET gewährleistet Sicherheit auf der Ebene des Anwendungsprogramms

Home Banking Computer Interface - HBCI

- Im Auftrag der Spitzenverbände der deutschen Kreditwirtschaft entwickelte Schnittstellenspezifikation als Standard für Homebanking (siehe http://www.hbci.de)
- HBCI ist deshalb eine multibankfähige (also bankunabhängige) Homebanking-Schnittstelle und beschreibt die Schnittstelle zwischen Kundenprodukt und Kreditinstituten.
- HBCI legt fest, in welcher Form bei der Ausführung der Standard-Transaktionen des Home-Bankings kryptographische Verfahren und Smartcards eingesetzt werden können.
- Bietet erheblich höhere Sicherheitsstandards als die bisher übliche Verwendung von Transaktionsnummern (TANs).
- Weitere Einzelheiten zu HBCI folgen in der nächsten Vorlesung.

Weitere Einzelheiten

- Thema Sicherheit wird vertieft in
  - Pflichtvorlesung "Sicherheit / Public Key Kryptographie" für Infowirte
  - Algorithms for Internet Applications

Zusammenfassung

Schwerpunkte der Vorlesung:

- Risiken
- Anforderungen
- Verschlüsselungsverfahren
- Symmetrische Verfahren
- Asymmetrische Verfahren
- Signaturen
- Zertifizierung

nächste Vorlesung: Zahlungssysteme und HBCI

Literatur und Referenzen

- ananymous, hacker's guide sicherheit im internet und im lokalen netz, new technology

  siehe speziell Kapitel: 9 Destruktive Programme, 10 Scanner, 11 Paßwort-Knacker, 12 Trojanische Pferde, 13 Sniffer, 16 Das Sicherheitsloch, 26 Spoofing-Attacken
- Farmer-Studie http://www.trouble.org/survey/ (siehe auch ananymous, hacker's guide sicherheit ... Seite 113)
- Ernst&Young LLP/Information Week Information-Security-Studie http://www.ey.com/publicate/aabs/isaaspdf/FF0148.pdf (siehe auch ananymous, hacker's guide sicherheit ... Seite 114)
- K. Fuhrberg: Internet-Sicherheit, Hanser München 1998

  Kapitel: Gefährdung bei der Nutzung des Internet (Seite 43 - 73),

  Kryptographie (Seite 75 - 97)

Literatur und Referenzen

- B. Schneider: Angewandte Kryptographie, Addison-Wesley, 1996
- siehe speziell Kapitel: 12 Data Encryption Standard (DES), 19 Public-Key-Algorithmen
- R. E. Smith: Internet-Kryptographie, Addison-Wesley, 1998
- Kapitel: Grundlagen der Verschlüsselung (Seite 51 - 79)
- R. Wobst: Abenteuer Kryptologie, Addison-Wesley, 1997
- Seiten: 13 - 19, 52 - 54, 105 - 154, 234 - 246
- H. Schmeck: Algorithms for Internet-Applications, Skript WS 1998/99
- A. Salomaa: Public-Key-Cryptography 2. Auflage, Springer-Verlag 1996

Ergänzende Literatur und Referenzen

- A. Hodges: Alan Turing: The Enigma of Intelligence, Cox & Wyman Ltd, Reading, 1983
- 3. AIK-Symposium: „Sicherheit im Electronic Business“

  http://www.aifb.uni-karlsruhe.de
- Otto Leiberich: Vom diplomatischen Code zur Falltürfunktion, Spektrum der Wissenschaft, Juni 6, 1999, S. 26 - 34
- Henri Cohen: Zahlentheoretische Aspekte der Kryptographie, Informatik Spektrum 24. Juni 2001, 129 - 139
- K. Brunnstein, Universität Hamburg: Aktuelle Probleme der Sicherheit ausgewählter Rechner- und Netz-gestützter Anwendungen, Vortrag im Kolloquium Angewandte Informatik, Universität Karlsruhe, 14.5.1999
- http://www.ietf.org (geltende Standards, Request for Comments)