Datensicherheit - Kryptographie

Thomas Mundt
thm@informatik.uni-rostock.de

Inhalt

- Kryptographie
  - Verschlüsselung
  - Digitale Signaturen

Ziele

- Vermittlung der Grundlagen der Kryptographie
- Einteilung der verfügbaren Verfahren

Anforderungen

- Erinnerung an einige „Grundbedürfnisse“ und Realisierungsmöglichkeiten
  - Vertraulichkeit
    - Verschlüsselung
  - Nichtabstreitbarkeit
    - Digitale Signaturen
  - Authentisierung
    - Digitale Signaturen
    - Public Key Authentication

Verschlüsselung - Einsatzmöglichkeiten

- Wann kann Vertraulichkeit sinnvoll durch Verschlüsselung erreicht werden?
  - Übertragung von Daten über offene Medien
    - E-Mails
    - Webseiten (z.B. Homebanking)
  - Speicherung, wenn Medium nicht speziell gegen physischen oder elektronisch auslesenden Zugriff gesichert werden kann
    - Mehrbenutzersysteme
    - Diebstahlgefährdete Systeme (z.B. Notebooks, Mobile Speicher)

Wert von Informationen

- Welchen Wert hat eine Information?
- Welchen Aufwand treibe ich, um diese Information zu schützen?
- Welcher Aufwand entsteht, um an eine Information zu gelangen?
- Was kosten die Lottozahlen von nächster Woche?
- Was kosten die Lottozahlen von letzter Woche?

Wert von Waren

- In der Klassischen Ökonomie
  - Wert einer Ware entspricht den Kosten für seine Herstellung

- Bei Karl Marx
  - „[Der] Wert einer Ware [ist] bestimmt durch das auf ihre Produktion verwendete Arbeitsquantum.“

Wert von Informationen

- In der Wissensgesellschaft
  - Information, die bei ihrer Verbreitung dem Urheber einen Gewinn oder Vorteil bringt - z.B. Reklame
  - Information, deren zunehmende Verbreitung für den Urheber einen Verlust bedeutet - z.B. die Kenntnis von einem Schatz
  - Preis, den jemand zu zahlen bereit ist

- „25 Prozent des Bruttosozialprodukts werden im Informationenssektor erwirtschaftet.“ Walther Umstätter „Was ist Information eigentlich wert?“ (1988)

Wert von Informationen

- Welchen Wert hat eine Information?
  - Direkt in Geld ausdrückbar
    - electronic money auf SmartCards
    - Kontostände
  - Vorteil für den Besitzer
    - Bekanntheit einer Marke / Markenwert
    - Vorteil vor Konkurrenten
  - Wiederherstellungskosten – z.B. für erneute Datenerfassung

Wert von Informationen

- Naive / Ingenieursmäßiges Herangehen
  - 90min Vorlesung erfordern ca. 300min Vorbereitung
  - 200 EUR Lohnkosten (für 300min)
  - 200 EUR Nebenkosten (Abgaben, Miete etc.)

  - Das Wissen aus dieser Vorlesung kostet also ca. 400 EUR
  - 50 Teilnehmer

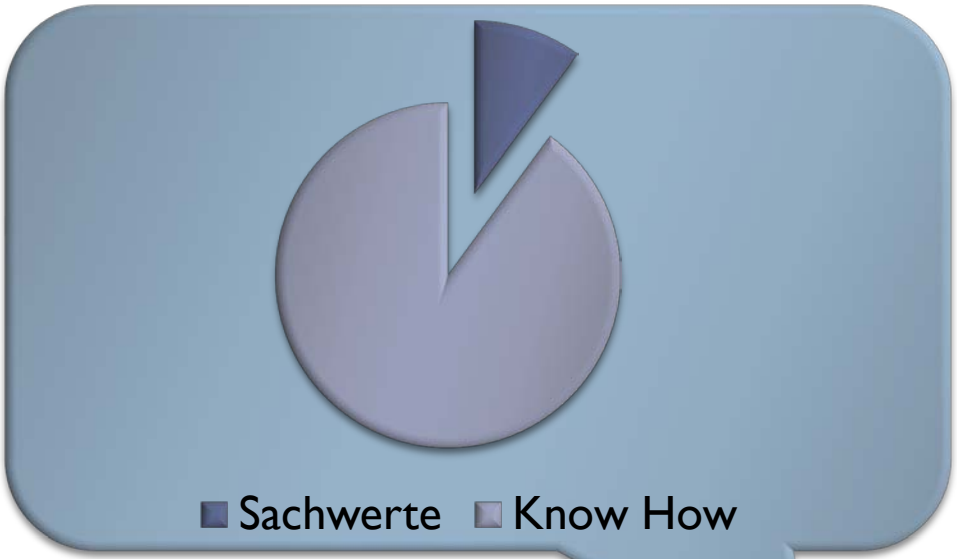
  - 8 EUR pro Download des Scripts

  - Was kosten die Prüfungsfragen?

Wert von Informationen - Beispiele

- Übernahme von Kraft Foods durch Philip Morris (heute Altria Group) kostete 1988 12,9 Mrd. US Dollar, davon
  - 1,3 Mrd. US Dollar Sachwerte
  - 11,6 Mrd. US Dollar Know-How, Kundenstamm, Markennamen, Personalbestand

Wert von Informationen - Beispiele



Wert von Informationen - Beispiele

- Wirtschaftsspionage in Deutschland
  - 1988 ca. 8 Mrd. DM Schaden
  - 2002 geschätzt 10 – 70 Mrd. EUR

- Informationen über den Aufenthaltsort von Kriegsschiffen
  - Lebensbedrohende Informationen
  - Kriegsentscheidende Informationen

Verschlüsseln vs. Verstecken

- Neben kryptografischen Verfahren, auch Verstecken von Informationen möglich
- Steganografie
  - Veränderte Texte
  - Ausnutzung des Rauschens in Bildern und Audiodaten / z.B. Verwendung des niederwertigsten Bits
  - Information kann so versteckt werden, dass ihre Existenz nicht nachweisbar ist
- Keine Kryptografie → jetzt nicht Thema

Begriffe

- Klartext / Plaintext / P
- Geheimtext / Ciphertext / C
- Schlüssel / Key / K
- Kryptografie
  - Lehre vom Verschlüsseln
- Kryptoanalyse
  - Rückgängigmachen der Verschlüsselung ohne Kenntnis des Schlüssels
- Klartextangriff
  - Kryptoanalyse mit (teilweise) bekanntem Klartext

Substitutionsverfahren

- Cäsar Addition mit „Schlüssel“ 3
  - A → D
  - B → E
  - C → F
  - …
  - W → Z
  - X → A
  - Y → B

Cäsar-Verschlüsselung

- $c = p + K \bmod 26$

- c Ciphertext — Verschlüsseltes Zeichen
- p Plaintext — Klartextzeichen
- K Key — Schlüssel

- Codierung des Alphabets als Zahl von A = 0 bis Z = 25

Substitutionsverfahren

- Anzahl der Schlüssel bei 26 Buchstaben
  - Cäsar-Methode: 25 sinnvolle Schlüssel
  - Substitionsverfahren allgemein: Permutation, also 26! = 403291461126605635584000000 (403 Quadrillionen, $4{,}03*10^{26}$), davon etwa 400 Quadrillionen sinnvoll

Substitutionsverfahren

- Angriffe über Buchstabenhäufigkeit
  - E 13,2%
  - N 8,1%
  - I 6,7%
  - R 5,7%
  - T 5,2%
  - S 4,4%
  - A 4,0%
  - H 3,8%
  - L 3,0%

Wobst, Abenteuer Kryptologie

Substitutionsverfahren

- Andere Aussagen zur Häufigkeit

Platz	Buchstabe	relative Häufigkeit
1.	E	17,40 %
2.	N	09,78 %
3.	I	07,55 %
4.	S	07,27 %
5.	R	07,00 %
6.	A	06,51 %
7.	T	06,15 %
8.	D	05,08 %
9.	H	04,76 %
10.	U	04,35 %

Albrecht Beutelspacher, *Kryptologie*, 7. Aufl., Wiesbaden: Vieweg Verlagsgesellschaft, 2005, Seite 10

Substitutionsverfahren

- Anfangsbuchstaben

Platz	Buchstabe	relative Häufigkeit
1.	D	14,2 %
2.	S	10,8 %
3.	E	07,8 %
4.	I	07,1 %
5.	W	06,8 %

Substitutionsverfahren

- Endbuchstaben

Platz	Buchstabe	relative Häufigkeit
1.	N	21,0 %
2.	E	15,1 %
3.	R	13,0 %
4.	T	10,3 %
5.	S	09,6 %

Substitutionsverfahren

- Angriffe über Paarhäufigkeit
  - EN 3,1%
  - ER 2,7%
  - CH 2,4%
  - EI 1,6%
  - TE 1,5%
  - IE 1,5%

Substitutionsverfahren

- Trotz der Vielzahl der Schlüssel absolut ungeeignet
- Sortieren der Buchstaben und Buchstabenpaare nach Häufigkeit dauert auf Pentium 4 (2 GHz) weniger als 0,1 s für 1 MB Text

- Verschlüsselter Text kann nahezu online mitgelesen werden

- Cäsar: „*pollice verso*“

Homophone Substitution

- Ziel: „Überlistung“ der Statistik
- Häufige Buchstaben werden durch mehrere Zeichen substituiert
  - z.B. „E“ → „A“, „4“, „%“
- Im Idealfall sind alle Zeichen gleich häufig im verschlüsselten Text

Homophone Substitution

- Angriffsmöglichkeiten
  - Schlechter Zufallsgenerator zur Auswahl der Zeichen
  - (Teilweise) bekannter Klartext
  - Bei teilweise bekanntem Schlüssel ist sprachliche Ergänzung zu einf - - h
  - Innere Gesetzmäßigkeiten der Sprache werden ausgenutzt

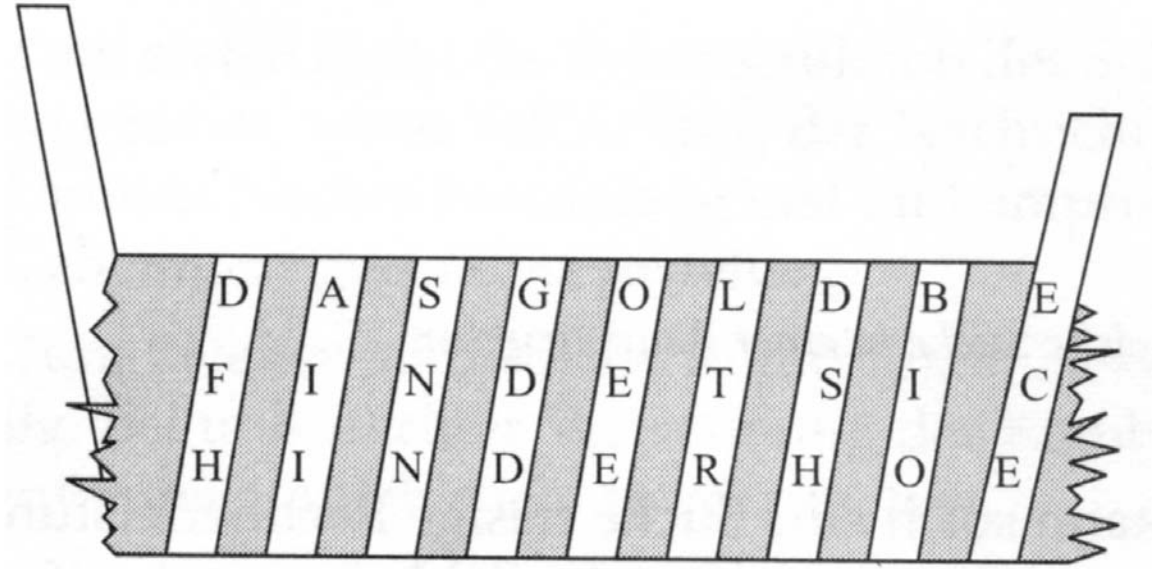
Transpositionsverfahren

DIEV
ORLE
SUNG
BEGI → **DOSBNMIIRUENDXELNGTRXVEGIUEX**
NNTU
MDRE
IXXX

- Das sogenannte „Rechteckverfahren“
- Schlüssel sind die Kantenlängen

Transpositionsverfahren

- Verschlüsselung basiert auf der Änderung der Reihenfolge der Zeichen



Reinhard Wobst, Abenteuer Kryptologie

Kryptoanalyse von Transpositionsverfahren

- Statistische Häufigkeit der Buchstaben führt nicht zum Resultat
- Auffüllen mit X ist gefährlich
- Nicht enthaltene Buchstaben schließen eine Reihe von Klartexten aus
- Klartextangriffe relativ einfach

Polyalphabetische Substitution

- Substitutionsvorschrift von der Position des Textes abhängig
- 1466 von Alberti erstmalig beschrieben
- Vigenère-Chiffrierung benutzt einen wiederholt aneinander gereihten Schlüssel

Vigenère-Chiffrierung

- ABCDABCDABCDABCDABCDABCDABCDA
- DIEVORLESUNGBEGINNTUMVIER

- A+D = D
- B+I = J
- C+E = G
- D+V = Y

C = P + K mod 26

Schlüssel: ABCD

A = 0, B = 1 usw.

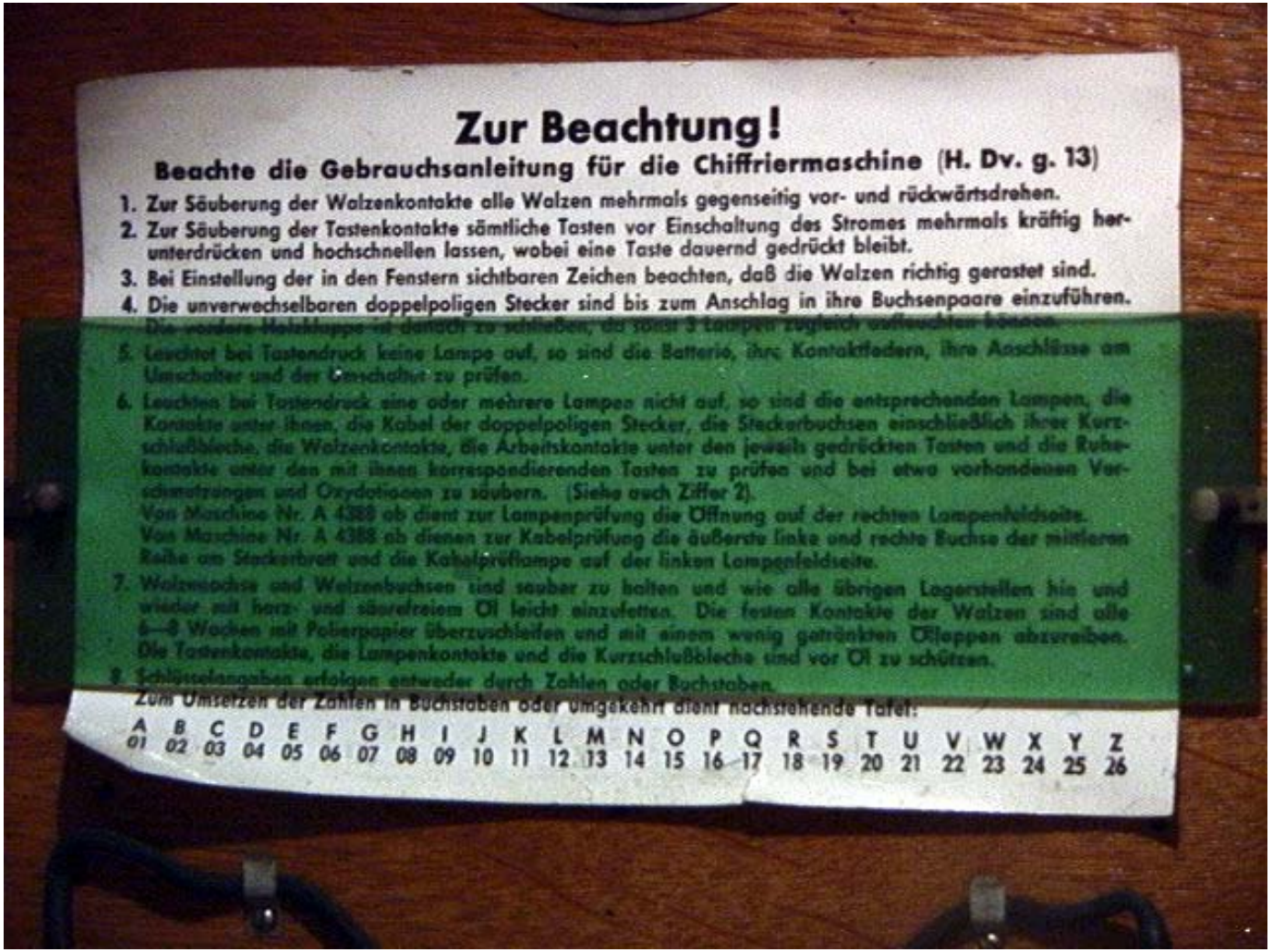
Kryptoanalyse der Vigenère-Chiffrierung

- Schlüssellängen durchprobieren (hier 4)
- 1., 5., 9., 13., 17. … Zeichen auswählen
- Statistische Analyse wie bei Cäsar-Verschlüsselung
- Analog für 2., 6., 10., 14., 21. … Zeichen

Rotormaschinen

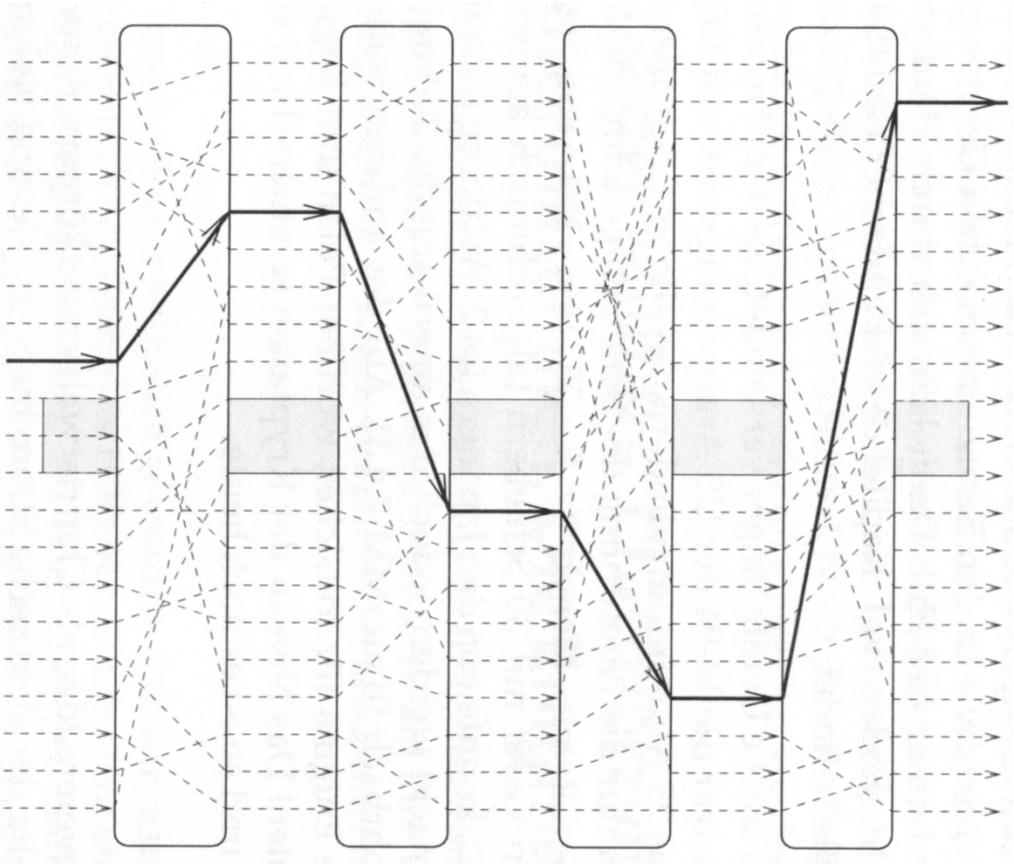
Thomas Mundt

Rotormaschinen



Thomas Mundt

Enigma



Enigma

- Mehrstufige Vigenère-Verschlüsselung (Substitution)
- Drehung einer Walze nach jedem Schritt
- Drehung der Walzen bewirkt Verlängerung der Periode der Substitution
  - Bei Enigma mit drei Walzen ist die Periodenlänge $26^3=17576$
- Zusätzliche Substitution durch Steckkontakte

Enigma

- Anzahl der Schlüssel
  - 3 Walzen aus 5 auswählbar (10)
  - Walzenreihenfolge (3!=6)
  - Walzen-Grundstellungen (26^3=17576)
  - Permutation auf dem Steckbrett (26! / (13! * 2^{13}) = 7905853580625)
  - Ergibt: 833719695198390000 (Trillionen, 8.3E18) verschiedene Schlüssel

Analyse der Enigma

- Maschine und Walzen waren vor Ausbruch des Krieges bekannt
- Der Gegner kennt immer den Algorithmus!
- Klartext teilweise vorhersagbar
  - Grußformel am Ende der Nachricht
  - Nachricht enthält „Absender“
  - Untergeschobene Klartexte / Provozierte Nachrichten

Analyse der Enigma - Fehler im Algorithmus

- Einbau eines Reflektors, um vermeintlich „doppelte“ Verschlüsselung zu erreichen
- Dadurch keine Verschlüsselung eines Zeichens auf sich selbst
- Ermöglichte negative Mustersuche mit bekannten Textstellen (“OBERKOMMANDO WEHRMACHT“)

Negative Mustersuche

```
AOODAZTEADGOIKDKJUEOZGVFW
OBERKOMMANDO
 OBERKOMMANDO
  OBERKOMMANDO
   OBERKOMMANDO      ←
    OBERKOMMANDO
```

Analyse der Enigma - Fehler der Benutzer

- Codebuch (mit Grundstellung und Auswahl der Walzen) kommt abhanden
- Spruchschlüssel falsch gewählt (AAA, BBB, TZU, …)
- Spruchschlüssel am Anfang zweimal gesendet, um Empfang trotz Funkstille (kein „Bitte noch einmal!“) sicher zu stellen
  - Dadurch Analyse der Verdrahtung der Walzen

Automatisierte Analyse

- Projekt „Bletchley Park“
- Erstmals „rechnerunterstützte“ Kryptoanalyse
- A. Turing und ca. 200 Kryptoanalytiker
- Bis ca. 1974 geheimgehalten

- Man weiß nie, was der Gegner kann oder weiß!

One-Time-Pads

- Schlüssel wir nur einmal benutzt (hier bitweise mod 2 anstatt buchstabenweise mod 25)

```
P=10011001100100010110100001011
K=01001000100111110001000101001
C=11010001000011100111100100010
```

- Das einzige beweisbar sichere Verfahren
- One-Time-Pads sind polyalphabetische Verfahren mit unendlicher Periode

One-Time-Pads

- Keine Gesetzmäßigkeiten, weil Schlüssel mindestens so lang wie der Klartext ist
- Vorteile
  - Extrem schnell als Hardware realisierbar
  - Absolut sicher
- Probleme
  - Handhabung des Schlüssels
    - Groß
    - Nur einmalig verwendbar
  - Gute Zufallszahl für den Schlüssel

Symmetrische Verschlüsselung

- Der gleiche Schlüssel wird zum Verschlüsseln und zum Entschlüsseln verwendet

$$C = P \otimes K \qquad P = C \overline{\otimes} K$$

- Alle bisher genannten Verfahren sind symmetrisch

(Zur Erinnerung: C Geheimtext, P Klartext, K Schlüssel)

Asymmetrische Verschlüsselung

- Es gibt zwei Schlüssel, einen zum Verschlüsseln (öffentlicher Schlüssel) und einen zum Entschlüsseln (privater Schlüssel)

- $C = P \otimes Kpub$ $\qquad\qquad$ $P = C \otimes Kpriv$

- Anderer Begriff: Public Key Verschlüsselung
- Der private Schlüssel lässt sich nicht (im kryptografischen Sinne) aus dem öffentlichen Schlüssel ableiten

Asymmetrische Verschlüsselung

- Kryptologische Sicherheit nötig
  - Klartext darf nicht aus dem Geheimtext ableitbar sein
  - Privater Schlüssel darf nicht aus dem öffentlichen Schlüssel ableitbar sein
- Vorteil
  - Private Schlüssel müssen nicht ausgetauscht bzw. transportiert werden

Nachteile asymmetrischer Verfahren

- Algorithmen sind langsam
- Nicht vollständig erforscht
- Jeder hat eigenen öffentlichen Schlüssel
- Verteilung der öffentlichen Schlüssel aufwendig

Hybride Verfahren - Verschlüsselung

- Öffentlichen Schlüssel des Empfängers besorgen
- Zufälligen Sitzungsschlüssel erzeugen
- Nachricht mit dem Sitzungsschlüssel verschlüsseln
- Sitzungsschlüssel mit dem öffentlichen Schlüssel des Empfängers verschlüsseln
- Beide Chiffrate an Empfänger senden

Hybride Verfahren - Entschlüsselung

- Sitzungsschlüssel mit dem privaten Schlüssel entschlüsseln
- Nachricht mit dem Sitzungsschlüssel entschlüsseln

Hybride Verfahren – Praktischer Einsatz

- SSL / TLS
  - HTTPS
  - http://www.ietf.org/rfc/rfc2246.txt
  - http://www.ietf.org/rfc/rfc2817.txt
  - http://www.ietf.org/rfc/rfc2818.txt
- Später mehr dazu

Stromverfahren

- Verschlüsselung eines laufenden Datenstroms online
- Berechnung auf Bit-Ebene
- Zum Beispiel durch XOR
  - Pad kann vorab berechnet und gespeichert werden

Blockverfahren

- Verschlüsselung ganzer Blöcke von Daten
- Blocklänge hängt vom Verfahren ab, beim Data Encryption Standard (DES) zum Beispiel 64 bit
- Zum Beispiel Transpositionen

Data Encryption Standard DES

- 1974 von IBM vorgeschlagen
- 56 bit langer Schlüssel
- Überführt 64 bit Klartext in 64 bit Geheimtext und umgekehrt
- Benutzt „Feistel-Netzwerke“

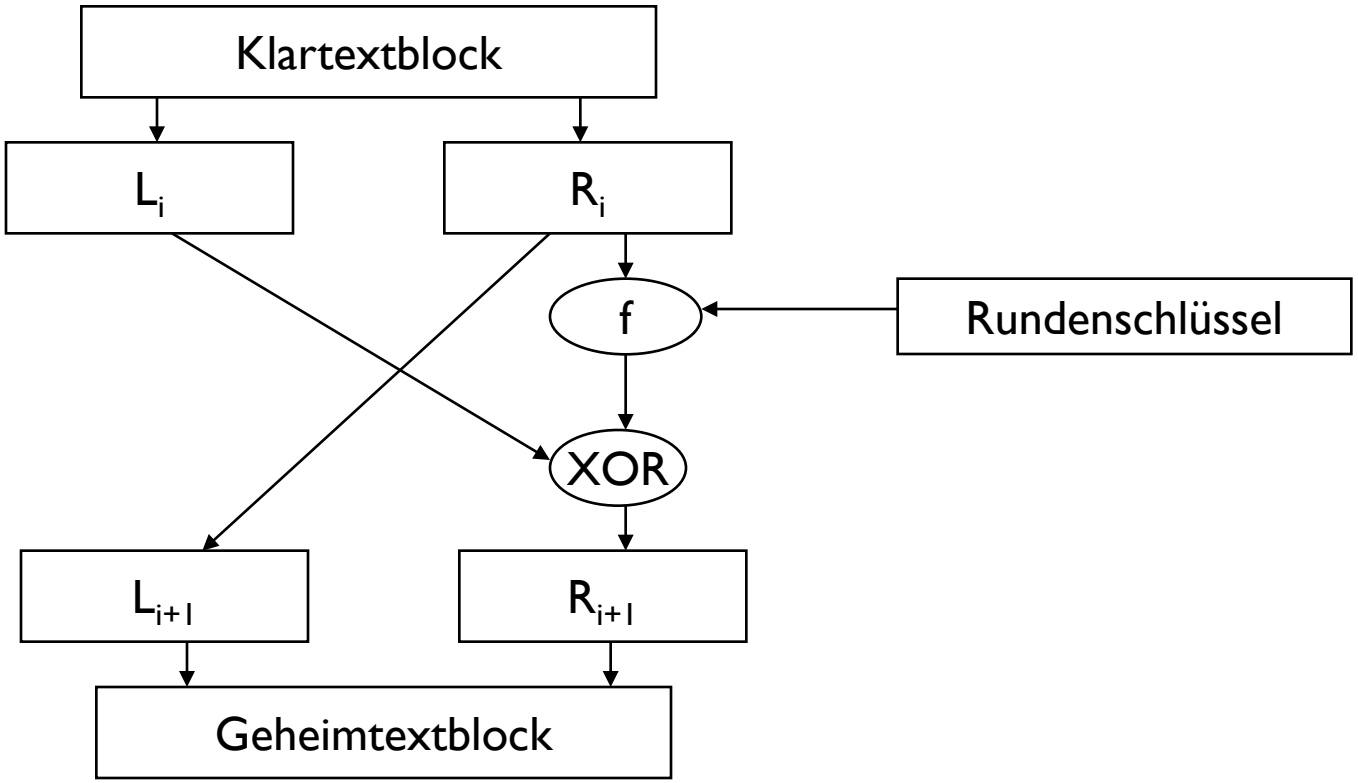
Feistel-Netzwerke - Verschlüsselung

- Blöcke in gleich große Hälften L und R teilen
- Mehrere Verschlüsselungs-Runden
- In jeder Runde i beide Halbblöcke vertauschen und Li mit $f_{Ki}(R_i)$ per XOR verknüpfen

Feistel-Netzwerke - Verschlüsselung

- $L_{i+1} = R_i$
- $R_{i+1} = L_i \quad XOR \quad f_{Ki}(R_i)$

(I)



Feistel-Netzwerke

Aus (I) folgt:

$L_i = L_i \text{ XOR } f_{Ki}(R_i) \text{ XOR } f_{Ki}(R_i)$

$= R_{i+1} \text{ XOR } f_{Ki}(R_i)$

und daraus folgt die Entschlüsselung für n-Runden-Verfahren

$R_{n-1} = L_n$ (folgt direkt aus $L_{i+1} = R_i$)

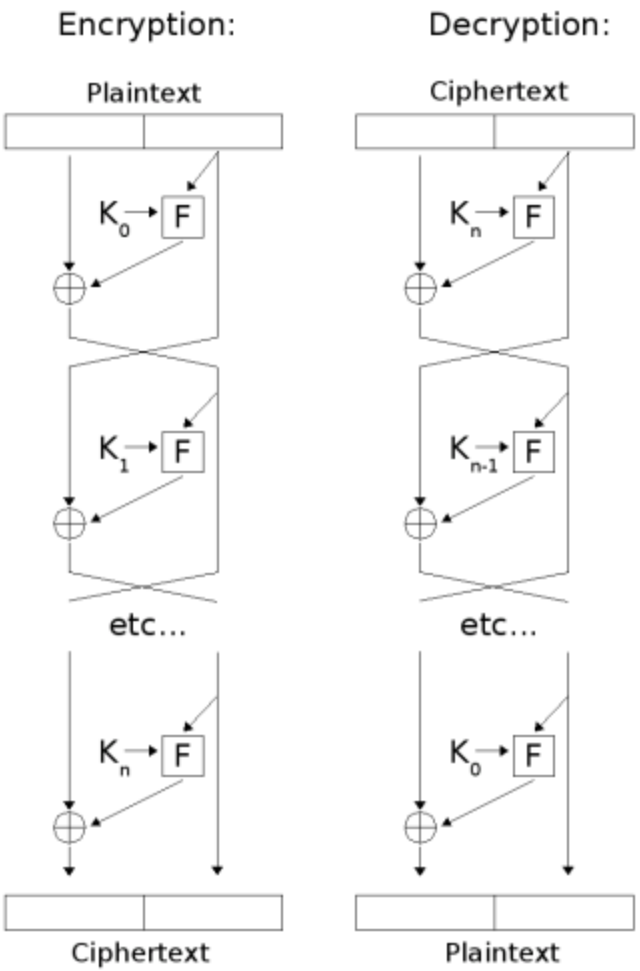
$L_{n-1} = R_n \text{ XOR } f_{Kn-1}(R_{n-1})$

...

$R_0 = L_1$

$L_0 = R_1 \text{ XOR } f_{K0}(R_0)$

Feistel-Chiffre

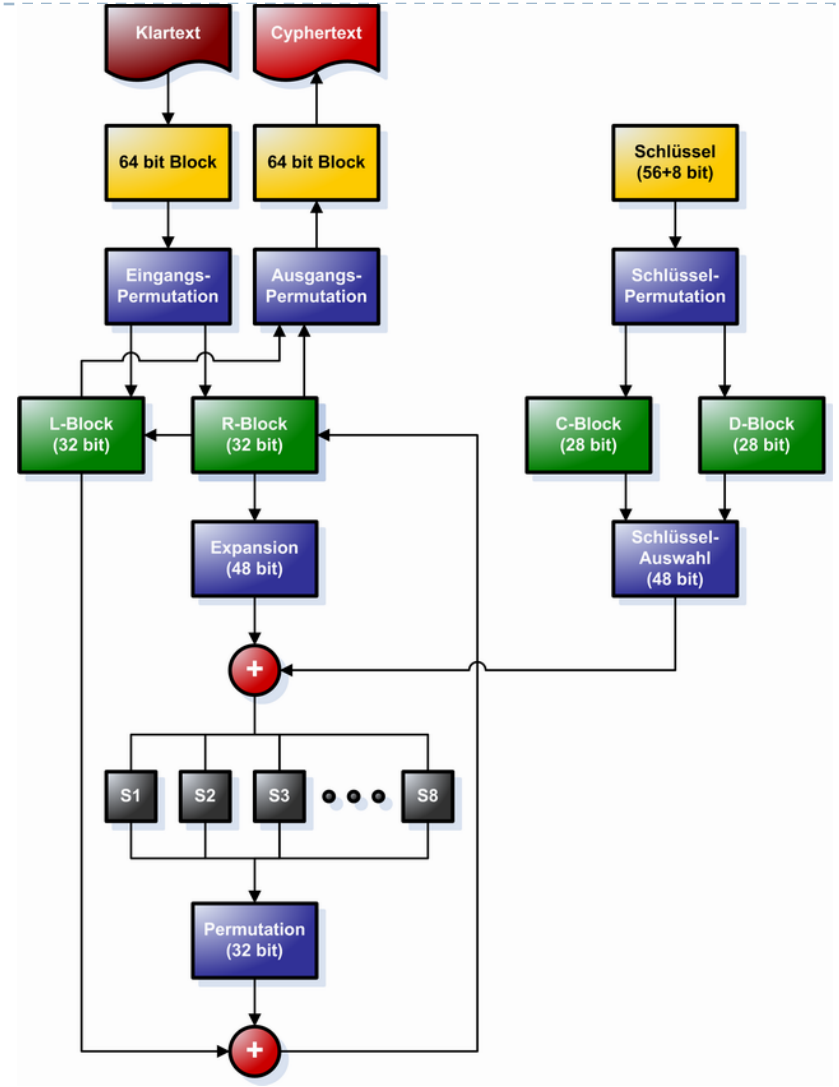


Feistel Cipher

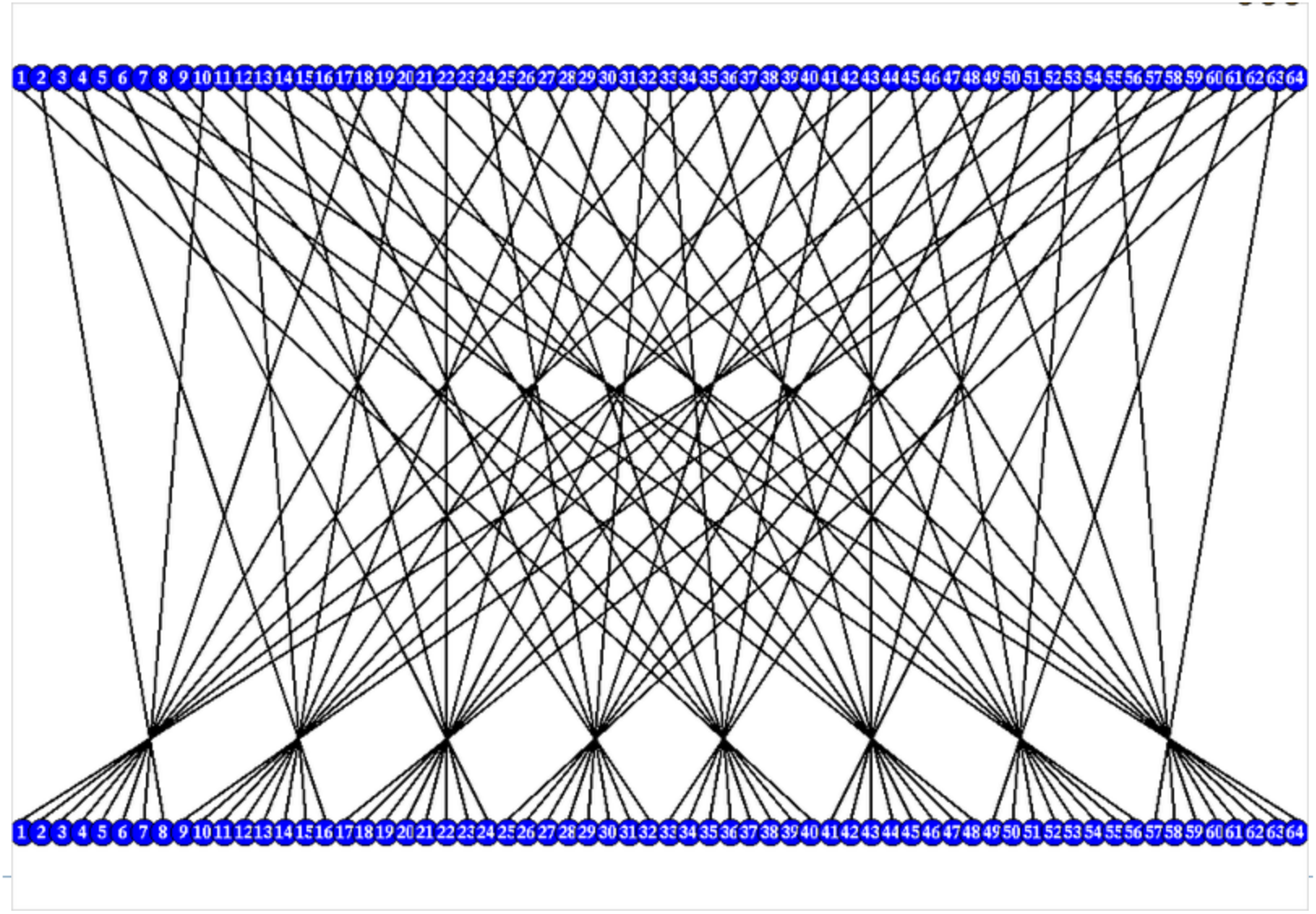
Feistel-Netzwerke - Vorteile

- Die Funktionen f_{Ki} brauchen nicht umkehrbar zu sein
- Bisher gefordert: Umkehrung von $C = f_K(P)$ nur mit Hilfe von K möglich
- Jetzt: f_{Ki} nicht ohne Kenntnis von K berechenbar

DES Verfahren



DES – Eingangspermutation



DES S-Boxen

- S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	**4**	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

- Wandelt 6 Bits in 4 Bits um
- Auswahl der Zeile durch äußere 2 Bit
- Auswahl der Spalte durch innere 4 Bit

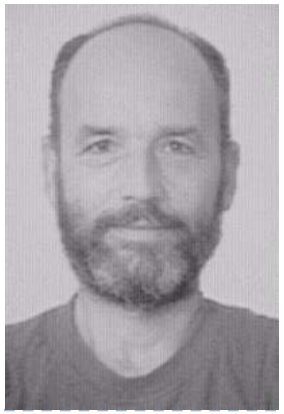
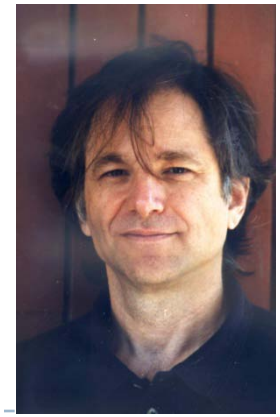
- 000111 → Zeile 1 (01), Spalte 3 (0011) → 4 (0100)

S-Boxen

- "In the development of DES, NSA convinced IBM that a reduced key size was sufficient; indirectly assisted in the development of the S-box structures; and certified that the final DES algorithm was, to the best of their knowledge, free from any statistical or mathematical weakness." D.W. Davies and W.L. Price, Security for computer networks, 2nd ed., John Wiley & Sons, 1989

RSA

- 1978 von Rivest, Shamir, Adleman veröffentlichter Algorithmus zur asymmetrischen Verschlüsselung
- Beruht auf der Faktorisierung großer Zahlen in Primfaktoren
  - Häufig ließt man falsch: „Faktorisierung von Primzahlen!“
- Faktorisierung ist mathematisch schwieriges Problem

Faktorisierung

- n = pq, wobei p und q Primzahlen
- Bei bekanntem n ist es sehr schwierig, p und q zu berechnen
- Schwierigkeit steigt mit der Stellenzahl von n

RSA Verfahren

- Schlüsselerzeugung
  - Wähle zwei große Primzahlen p und q
  - Bilde n=pq
  - Wähle e > 1, das zu (p-1)(q-1) teilerfremd ist
  - Berechne ein d mit de = 1 (mod ((p-1)(q-1)))
  - n und e bilden öffentlichen, n und d privaten Schlüssel (→ d != e)

RSA Ver- und Entschlüsselung

- Verschlüsselung

$$C = P^e \bmod n$$

- Entschlüsselung

$$P = C^d \bmod n$$

Plaintext P in Stücke kleiner als n zerlegen

RSA - Beispiel

- p = 3, q = 5
- n = pq = 15
- Wähle e > 1, das zu (p-1)(q-1) teilerfremd ist
  - e muss kleiner als der Wert sein, der sich ergibt, wenn man p und q miteinander multipliziert (Produkt aus p und q)
  - e darf keine gemeinsamen Faktoren mit dem Produkt (q-1)(p-1) haben
  - e braucht keine Primzahl zu sein, muss aber ungerade sein ((q-1)(p-1) ist immer gerade)

RSA - Beispiel

- (p-1)(q-1) = 8 = 2 * 2 * 2 → e darf nicht 2 als Teiler haben (stimmt hier sowieso, wegen „e ungerade“)
- e = 3 ist eine einfache Möglichkeit
- d wählen, so dass de = 1 (mod ((p-1)(q-1)))
  - d = (x(p-1)(q-1)+1)/e x ganzzahlig
  - d durch Ausprobieren finden
  - x = 4 kann gewählt werden
  - d = 11

RSA - Beispiel

- p = 3, q = 5, n = 15, d = 11, e = 3
- **$C = P^e \bmod n$ Verschlüsselung**
- **P = 12130508**
- P in Blöcke < n aufteilen
- **12 13 05 08**
- **$12^3 = 1728$ 1728/15 = 115 Rest 03**
- **$13^3 = 2197$ 2197/15 = 146 Rest 07**
- **$05^3 = 125$ 125/15 = 8 Rest 05**
- **$08^3 = 512$ 512/15 = 34 Rest 02**

RSA - Beispiel

- p = 3, q = 5, n = 15, d = 11, e = 3
- **P = C^d mod n Entschlüsselung**
- **C = 03 07 05 02**
- **03^{11} = 177147**
  177147/15 = 11809 Rest 12

- **07^{11} = 1977326743**
  1977326743/15 = 131821782 Rest 13

...

RSA

- Typische Schlüssellängen > 1024 bit
- Keine Garantie, dass Faktorisierung nicht doch „schneller“ möglich ist

RSA in einer Zeile Perl

- ```
  print pack"C*",split/\D+/,`echo
  "16iII*o\U@{$/=$z;[(pop,pop,unpack"H*",<>
  )]}\EsMsKsN0[lN*1lK[d2%Sa2/d0<X+d*lMLa^*lN
  %0]dsXx++lMlN/dsM0<J]dsJxp"|dc`
  ```
  Adam Back

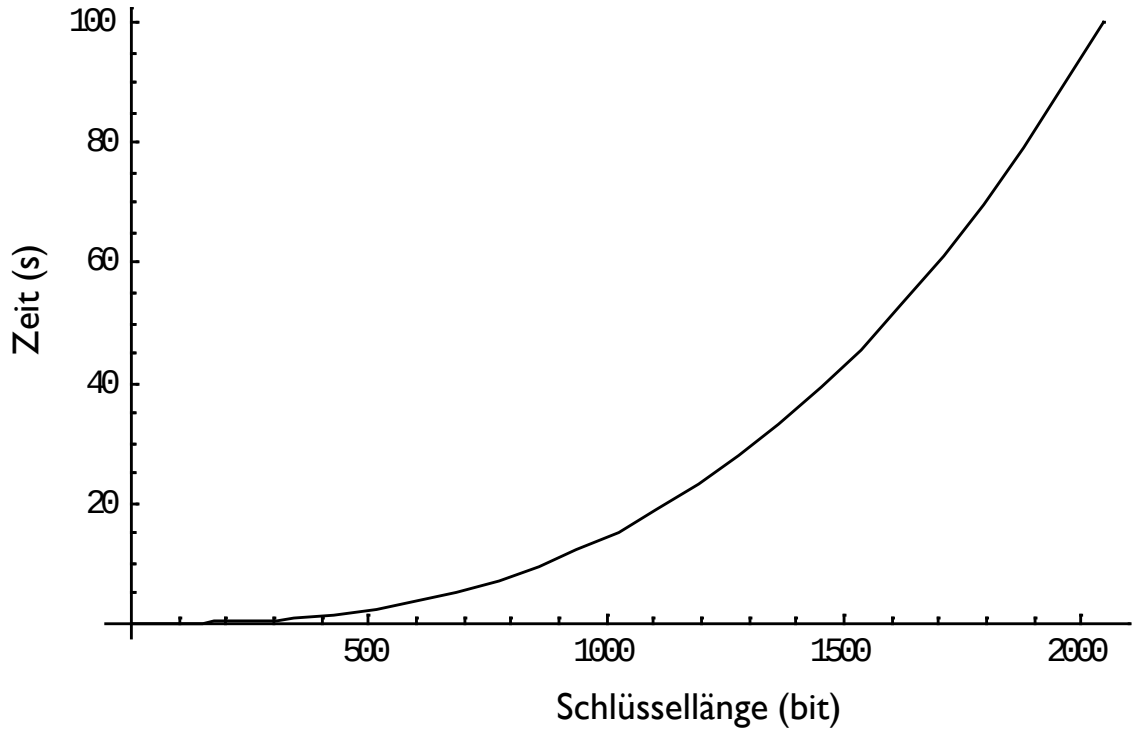

  ```

Andere asymmetrische Verfahren

- Diffie Hellmann und El-Gamal
  - Basieren auf der Schwierigkeit, den diskreten Logarithmus zu berechnen
- Elliptic Curve Cryptography
  - Mitte der 80er Jahre von Victor Miller und Neal Koblitz vorgeschlagen

RSA Rechenzeit

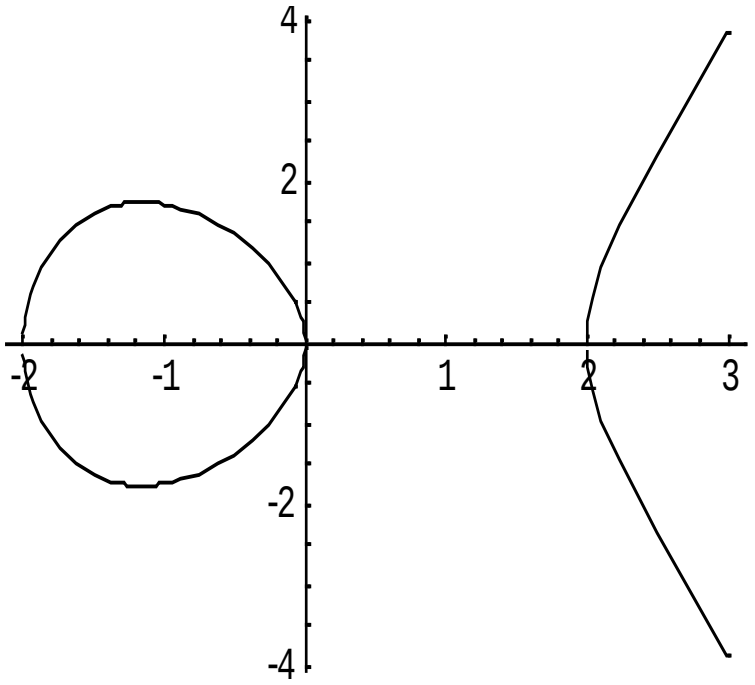
- Zeitaufwand steigt exponentiell mit der Länge des Schlüssels in Bit



Elliptic Curve Cryptography

- Elliptische Kurve allgemein:
  $y^2 = x^3 + ax + b$

- Hier a = -4, b = 0



Elliptic Curve Cryptography

- $< G, \bullet >$ sei Gruppe

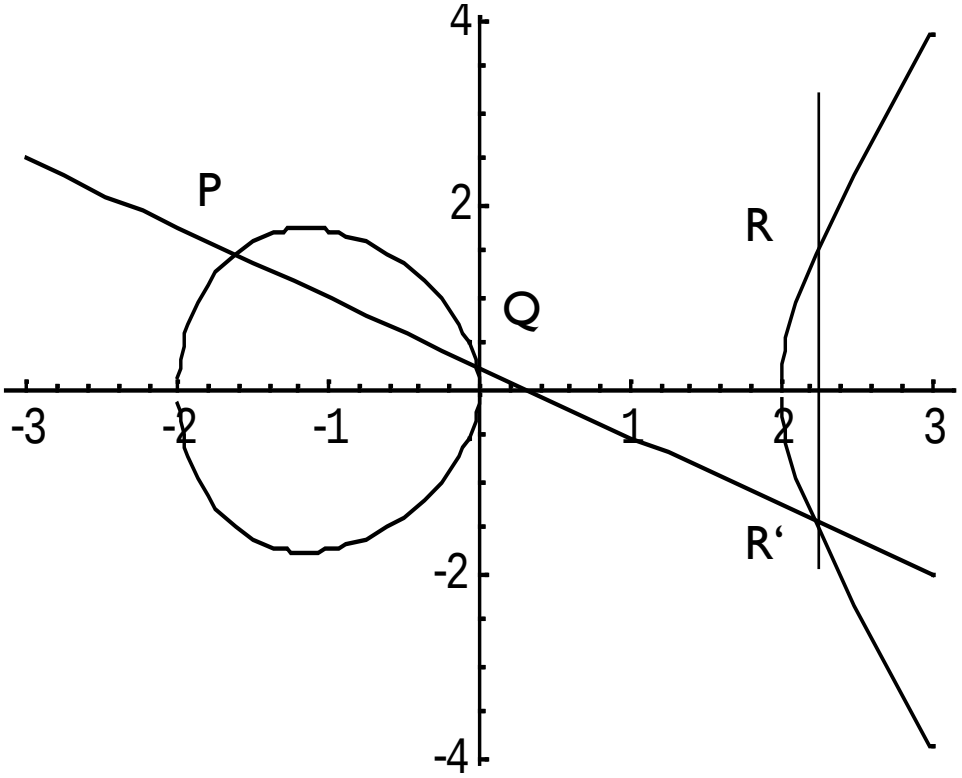
- Erinnerung Gruppeneigenschaften:
  - Abgeschlossenheit $\forall a, b \in G : a \bullet b \in G$
  - Assoziativität $\forall a, b, c \in G : a \bullet (b \bullet c) = (a \bullet b) \bullet c$
  - Neutrales Element $a \bullet e = a$
  - Inverses Element $a \bullet a' = e$

Elliptic Curve Cryptography

- P(x,y) bilden eine Gruppe

$$P \bullet Q = R$$

- $\bullet$ bezeichnet man als Punktaddition



Elliptic Curve Cryptography

- Gerade $g : y = d \cdot x - y_0$
- Anstieg $d = \frac{y_Q - y_P}{x_Q - x_P}$

$$y_0 = y_p - d \cdot x_p$$

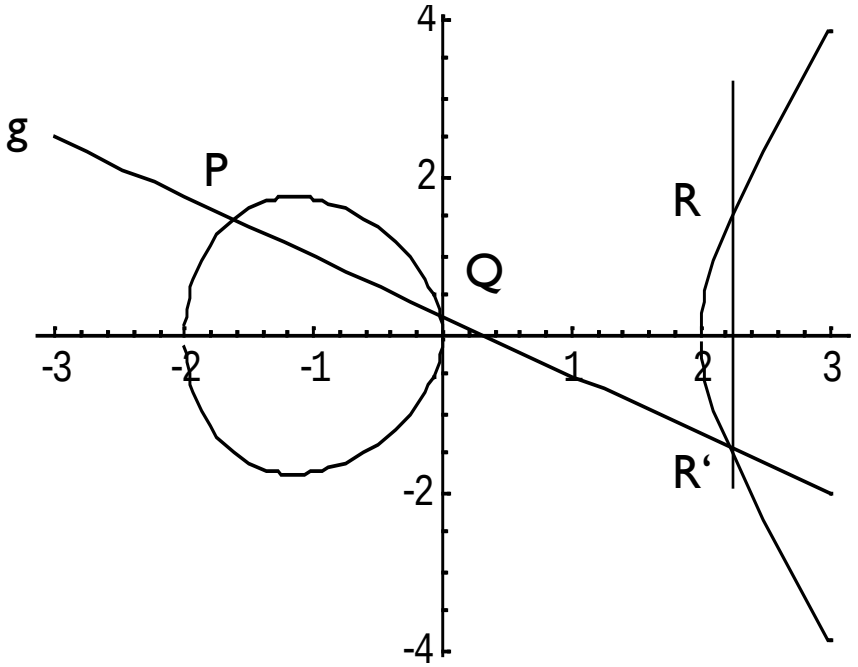
- Schnittpunkt mit Kurve

$$(d \cdot x - y_0)^2 = x^3 + ax + b$$

- Punkt R

$$x_R = d^2 - x_P - x_Q$$

$$y_R = -(d \cdot x_R + y_0)$$



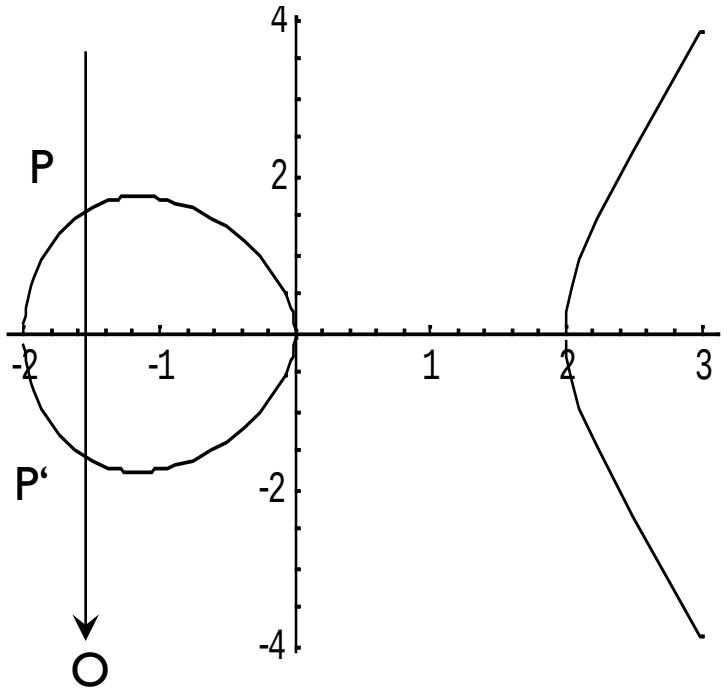
Elliptic Curve Cryptography

- Inverses Element

$$P \bullet P' = O$$

- Neutrales Element

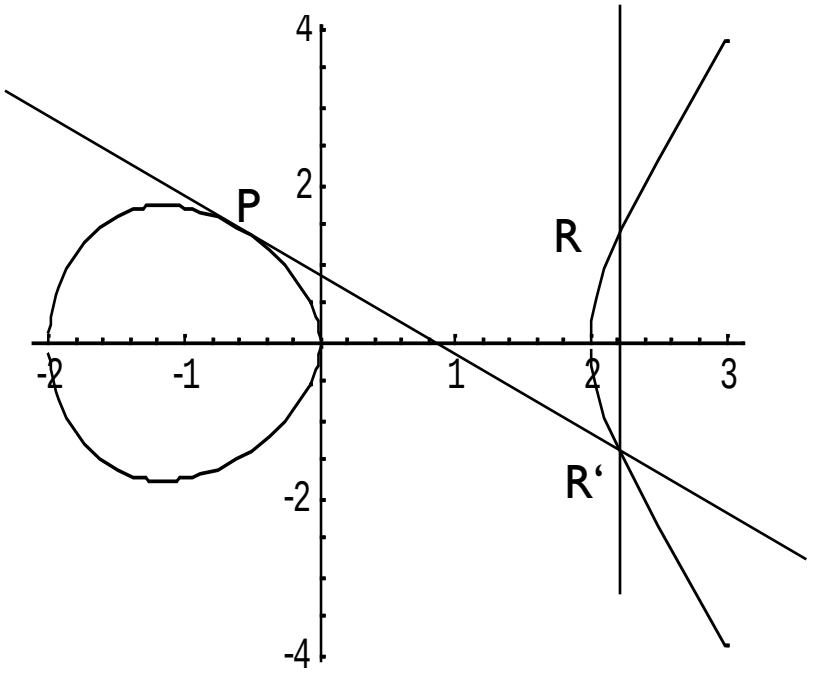
$$O(x, \infty)$$



Elliptic Curve Cryptography

- Punktaddition mit sich selbst

$$P \bullet P = R$$



Elliptic Curve Cryptography

$$g : y = d \cdot x - y_0$$

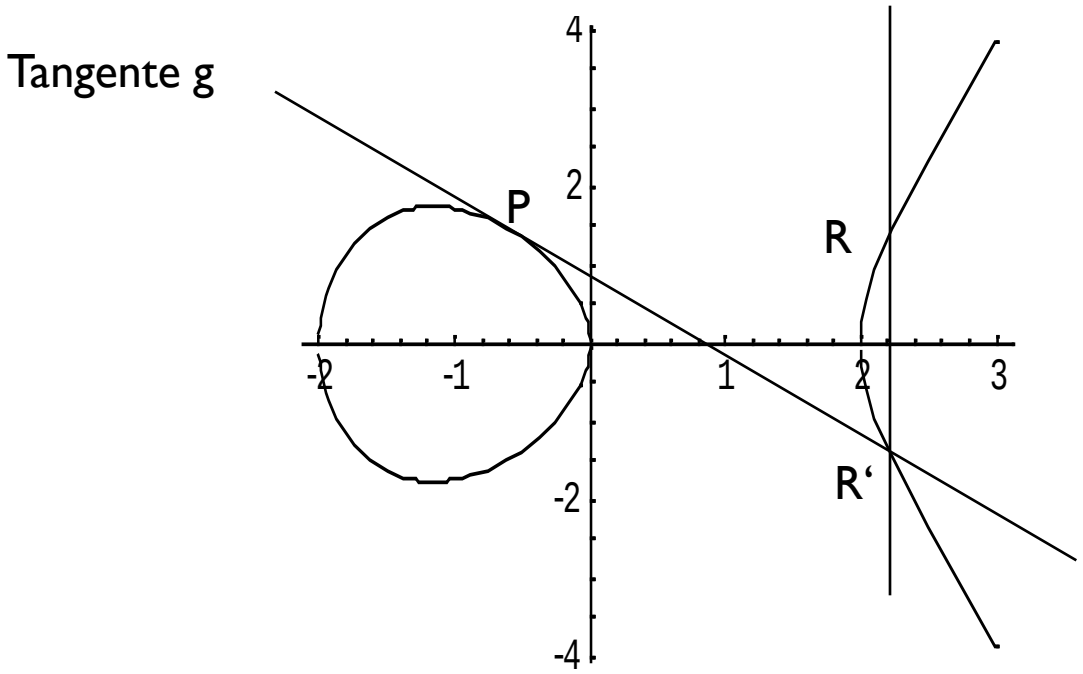
$$d = \frac{dy}{dx} = \frac{3x_P^2 + a}{2y_P}$$

$$y_0 = y_P - d \cdot x_P$$

$$(d \cdot x - y_0)^2 = x^3 + ax + b$$

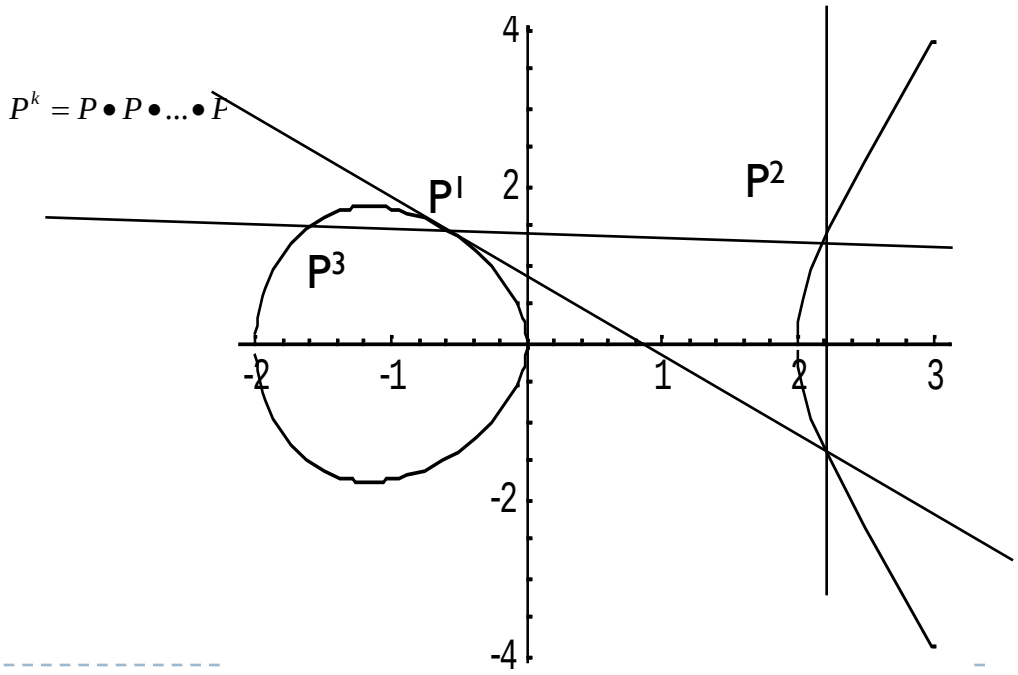
$$x_R = d^2 - 2x_P$$

$$y_R = -(d \cdot x_R + y_0)$$



Elliptic Curve Cryptography

- Iteration



Elliptic Curve Cryptography

- Elliptische Kurven im Galois Feld (GF) (GF ist endlicher Körper)

$$GF_p : y^2 = x^3 + ax + b \bmod p$$

  wobei die Feldgröße p eine Primzahl ist

- {0,1,...,p-1} ist abelsche Gruppe mit Addition mod p
- {1,2,...,p-1} ist abelsche Gruppe mit Multiplikation mod p

- Abelsche Gruppe: Gruppe, bei der Kommutativität gilt

Elliptic Curve Cryptography

$$c = a \cdot b$$

in GF_{11}

*	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10
2	0	2	4	6	8	10	1	3	5	7	9
3	0	3	6	9	1	4	7	10	2	5	8
4	0	4	8	1	5	9	2	6	10	3	7
5	0	5	10	4	9	3	8	2	7	1	6
6	0	6	1	7	2	8	3	9	4	10	5
7	0	7	3	10	6	2	9	5	1	8	4
8	0	8	5	2	10	7	4	1	9	6	3
9	0	9	7	5	3	1	10	8	6	4	2
10	0	10	9	8	7	6	5	4	3	2	1

Elliptic Curve Cryptography

$$x^2 = 5$$

in GF_{11}

Lösung:

$x_1=4$

$x_2=7$

*	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10
2	0	2	4	6	8	10	1	3	5	7	9
3	0	3	6	9	1	4	7	10	2	5	8
4	0	4	8	1	5	9	2	6	10	3	7
5	0	5	10	4	9	3	8	2	7	1	6
6	0	6	1	7	2	8	3	9	4	10	5
7	0	7	3	10	6	2	9	5	1	8	4
8	0	8	5	2	10	7	4	1	9	6	3
9	0	9	7	5	3	1	10	8	6	4	2
10	0	10	9	8	7	6	5	4	3	2	1

Elliptic Curve Cryptography

- Gegeben:
  - Elliptische Kurve mit a=1, b=6

$$y^2 = x^3 + x + 6$$

  - p=11 (GF_{11})
- Gesucht:
  - Punkte P(x,y), x und y in GF_{11}, die die Gleichung erfüllen

Elliptic Curve Cryptography

- Lösung:

$$y^2 = x^3 + x + 6$$

- Es gibt 12 Punkte auf der Elliptischen Kurve in GF11
- Punkte bilden Gruppe <P> der Ordnung n=13
  - 12 Punkte auf der Kurve mit dem Punkt O
  - Ordnung hängt von der Wahl von a und b ab

X	y^2	y_1	y_2	P	P‘
0	6				
1	8				
2	5	4	7	(2,4)	(2,7)
3	3	5	6	(3,5)	(3,6)
4	8				
5	4	2	9	(5,2)	(5,9)
6	8				
7	4	2	9	(7,2)	(7,9)
8	9	3	8	(8,3)	(8,8)
9	7				
10	4	2	9	(10,2)	(10,9)

Elliptic Curve Cryptography

- Gegeben: P(2,4), der auf $y^2 = x^3 + x + 6$ in GF11 liegt
- Gesucht: Iteration P_k
- Lösung:
- P2(5,9)
- P3(8,8)
- P4(10,9)

$$P^2 = P \bullet P = R$$

$$d = \frac{dy}{dx} = \frac{3x_P^2 + a}{2y_P}$$

$$y_0 = y_P - d \cdot x_P$$

$$(d \cdot x - y_0)^2 = x^3 + ax + b$$

$$x_R = d^2 - 2x_P$$

$$y_R = -(d \cdot x_R + y_0)$$

Elliptic Curve Cryptography

- Iterationstabelle für P(2,4)

$$d = \frac{dy}{dx} = \frac{3x_P^2 + a}{2y_P}$$

$$y_0 = y_P - d \cdot x_P$$

$$(d \cdot x - y_0)^2 = x^3 + ax + b$$

$$x_R = d^2 - 2x_P$$

$$y_R = -(d \cdot x_R + y_0)$$

k	P^k	d	y_0
1	(2,4)	3	9
2	(5,9)	9	8
3	(8,8)	8	10
4	(10,9)	2	0
5	(3,5)	1	2
6	(7,2)	4	7
7	(7,9)	1	2
8	(3,6)	2	0
9	(10,2)	8	10
10	(8,3)	9	8
11	(5,2)	3	9
12	(2,7)	∞	
13	O	∞	

Elliptic Curve Cryptography

- Elliptic Curve Discrete Logarithm Problem (ECDLP)
  - Gegeben:

    $y^2 = x^3 + ax + b \bmod p$ und Basispunkt P
  - $Q=P_k$ (Punktiteration)
  - Frage: Ist es möglich, k zu einem gegebenen Punkt Q zu berechnen?
  - Antwort: Schwieriges Problem!

EC Cryptosystem

- Gegeben: Kurve, Basispunkt P und Primzahl p

- Verschlüsselung: $Q_A = P^a$
- Entschlüsselung: $Q_B = P^b$

- Geheimnis: $S = Q_A{}^b = Q_B{}^a = P^{ab}$

Elliptic Curve Cryptography

- Einige EC Cryptosysteme sind genormt:
  - ECDH (Elliptic Curve Diffie-Hellman) für geheimen Schlüsselaustausch
  - ECIES (Elliptic Curve Integrated Encryption Scheme) für public key Verschlüsselung
  - ECDSA (Elliptic Curve Digital Signature Algorithm)

Elliptic Curve Cryptography

- Equivalente Schlüssellängen

DES	56	80	112	128	192	256	512
RSA	512	1024	2048	3072	7680	15360	61440
ECC	112	160	224	256	384	512	1024
Verh.	5:1	6:1	9:1	12:1	20:1	30:1	60:1

Quantenkryptographie

- Schlüsselaustausch für andere Verfahren, idealerweise für One-Time-Pads, mittels Quanten
- Hört ein Angreifer eine Übertragung ab, soll er entdeckt werden
- Zwei grundsätzliche Verfahren möglich
  - BB84-Protokoll (Übertragung mittels einzelner Quanten)
  - Ekert Protokoll (Übertragung mittels verschränkter Zustände)

Beispiel BB84 - Funktionsweise

- 1984 von Charles H. Bennett und Gilles Brassard vorgeschlagen
- Informationen werden mittels Photonen übertragen
- Photonen können horizontal oder vertikal polarisiert sein
  - –
  - |

BB84 - Funktionsweise

- Ein horizontal polarisiertes Photon wird durch einen vertikalen Filter nicht durchgelassen, durch einen horizontalen Filter mit Sicherheit
- Photonen können auch verschiedenartig diagonal polarisiert sein
  - /
  - \

BB84 - Funktionsweise

- Dies ist messbar, indem der Filter einfach um 45° gedreht wird
- Empfänger (Bob) erhält nur mit 50-prozentiger Wahrscheinlichkeit ein richtiges Meßergebnis, wenn er einen anders polarisierten Filter (x oder +) verwendet als der Sender (Alice)
- Alice erzeugt ein Photon mit einer von ihr gewählten Polarisation
  - –,|,/ oder \
- Alice sendet Photon zu Bob
- Bob misst es in einem von ihm zufällig gewählten Filter
  - x oder +

BB84 - Funktionsweise

- Diese Prozedur wird so oft wiederholt, bis Alice und Bob eine ausreichende Anzahl von Werten erhalten haben, die sie in Bits umsetzen können

- Alice muss dabei während des Ablaufs darauf achten, dass sie alle vier Polarisationsmöglichkeiten mit gleicher Wahrscheinlichkeit erzeugt, und Bob sollte seinen Filter ebenfalls mit gleicher Wahrscheinlichkeit auswählen

BB84 - Funktionsweise

- Alice und Bob verständigen sich über eine (unsichere aber authentifizierte) Leitung, in welchen Fällen sie den gleichen Filter verwendet haben
  - Bei diesen können sie sicher sein, dass sie die gleichen Polarisationsrichtungen gemessen haben
  - Aus den erhaltenen Werten soll Schlüssel erzeugt werden
- Zuweisung von Bitwerten zur Polarisation
  - Zum Beispiel 0 für horizontale Polarisation (–) oder Polarisation von links oben nach rechts unten (\), 1 für vertikale Polarisation (|) oder Polarisation von rechts oben nach links unten (/)

BB84 - Funktionsweise

- Die Polarisationen, für die sie den gleichen Filter verwendet haben, liefern Alice und Bob das gleiche Bit, können also für einen Schlüssel verwendet werden
- Die restlichen Bits sind nur mit 50-prozentiger Wahrscheinlichkeit richtig und werden deshalb verworfen
- Im Mittel können Alice und Bob also die Hälfte aller Bits für die Schlüsselerstellung weiterverwenden

Beispiel

Von Alice gesendete Polarisation	/	/	/	\	\	\	–	–	–	|	|	|
Von Bob verwendete Basis	x	+	+	x	+	+	x	x	+	x	x	+
Von Bob gemessene Polarisation	/	\	/	\	\	/	–	|	–	–	|	|
Basis gleich?	J	N	N	J	N	N	N	N	J	N	N	J
Verwendeter Schlüssel	I	·	·	0	·	·	·	·	0	·	·	I

Kleiner Lauschangriff

- Quantenmechanische Effekte werden genutzt, um Lauschangriffe zu entdecken
  - Durch seine Messung ändert ein Angreifer unter Umständen den von Alice gesendeten Basiszustand
- Einfacher Weg zum Mithören
  - Eva fängt jedes Qubit ab
  - Eva misst es in einer der zwei möglichen Basen
  - Eva schickt das gemessene Ergebnis anschließend weiter zu Bob

Kleiner Lauschangriff

- Da Alice und Bob nur die Bits weiterverwenden, bei denen sie die gleiche Basis verwendet haben, gibt es hier zwei mögliche Fälle:
  - Eve misst in der gleichen Basis wie Alice: in diesem Fall bemerken Alice und Bob nichts, und Eve kennt das Bit.
  - Eve misst in der anderen Basis: in diesem Fall stört Eve Bobs Messung, da sie den Zustand des Qubits verändert, so dass er in der Hälfte aller Fälle ein falsches Bit empfängt.

Kleiner Lauschangriff

- Eve kennt zum Zeitpunkt ihrer Messungen weder Alices noch Bobs Basis, daher kommen beide Fälle gleichhäufig vor
  - Im Mittel sind 25 Prozent aller Bits fehlerhaft
  - Feststellung durch Auswahl einiger Bits von Alice und Bob und Vergleich über den unsicheren Kanal
  - Abschätzung der Fehlerrate (mittels statistischer Tests)
  - Ist Fehlerrate zu hoch (also im Beispiel nahe 25%), so müssen sie befürchten, dass gelauscht wurde

Zusammenfassung

- Informationen können einen enormen Wert haben, den es zu schützen gilt.

- Ziel der Kryptographie ist es, das zugrundeliegende Problem ohne Kenntnis des Schlüssels schwer und mit seiner Kenntnis einfach lösen zu können.

Zusammenfassung

- Kryptographische Verfahren lassen sich in eine Reihe von Klassen einteilen. Diese sind:
  - Substitution / Transposition
  - Asymmetrische Verschlüsselung / Symmetrische Verschlüsselung
  - Blockverfahren / Stromverfahren

- Kryptographischen Verfahren liegen komplexe mathematische Zusammenhänge zugrunde.

Datensicherheit – Signaturen

Thomas Mundt
thm@informatik.uni-rostock.de

Inhalt

- Kryptographie
  - Verschlüsselung
  - Digitale Signaturen

Ziele

- Vermittlung der Grundlagen der Kryptographie
- Einteilung der verfügbaren Verfahren

- Hier speziell auf elektronische Signaturen bezogen

Anforderungen

- Erinnerung an einige „Grundbedürfnisse“ und Realisierungsmöglichkeiten mit Hilfe digitaler Signaturen
  - Identifizierung / Authentifikation
  - Unbestreitbarkeit und Nachvollziehbarkeit
- Digitale Signaturen sollen die gleichen Anforderungen erfüllen, wie konventionelle Signaturen
  - § 126 BGB Schriftform
    - (3) Die schriftliche Form kann durch die elektronische Form ersetzt werden, wenn sich nicht aus dem Gesetz ein anderes ergibt.

Anforderungen

- § 126a BGB Elektronische Form
  - (1) Soll die gesetzlich vorgeschriebene schriftliche Form durch die elektronische Form ersetzt werden, so muss der Aussteller der Erklärung dieser seinen Namen hinzufügen und das elektronische Dokument mit einer qualifizierten elektronischen Signatur nach dem Signaturgesetz versehen.
  - (2) Bei einem Vertrag müssen die Parteien jeweils ein gleichlautendes Dokument in der in Absatz 1 bezeichneten Weise elektronisch signieren.

Anforderungen

- EG „Signaturrichtlinie“ Artikel 2 Nr. 2 fordert für **fortgeschrittene elektronische Signaturen**, dass diese:
  - a) ausschließlich dem Unterzeichner zugeordnet sind,
  - b) die Identifizierung des Unterzeichners ermöglichen,
  - c) mit Mitteln erzeugt werden, die der Unterzeichner unter seiner alleinigen Kontrolle halten kann, und
  - d) mit den Daten, auf die sie sich beziehen, so verknüpft sind, dass eine nachträgliche Veränderung der Daten erkannt werden kann.

Nebenbei

- Die genannten Anforderungen sind teilweise weitreichender als diejenigen an „konventionelle“ Unterschriften

- Probleme bei Unterschriften
  - Lesbarkeit
  - Identifizierbarkeit
  - Fälschungssicherheit

Unterschriften berühmter Menschen

Lösung

- Elektronische Signaturen lassen sich mit Public-Key-Verfahren realisieren
- Diesmal Verschlüsselung mit dem privaten Schlüssel
- Anhängen des verschlüsselten Textes an die Nachricht
- Kontrolle mit dem öffentlichen Schlüssel

$$C = f(K_{private}, P)$$
$$P = f(K_{public}, C)$$

- Nur der Urheber kennt den eigenen privaten Schlüssel

Umsetzung

- a) ausschließlich dem Unterzeichner zugeordnet sind
  - Eindeutige ID in den verwendeten Zertifikaten
  - Sichere Aufbewahrung des privaten Schlüssels
  - Eventuell Aufbewahrung auf einer SmartCard
  - Keine Zweitschlüssel
  - Sicheres Verfahren

Umsetzung

- b) die Identifizierung des Unterzeichners ermöglichen
  - ID des Unterzeichners ablegen
  - Öffentlicher Schlüssel wird selbst signiert und von einem vertrauenswürdigen Dritten herausgegeben
    - Ein solcher öffentlicher Schlüssel heißt dann Zertifikat
    - Hohe gesetzliche Anforderungen an offizielle Zertifizierungsstellen
  - Einsatz einer Public Key Infrastructure (PKI) in Unternehmen

Umsetzung

- c) mit Mitteln erzeugt werden, die der Unterzeichner unter seiner alleinigen Kontrolle halten kann
  - Kein Zweitschlüssel
  - Ausreichend viele verschiedene Schlüssel
    - Müssen zufällig, also nicht vorhersagbar, erzeugt werden
  - SmartCards zur Aufbewahrung der Schlüssel und zur Durchführung der Signatur
  - Sperrlisten für verlorene Zertifikate, also für Zertifikate deren korrespondierender privater Schlüssel öffentlich wurde

Umsetzung

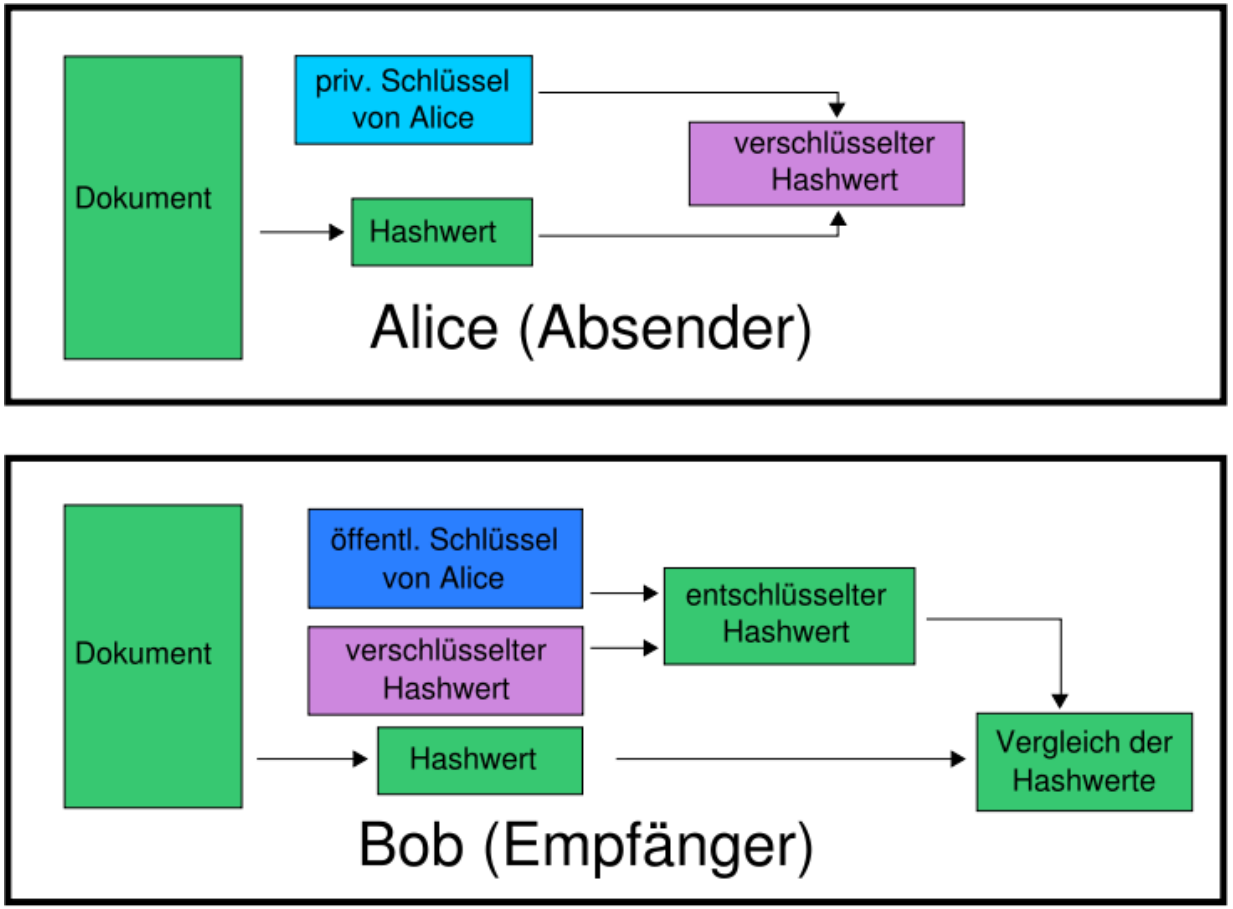
- d) mit den Daten, auf die sie sich beziehen, so verknüpft sind, dass eine nachträgliche Veränderung der Daten erkannt werden kann
  - Daten werden als Plaintext mit dem genannten Verfahren verschlüsselt
  - Änderungen an den ursprünglichen Daten können im Ciphertext ohne den privaten Schlüssel nicht nachvollzogen werden

Praktisches Problem

- Problem
  - Nachrichtenlänge beim Anhängen der Signature ca. zweimal so groß
- Lösung
  - Bilden eines Hash-Wertes

Ablauf

- Hier mit Hashwerten



Ablauf

- Der Absender (im Beispiel: Alice) wählt den zu signierenden Plaintext aus
- Die Signatur-Software bildet über die Nutzdatei einen Hash-Wert
  - Optional, dient der Verkleinerung der Datenmenge
- Der Absender bildet aus dem Hash-Wert (oder Plaintext) mit Hilfe des geheimen Schlüssels die elektronische Signatur
- Der Absender verschickt Plaintext und Signatur
- Der Empfänger (im Beispiel: Bob) erhält den Plaintext und die Signatur

Ablauf

- Der Empfänger verifiziert die Signatur mit Hilfe des öffentlichen Schlüssels und des Plaintexts
  - Authentisierung des öffentlichen Schlüssels muss durchgeführt werden
  - Einsatz von Zertifikaten
- Ist die Prüfung erfolgreich, dann wurde die Datei vom richtigen Absender verschickt (Authentifizierung, Authentizität) und nicht verändert (Integrität)
  - Absender ist im Besitz des privaten Schlüssels

Kryptographische Probleme

- Gültigkeitsdauer von Zertifikaten beschränken
  - In Zukunft höhere Rechenleistung verfügbar
  - Durchprobieren aller Schlüssel in endlicher Zeit möglich
  - Zeitstempeldienst
  - Nachsignieren vor Ablauf des ursprünglichen Zertifikats mit neuem Zertifikat
- Überprüfung der Signatur erfordert eine Echtheitsprüfung des öffentlichen Schlüssels (also des Zertifikats)
  - Zertifikat der Zertifizierungsstelle muss geeignet authentisiert werden

Offen geblieben

- Hash
- Zeitstempel
- Zertifikate
  - PKI
  - Sperrlisten / Certificate Revocation Lists
- Anwendungen
  - VPN
  - Login allgemein
  - Elektronischer Geschäftsverkehr

Zusammenfassung

- Elektronische Signaturen können konventionelle Unterschriften im Geschäftsverkehr ersetzen

- Beim Einsatz elektronischer Signaturen sind eine Reihe von Problemen zu beachten
  - Sicherer Umgang mit dem privaten Schlüssel ist erforderlich
  - Authentisierung des Benutzer-Zertifikats ist notwendig
  - Private Schlüssel sind in endlicher Zeit berechenbar, also ist eine Gültigkeitsbeschränkung der Zertifikate und ggfs. eine Nachsignierung notwendig

Datensicherheit – Hash

Thomas Mundt
thm@informatik.uni-rostock.de

Inhalt

- Kryptographie
  - Verschlüsselung
  - Digitale Signaturen

Ziele

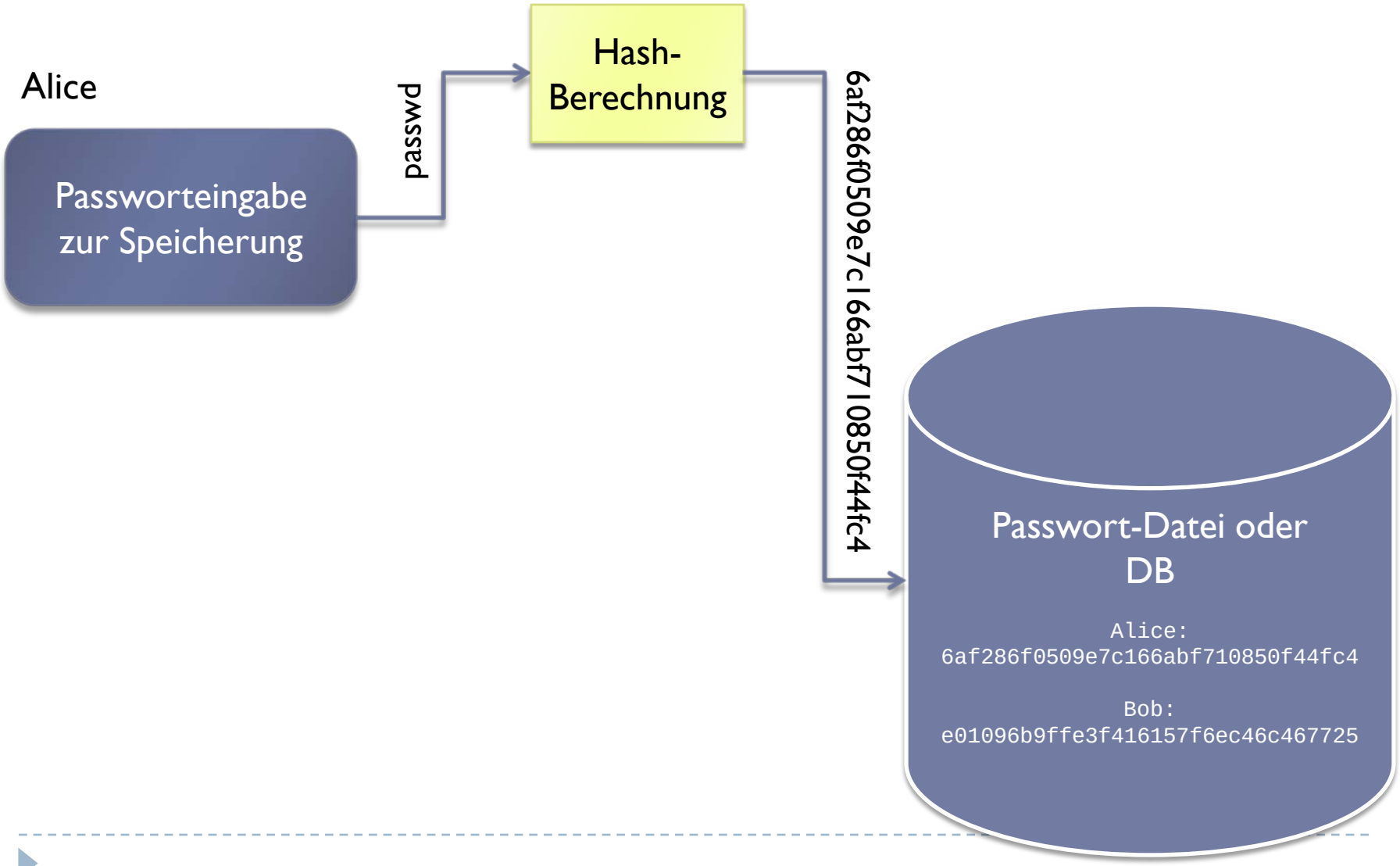
- Erkennen der Nützlichkeit und Funktionsweise von Hash-Funktionen
- Erweiterte Anforderungen an kryptographische Hash-Funktionen

Anforderungen

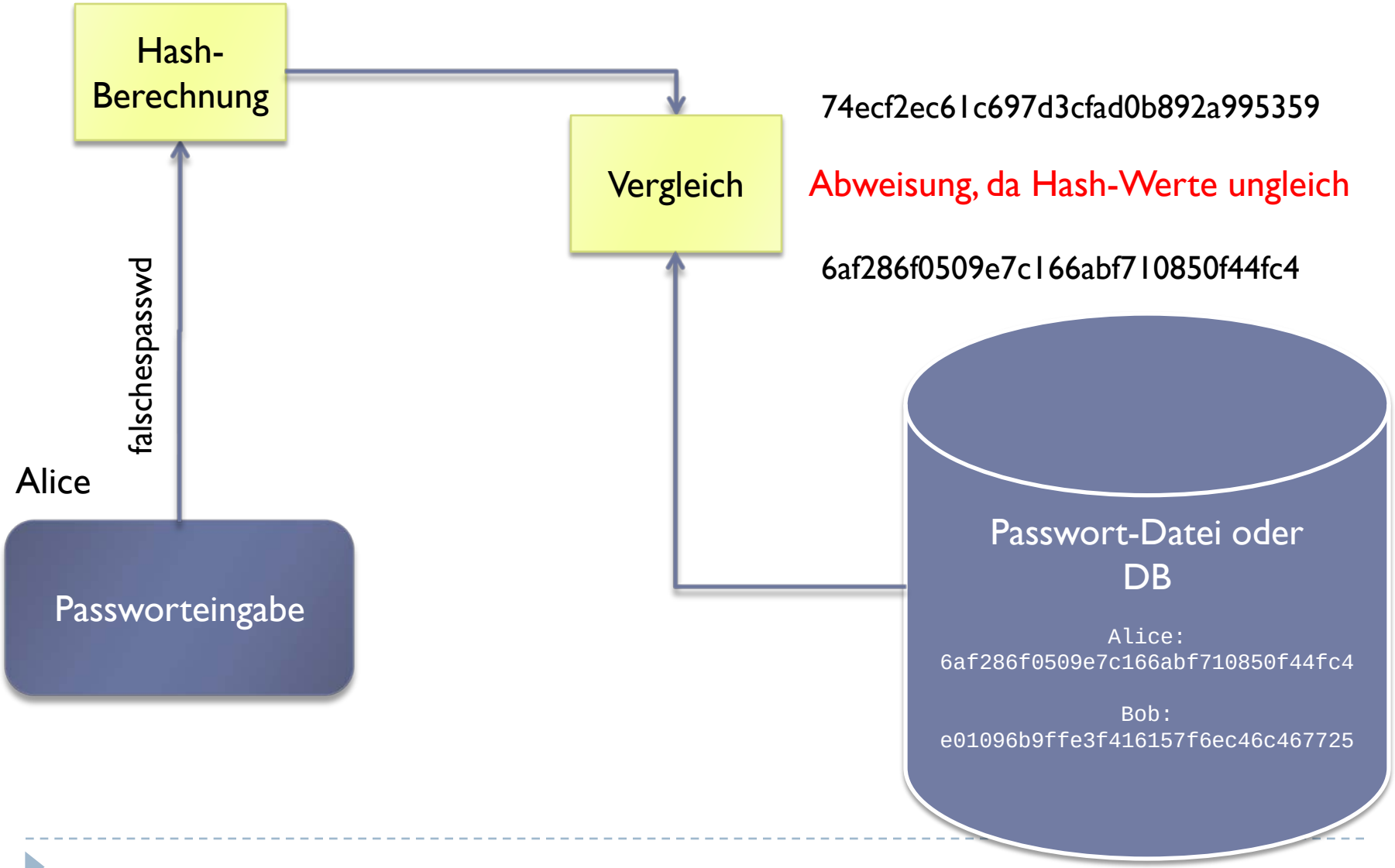
- Eine Hash-Funktion soll zu einer Eingabe beliebiger Länge einen kurzen, möglichst eindeutigen „Fingerabdruck“ fester Länge liefern

- Kryptographische Hash-Funktionen sollen wie folgt eingesetzt werden
  - Zur Einweg-Verschlüsselung, z.B. von Passwort-Listen
  - Zur Bildung von Message Digests, z.B. bei Signaturen

Einwegverschlüsselung



Passwortüberprüfung



Weitere Einsatzmöglichkeiten

- Datenbank-Index
  - Hashtabellen
  - Datenbankindex
- P2P-Netzwerke (ähnlich einer verteilten Datenbank)
  - Zur Indizierung der Dateien
- Prüfsummen
  - Redundanz zur Fehlererkennung

Hash-Funktionen

- Abbildung $h : K \rightarrow S$ heißt Hash-Funktion, wenn $|K| \geq |S|$
- Typischerweise wird *h* nur auf Teilmenge $K' \subseteq K$ angewendet
- Tatsächlich genutzte Schlüssel sind $S' := \{h(k) | k \in K'\}$
- Belegungsfaktor $\beta = \frac{|S'|}{|S|}$
- Kollision $k \neq k' \wedge h(k) = h(k')$

Anforderungen an Hash-Funktionen

- Datenreduktion
  - Größe (Speicherplatzbedarf) des Hash-Wertes soll deutlich kleiner sein als die der Nachricht
- Chaotisch
  - Ähnliche Quellelemente sollen zu völlig verschiedenen Hash-Werten führen
  - Im Idealfall führt Änderung eines Bits in der Eingabe zu einer Änderung der Hälfte der Bits im Ergebnis
- Surjektiv
  - Jedes mögliche Ergebnis aus dem Wertebereich soll tatsächlich vorkommen können

Anforderungen an Hash-Funktionen

- Effizienz
  - Schnell berechenbar sein
  - Wenig Ressourcen-Bedarf

Zusatzanforderungen

- Kryptographisch sichere Hash-Funktionen müssen zwei weitere, speziellere Eigenschaften bieten

- Kollisionsfreiheit
  - Es darf nicht effizient möglich sein, zwei Quellelemente mit demselben Hash-Wert zu finden
- Unumkehrbarkeit
  - Zu der Funktion gibt es keine effizient berechenbare inverse Funktion, mit der es möglich wäre, für ein gegebenes Zielelement ein passendes Quellelement zu finden

Beispiel

- `> echo "Tom demonstriert eine Hash Funktion." | md5sum`
  `ca6d81f2ca377e4c65f4511c385bf4c8`
- `1100 1010 0110 1101 1000 0001 1111 0010`
  `1100 1010 0011 0111 0111 1110 0100 1100`
  `0110 0101 1111 0100 0101 0001 0001 1100`
  `0011 1000 0101 1011 1111 0100 1100 1000`
- `> echo "Tim demonstriert eine Hash Funktion." | md5sum`
  `048a135ab39f87dde7fcddb91613eade`
- `0000 0100 1000 1010 0001 0011 0101 1010`
  `1011 0011 1001 1111 1000 0111 1101 1101`
  `1110 0111 1111 1100 1101 1101 1011 1001`
  `0001 0110 0001 0011 1110 1010 1101 1110`

Unterschiede

- **11**00 **101**0 **011**0 1**101** **1**00**0** 000**1** **1111** **0**010
  1**100** **1**010 **00**11 **0**111 **0111** **111**0 **0**10**0** 110**0**
  0110 01**0**1 1111 **0**100 **0**101 **00**01 **000**1 **11**00
  00**11** **100**0 0**1**01 **1**011 111**1** **010**0 110**0** **1**00**0**

- **00**00 **010**0 **100**0 1**010** **0**001 00**11** **0**10**1** **1**010
  1**011** **0**011 **10**01 **1**111 **1000** **0**111 **1**10**1** 110**1**
  1110 011**1** 1111 **1**100 **1**101 **11**01 **101**1 1**00**1
  00**0**1 **011**0 **0**001 **0**011 111**0** **101**0 110**1** 11**1**0

Hash-Funktionen

- Quersumme
  - Hash-Funktion, aber nicht kryptographisch sicher
- Letzte Ziffer
  - Hash-Funktion, aber nicht kryptographisch sicher
- Modulo-Operation
  - Hash-Funktion, aber nicht kryptographisch sicher

Kryptographische Hash-Funktionen

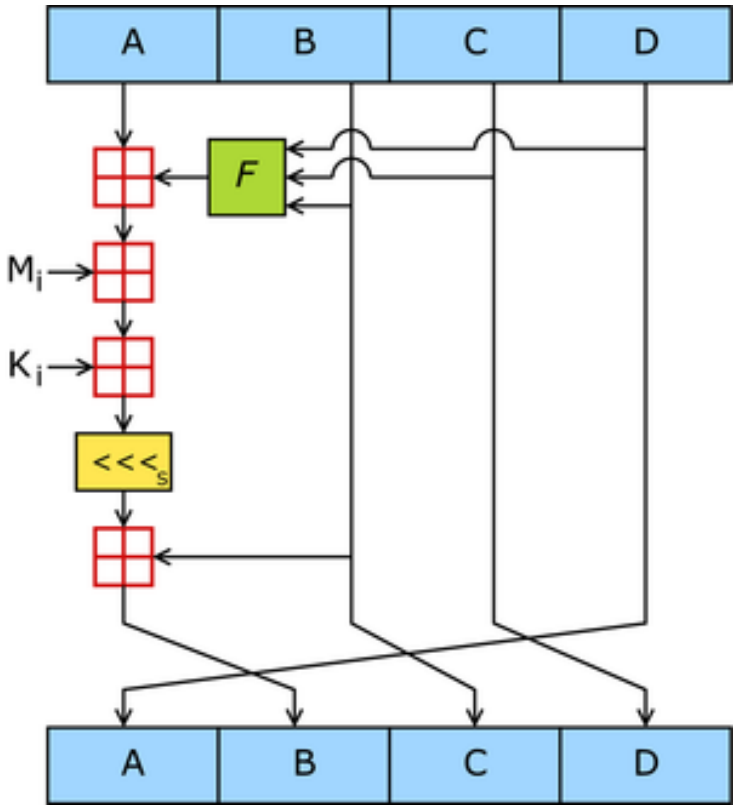
- Message Digest 5 (MD5)
  - RFC 1321
  - 1991 von Rivest vorgeschlagen
  - Erzeugt 128 bit lange Hash-Werte
  - Einsatz für Prüfsummen und früher als Einweg-Verschlüsselung von Passwörtern, z.B. unter Unix
  - Erzeugung von Kollisionen inzwischen sehr einfach
- Secure Hash Algorithm 1 (SHA-1)
  - 1995 von NIST vorgeschlagen
  - Erzeugt 160 bit lange Hash-Werte

MD5

- Nachricht wird in 512bit lange Blöcke aufgeteilt (eventuell mit Pad verlängert)
- A, B, C, D sind jeweils 32-bit-Wörter, vor der 1. Runde mit Konstanten vorbelegt
- M_j ist Nachricht, pro Runde werden 32bit verarbeitet
- K_j ist Konstante für jede Runde
- F ändert sich pro Runde

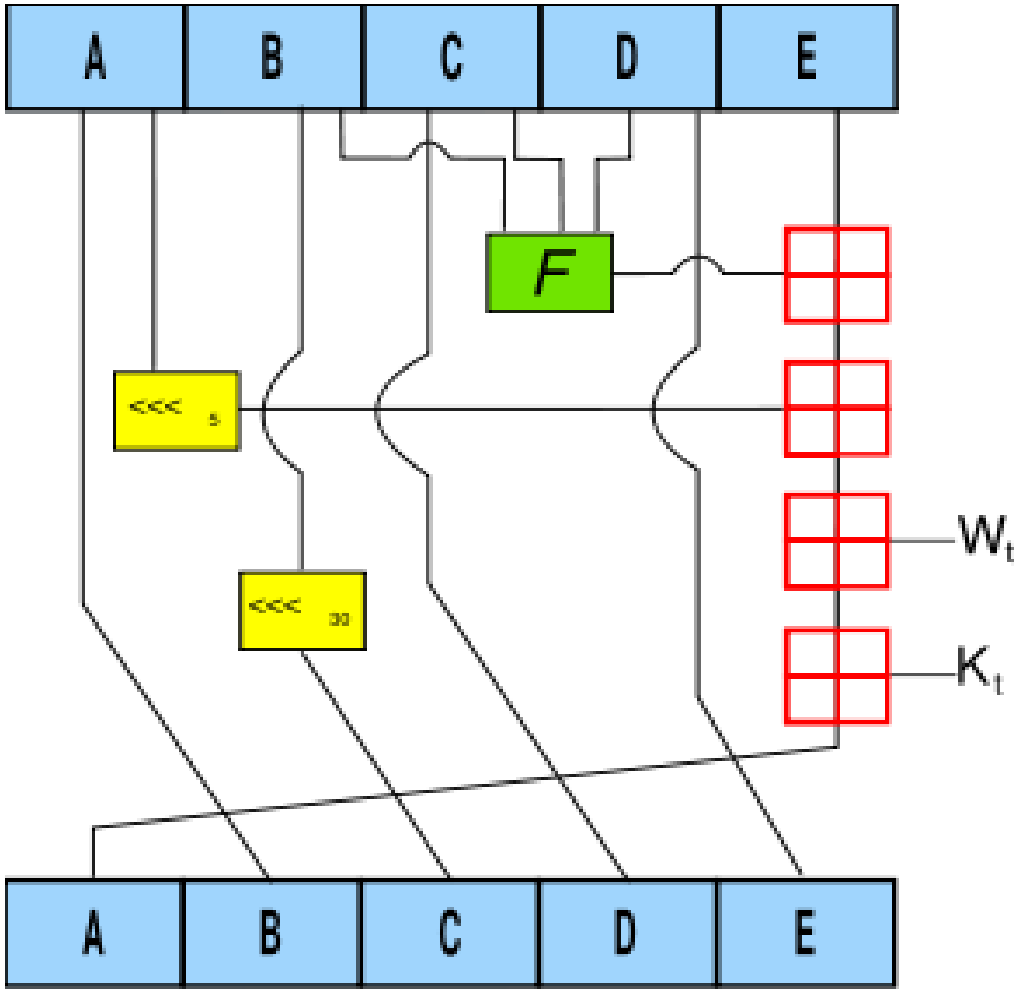
$$F1(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$F2(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$
$$F3(X,Y,Z) = X \oplus Y \oplus Z$$
$$F4(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

- $\boxplus$: Addition mod 2^{32} ,<<<: Rotation



Wird 16 mal pro Runde ausgeführt, es gibt 4 Runden (4 x 128 bit)

SHA-1



Zusammenfassung

- Allgemeine Hash-Funktionen verkürzen einen Eingabewert auf eine feste Länge.
- Kryptographisch sichere Hash-Funktionen lassen sich zur Einwegverschlüsselung und als Prüfsummen verwenden.

Zusammenfassung

- Hash-Funktionen müssen folgende Eigenschaften haben:
  - Datenreduktion
  - Chaotisch
  - Surjektiv
  - Effizient berechenbar
- Kryptographisch sichere Hash-Funktion zusätzlich:
  - Kollisionsfreiheit
  - Unumkehrbarkeit

Datensicherheit - Zufallszahlen

Thomas Mundt
thm@informatik.uni-rostock.de

Inhalt

- Kryptographie
  - Verschlüsselung
  - Digitale Signature

Ziele

- Vermittlung von Möglichkeiten zur Erzeugung von Zufallszahlen

- Risiken beim Umgang mit Zufallszahlengeneratoren

Anforderungen

- Viele kryptographische Verfahren benötigen Zufallszahlen
  - Schlüsselerzeugung (z.B. One-Time-Pads)
  - Challenge-Response-Verfahren
- Vorhersagbarkeit führt zu einer Beschränkung des Wertebereichs
- Gute Zufallszahlen wichtig
  - Zahlen dürfen nicht vorhersagbar sein
  - Vorhersagbar, berechnete Pseudozufallszahlen ungeeignet

Zufallszahlengeneratoren

- Ausnutzung von physikalisch zufälligen Prozessen / Nichtdeterministische Zufallszahlengeneratoren
  - Geigerzähler nutzen radioaktiven Zerfall als Zufallsquelle aus
    - Zerfall ist von Umgebung unabhängig
    - Durchschnittliche Zerfallsrate ist für ein Isotop konstant
  - Thermisches Rauschen
    - Umkippen eines Oszillators
  - Umgebungsrauschen
    - Zusammenfassung einiger Ereignisse
      - Nutzerinteraktionen (Keyboard, Mouse)
      - Geräte-Interupts

Implementierung

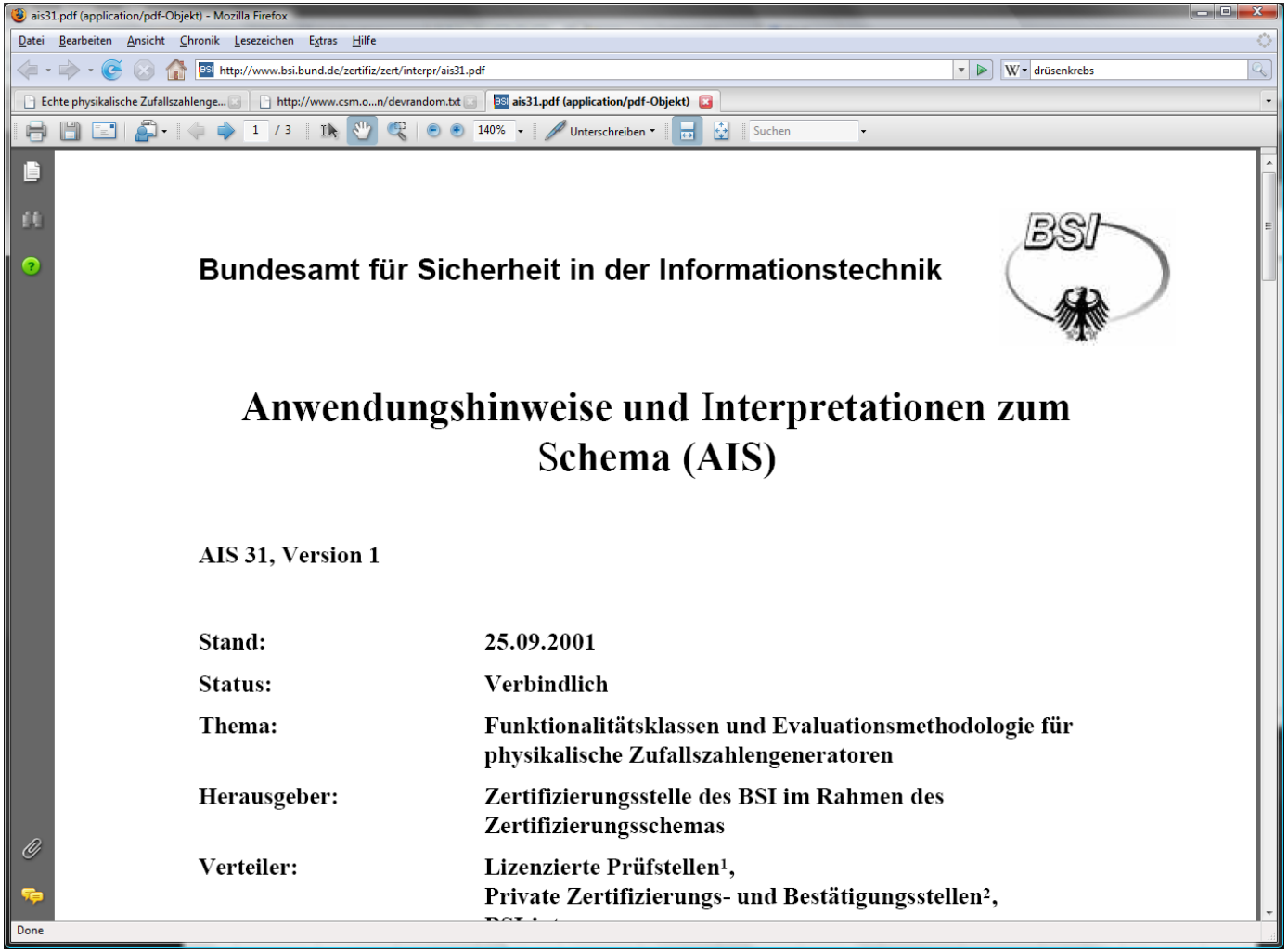
- /dev/random benutzt eine Reihe von Zufallsquellen
  - * void add_keyboard_randomness(unsigned char scancode);
  - * void add_mouse_randomness(__u32 mouse_data);
  - * void add_interrupt_randomness(int irq);
  - * void add_blkdev_randomness(int irq);
- /dev/random blockiert, wenn Entropiepool erschöpft ist

/dev/random

```
[root@sol1 ~]# hexdump /dev/random
0000000 3a90 4ac9 70d8 0521 2498 2759 6e8e 826c
0000010 e702 358b 59e8 4ff4 9a89 736d ba7d 603c
0000020 4dc1 df8c 8deb d8f8 e9d7 0e13 409a 247e
0000030 faf6 0c48 19d2 28d2 fdd3 dfb0 66aa 0301
0000040 96ff dcfd 2223 833a 6b13 39cf db04 0f1f
0000050 b740 ae8d 120d 807f 8927 5c54 354a d89e
0000060 31c4 d5ce 2d89 3361 663f 66aa 2cb1 c391
0000070 96fb 4097 2c3c fc01 eed8 b145 8eb6 1d4c
0000080 5b62 5180 1fd0 5ae9 cd76 090d 428c 6486
0000090 b4a8 9f06 1acb 6344 e8d5 4112 71a9 2deb
```

Implementierung

Zertifizierung



Zufallszahlengeneratoren

- **Deterministische Zufallszahlengeneratoren**
  - Zufallszahlen können später reproduziert werden
    - Sinnvoll für Tests
    - Anerkennung von wissenschaftlichen Experimenten
  - Arithmetische Zufallszahlengeneratoren
    - Verwenden irrationale Zahlen
    - Beispiel 1 - Allgemeine Rekursion
      - Saat $y_0, y_1, \ldots, y_{k-1}$
      - Funktion $f : \{0, \ldots, m-1\}^k \rightarrow \{0, \ldots, m-1\}$
      - Erzeugte Werte $y_n := f(y_{n-k}, y_{n-k+1}, \ldots, y_{n-1})$ (für $n \geq k$)
      - Normalisiert $u_i := \frac{y_i}{m}$

Zufallszahlengeneratoren

- Beispiel 2 - Linearer Kongruenzgenerator
  - Modul $m \in \{2, 3, 4, ...\}$
  - Faktor $a \in \{1, ..., m-1\}$
  - Inkrement $b \in \{1, ..., m-1\}$
  - Startwert $y_1 \in \{0, ..., m-1\}$

$$y_i = (ay_{i-1} + b) \bmod m$$

  - Nach wenigen Schritten können a und b berechnet werden
  - Kryptographisch unbrauchbar

- Andere Beispiele
  - Fibonacci-Generator
  - Mersenne-Twister

/dev/urandom

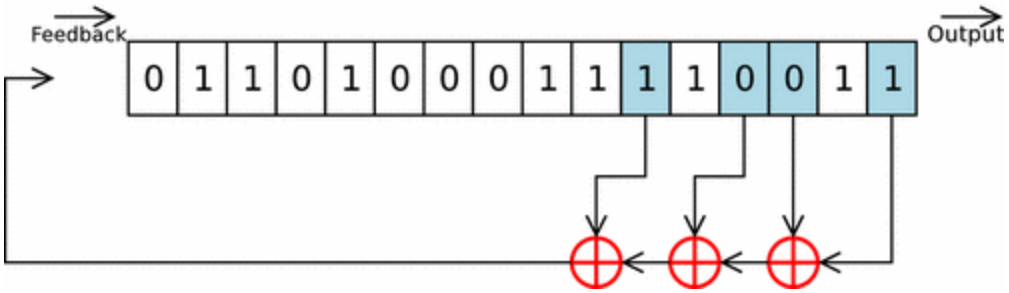
```
[root@sol1 ~]# hexdump /dev/urandom
0000000 345f 7346 92d6 fa5f 803a 9747 33c8 77fb
0000010 0da9 5485 5923 6851 aa69 01b6 4cc2 760f
0000020 84cb a850 b639 3e71 e048 daef 8cf2 ce69
0000030 de09 99de b21a 891e 87d3 c823 118e 921d
0000040 7d45 91ac a100 6c94 e0e3 fe00 ef2a f43c
0000050 0d04 ecc7 6467 4487 095d a7a0 0d77 36d9
```

Beispiel für schlechte Generatoren

- RC4 in Netscape früher mit 40 (international) bzw. 128 bit (USA) Schlüssel (3,4 *10^{38} verschiedene Schlüssel)
- Sitzungsschlüssel wird unter HP-UX und Solaris mit Hilfe der Systemzeit (auf Mikrosekunden genau), der PID und PPID erzeugt (10^{18} Möglichkeiten)
- PID und PPID leicht mit `ps -ef` auslesbar
- `date` liefert Zeit auf Sekunden genau, auf Minute genau, kann man schätzen
- Es blieben nur 10^{7} Möglichkeiten übrig

Hardware

- Linear rückgekoppeltes Schieberegister



Zusammenfassung

- Zufallszahlen lassen sich nichtdeterministisch oder deterministisch erzeugen.
- Nichtdeterministische Zufallszahlengeneratoren benötigen äußere Ereignisse.
- Für den kryptographische Einsatz empfehlen sich nichtdeterministische Zufallszahlengeneratoren.

Datensicherheit - Zertifikate

Thomas Mundt
thm@informatik.uni-rostock.de

Inhalt

2. Basisverfahren

- Zertifikate

Ziele

- Sicherer Umgang mit Zertifikaten
- Kenntnis der Prozesse bei der Zertifitikatserzeugung
- Kenntnis der Protokolle bei der Zertifikatsverwaltung

Begriff

- Digitale Zertifikate sind Daten in einem vorgegebenen Format, die den öffentlichen Schlüssel einer Identität (z.B. Person) zuordnen und bestätigen.

Informationen in Zertifikaten

- Zertifikate enthalten in der Regel die folgenden Informationen:
  - Den Namen des Ausstellers des Zertifikats
  - Policy für den Umgang mit dem Zertifikat
  - Informationen zu Gültigkeitsdauer des Zertifikats
  - Öffentlichen Schlüssel
  - Name und weitere Informationen des Eigentümers
  - Digitale Signatur des Ausstellers über das Zertifikat

Standards

- Standard X.509
  - Definiert den Aufbau des Zertifikats
  - Adressen gemäß X.500
  - SSL und TLS, S/MIME, VPN, SSH
  - Verschiedene Kodierungen
- RFC 2440
  - PGP

X.509

- Zertifikat
  - Version
  - Seriennummer
  - Algorithmen ID
  - Aussteller
  - Gültigkeit
    - von
    - bis
  - Subject
  - Subject Public Key Info
    - Public Key Algorithmus
    - Subject Public Key
  - Eindeutige ID des Ausstellers (optional)
  - Eindeutige ID des Inhabers (optional)
  - Erweiterungen
- Zertifikat Signaturalgorithmus
- Zertifikat Signatur

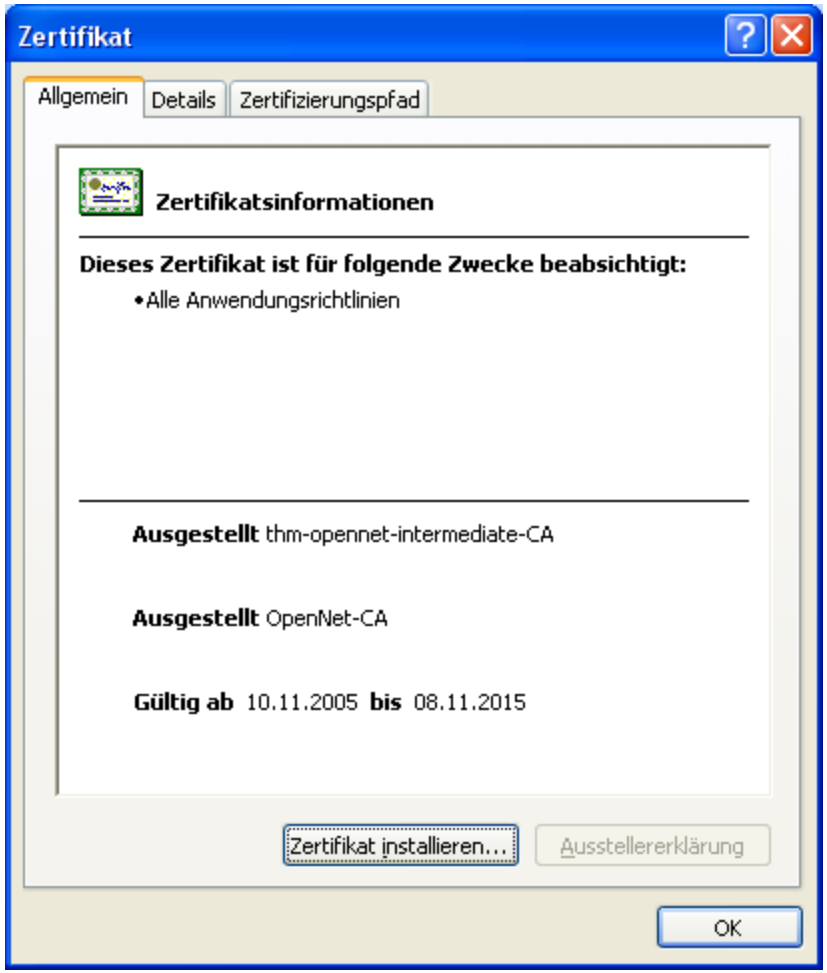
Beispiel X.509 Cert

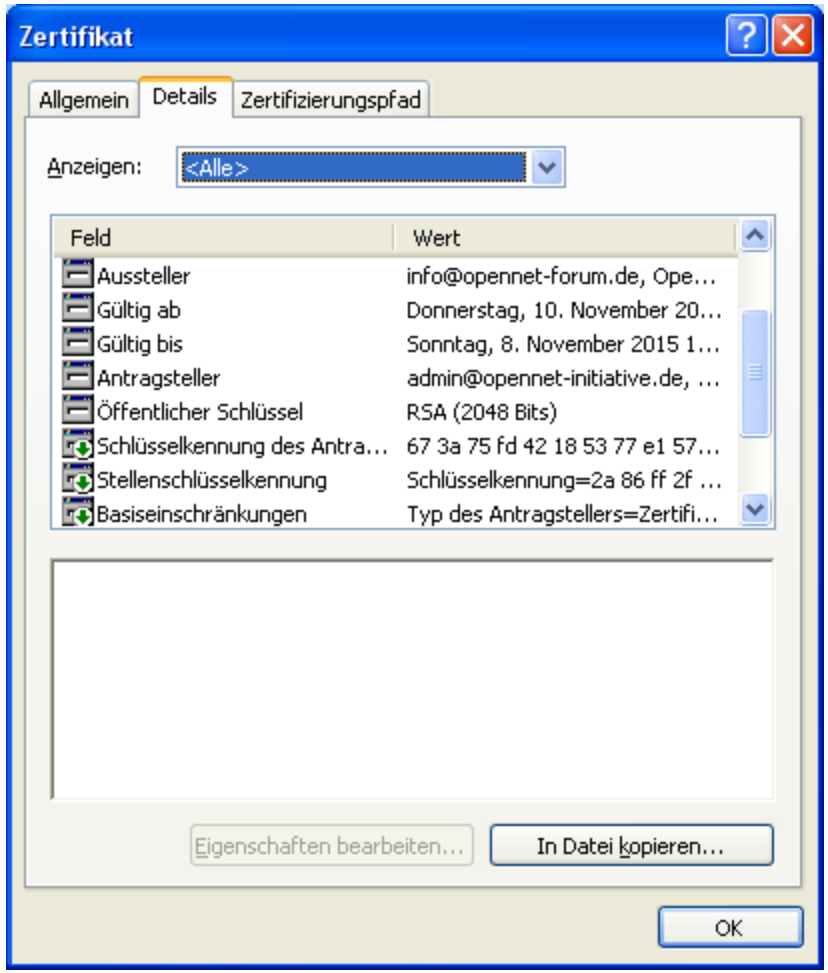
```
titan:/etc/openvpn/auth/opennet/certs# cat thm.crt
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 105 (0x69)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=DE, ST=Mecklenburg-Vorpommern,
      L=Rostock, O=OpenNet, CN=OpenNet-
      CA/emailAddress=info@opennet-forum.de
        Validity
            Not Before: Nov 10 11:50:21 2005 GMT
            Not After : Nov  8 11:50:21 2015 GMT
        Subject: C=DE, ST=Mecklenburg-Vorpommern,
      O=OpenNet, CN=thm-opennet-intermediate-
      CA/emailAddress=admin@opennet-initiative.de
```

Beispiel X.509 Cert

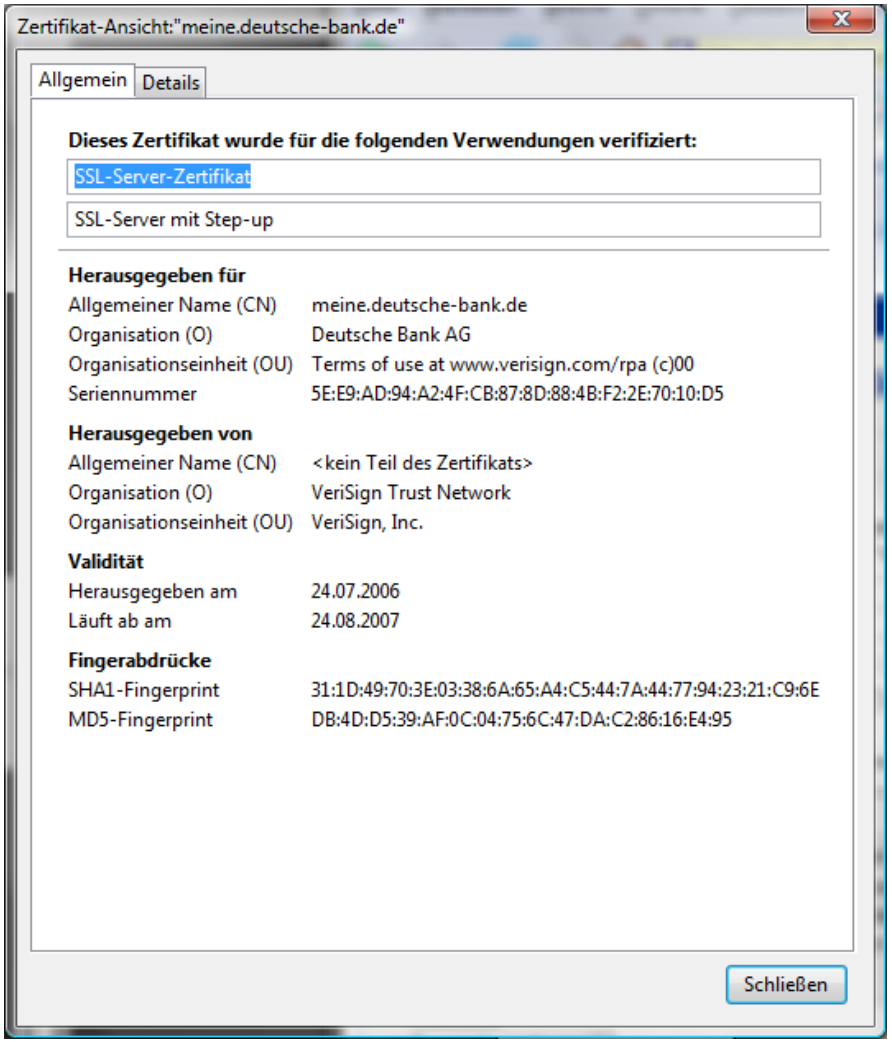
```
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
    Modulus (2048 bit): 00:cb …
    Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Subject Key Identifier:   67:…
        X509v3 Authority Key Identifier: keyid:2A:86…
        DirName:/C=DE/ST=Mecklenburg-
Vorpommern/L=Rostock/O=OpenNet/CN=OpenNet-
CA/emailAddress=info@opennet-forum.de
serial:00
    X509v3 Basic Constraints: CA:TRUE
```

Beispiel X.509 Cert





Beispiel

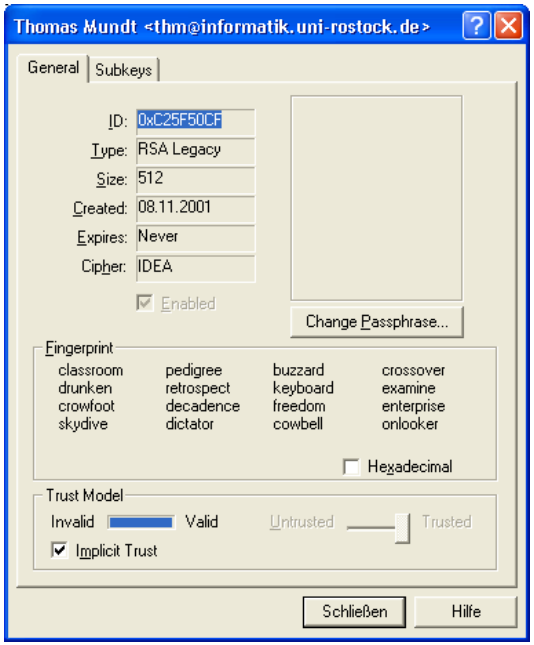

Zertifikat-Ansicht:"meine.deutsche-bank.de"
Allgemein
Details
Dieses Zertifikat wurde für die folgenden Verwendungen verifiziert:
SSL-Server-Zertifikat
SSL-Server mit Step-up
Herausgegeben für
Allgemeiner Name (CN) meine.deutsche-bank.de
Organisation (O) Deutsche Bank AG
Organisationseinheit (OU) Terms of use at www.verisign.com/rpa (c)00
Seriennummer 5E:E9:AD:94:A2:4F:CB:87:8D:88:4B:F2:2E:70:10:D5
Herausgegeben von
Allgemeiner Name (CN) <kein Teil des Zertifikats>
Organisation (O) VeriSign Trust Network
Organisationseinheit (OU) VeriSign, Inc.
Validität
Herausgegeben am 24.07.2006
Läuft ab am 24.08.2007
Fingerabdrücke
SHA1-Fingerprint 31:1D:49:70:3E:03:38:6A:65:A4:C5:44:7A:44:77:94:23:21:C9:6E
MD5-Fingerprint DB:4D:D5:39:AF:0C:04:75:6C:47:DA:C2:86:16:E4:95
Schließen

Anforderungen

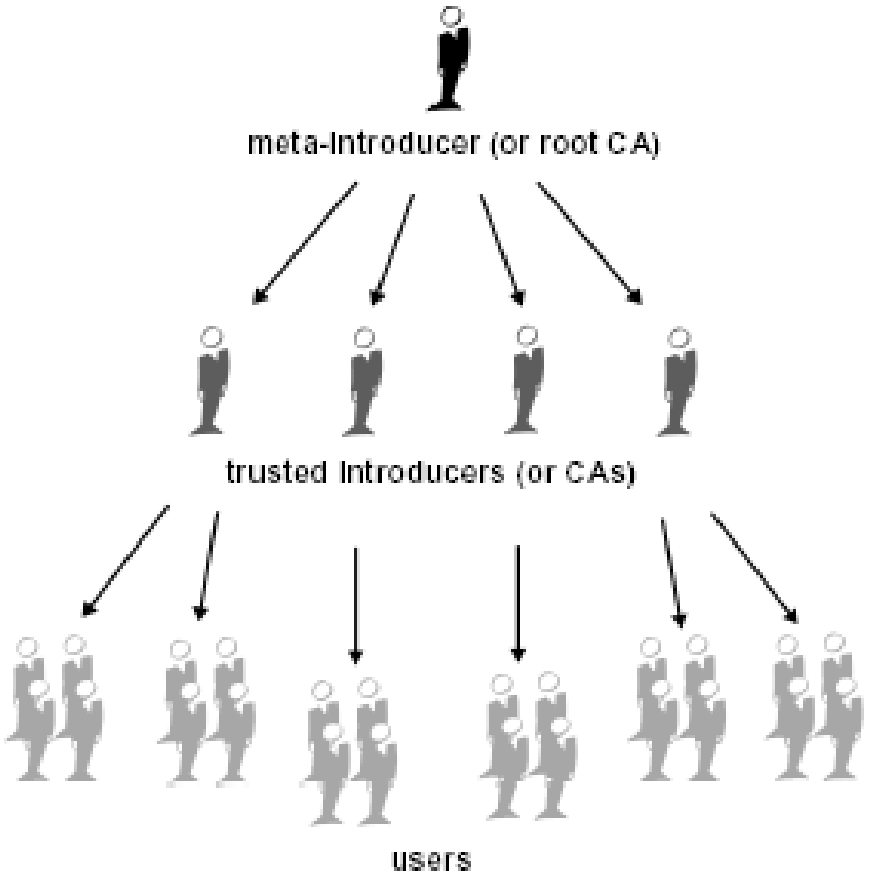
- Authentisierung des Schlüssels, entweder selbst oder durch einen vertrauenswürdigen Dritten
- Signieren des Schlüssels durch einen vertrauenswürdigen Dritten bestätigt die Authentizität
- Vertrauenswürdiger Dritter
  - Zentral
    - Certifying authority (CA)
    - Trustcenter
    - Zertifizierungsstelle
  - Dezentral
    - Web of Trust

Authentisierung

- Möglichkeiten zum Sicherstellen, dass der Schlüssel mit dem Namen „Bob“ auch zu Bob gehört
  - Persönliche Übergabe des Schlüssels
  - Rückbestätigung z.B. durch Telefonanruf und Durchgeben eines Hash-Wertes (Fingerprint) des Schlüssels
  - Signieren des Public Keys durch einen vertrauenswürdigen Dritten



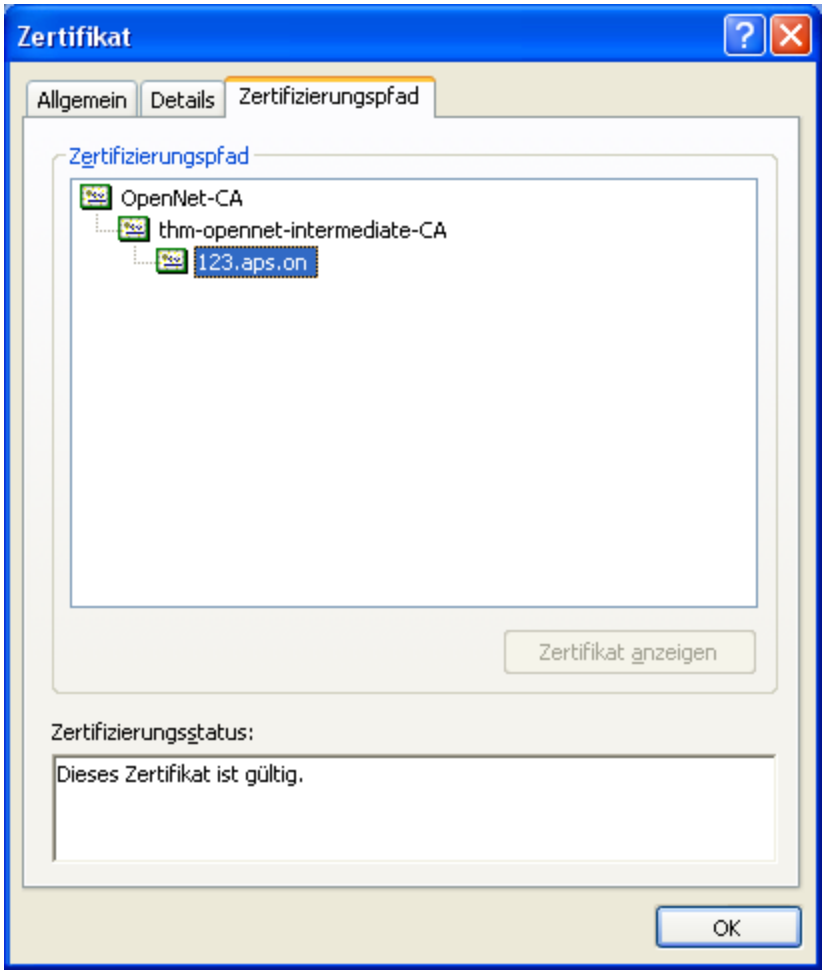
Hierarchisches Vertrauen



Phil Zimmermann, An Introduction to Cryptography

Beispiel X.509 Cert

- Mehrstufige CA ebenfalls möglich



Ablauf

- Nutzer erstellt Schlüsselpaar
  - Private Key (verbleibt immer beim Nutzer, z.B. SmartCard)
  - Public Key
- Nutzer fügt Informationen für das Zertifikat hinzu
  - Adressdaten / Identitätsdaten
- Nutzer erstellt einen Zertifizierungsantrag (Certificate Signing Request / CSR)
- CA überprüft den CSR auf Authentizität
- CA signiert den CSR und sendet CERT an Benutzer
- Aufnahme des Zertifikats in eine Datenbank

Beispiel (OpenSSL)

> openssl req -days 365 -nodes -new –keyout name.key \ -out name.csr -config /etc/ssl/openssl.cnf

Loading 'screen' into random state - done

Generating a 1024 bit RSA private key

.....++++++....++++++ writing new private key to 'keys\name.key'

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

Country Name (2 letter code) [US]:de

State or Province Name (full name) [CA]: Mecklenburg-Vorpommern

Locality Name (eg, city) [SanFrancisco]: Rostock

Organization Name (eg, company) [FortFunston]:VornameNachname

Organizational Unit Name (eg, section) []: OUN

Common Name (eg, your name or your server's hostname) []: 99.aps.on

Email Address [mail@host.domain]: du@provider.de

Beispiel (OpenSSL)

Zertifikatserstellung

```
> openssl ca -config /etc/ssl/openssl.cnf -days 365 -in csrs/name.csr -out
    certs/name.crt
```

Certifying Authority

- Dem Public Key des Trustcenters muss explizit vertraut werden
- Möglichkeiten:
  - Durch Mitliefern im Browser, Programm oder Betriebssystem (Root CA)
  - Durch Übermittlung des CA Keys auf sicherem Wege mit Feststellung der Identität
  - Durch Fingerprint und Überprüfung des Hash-Werts

Public Key Infrastructure

- Zertifizierungsstelle (Certificate Authority, CA)
  - Organisation, die das CA-Zertifikat bereitstellt und Nutzerzertifikate signiert
- Registrierungsstelle (Registration Authority, RA)
  - Organisation zur Beantragung der Zertifikate
- Zertifikatsperrliste (Certificate Revocation List, CRL)
  - Enthält IDs in Verlust geratener Zertifikate

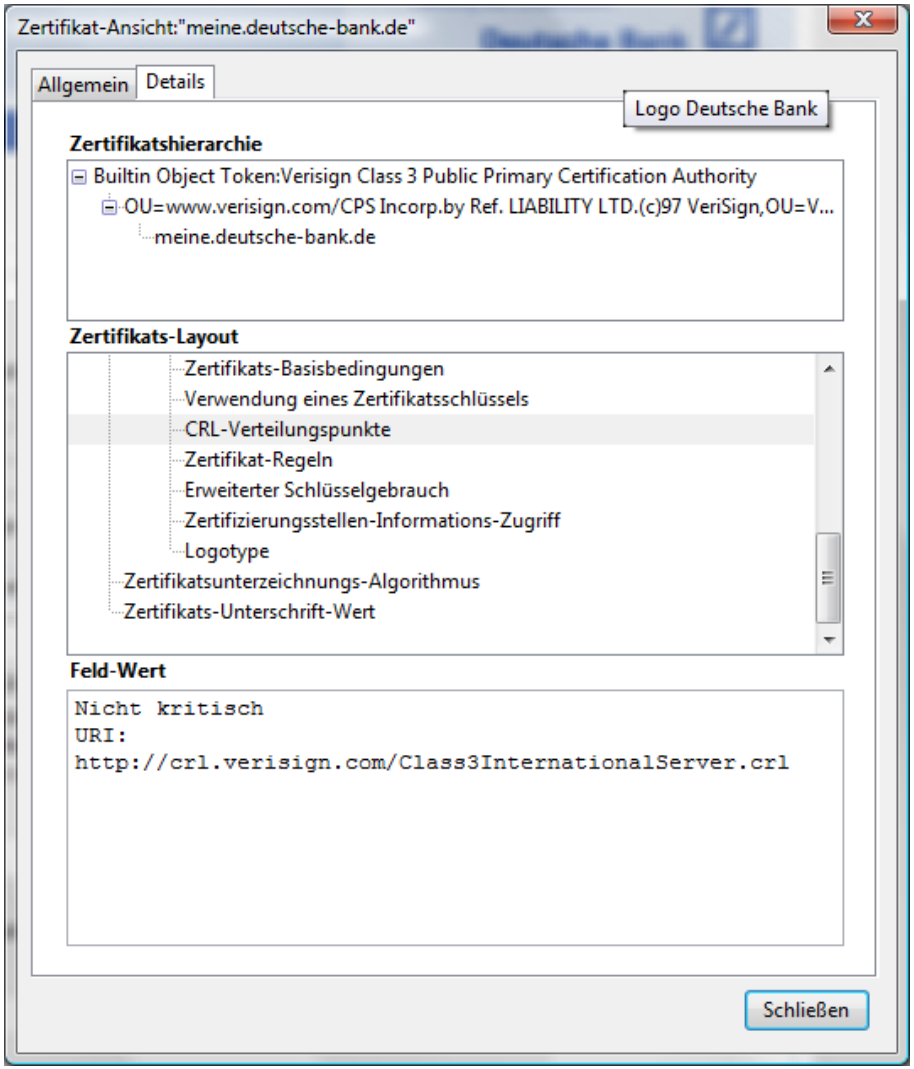
Revocation

- Was passiert, wenn der private Schlüssel abhanden kommt?
  - Versehentlich öffentlich
    - Dritte können im Namen des Zertifikatsinhabers Nachrichten signieren
  - Verlust / Private Key nicht mehr lesbar
    - Eingehende Nachrichten können nicht mehr entschlüsselt werden
    - Alle verteilten Public Keys werden nutzlos
  - Zertifikat muss für ungültig erklärt werden

Revocation

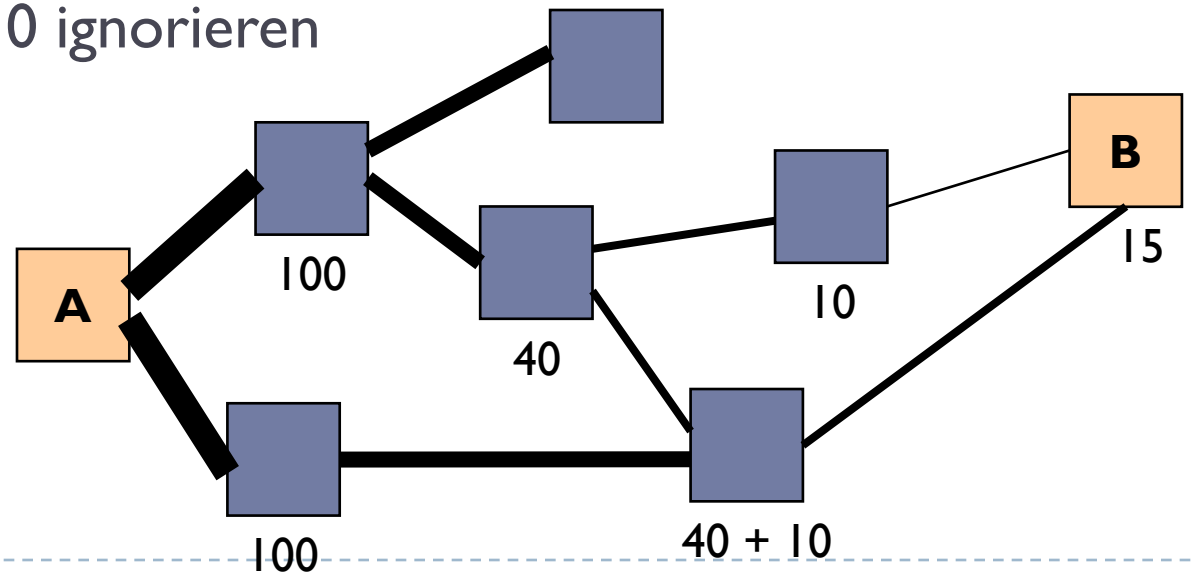
- Designated Revoker
  - Vorher im Schlüssel festgelegte Person, die den Schlüssel für ungültig erklären kann (durch Senden einer signierten Nachricht)
- Ungültigmachen durch Offenlegung des eigenen Private Keys
  - Wenn Private Key kompromittiert worden ist, ist Schlüssel sowieso unbrauchbar
  - Verhindert, dass andere den Key mutwillig „revoken“
- Revocation durch vorherige Angabe einer Schlüssellebensdauer

CRL


Zertifikat-Ansicht:"meine.deutsche-bank.de"
Allgemein
Details
Logo Deutsche Bank
Zertifikatshierarchie
Builtin Object Token:Verisign Class 3 Public Primary Certification Authority
OU=www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign,OU=V...
meine.deutsche-bank.de
Zertifikats-Layout
Zertifikats-Basisbedingungen
Verwendung eines Zertifikatsschlüssels
CRL-Verteilungspunkte
Zertifikat-Regeln
Erweiterter Schlüsselgebrauch
Zertifizierungsstellen-Informations-Zugriff
Logotype
Zertifikatsunterzeichnungs-Algorithmus
Zertifikats-Unterschrift-Wert
Feld-Wert
Nicht kritisch
URI:
http://crl.verisign.com/Class3InternationalServer.crl
Schließen

Web of Trust

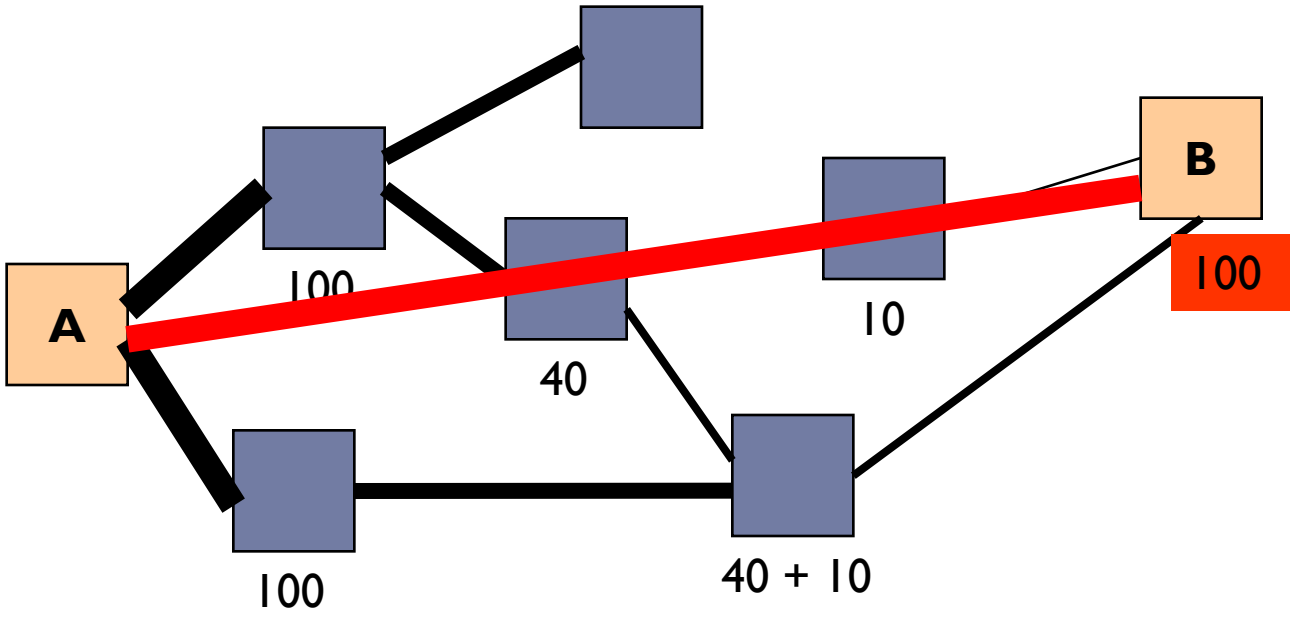
- Vertrauen ist teilweise „vererblich“
- Stärke des Vertrauens nimmt mit jeder Stufe ab
- Beispiel
  - Direkte Nachbarn 100
  - Für jeden Schritt Halbierung und 10 abziehen
  - Alles unter 10 ignorieren



Web of Trust

- Notwendige „Stärke“ (Schwellwert) des Vertrauens ist einstellbar
- Wenn Schwellwert unterschritten ist, kann das Vertrauen durch eine Authentisierung (z.B. direkte Schlüsselauthentisierung) hergestellt werden

Web of Trust



Zusammenfassung

- Zertifikate enthalten öffentliche Schlüssel. Diese Schlüssel müssen authentisiert werden.
- Zertifikate werden in einem asymmetrischen Kryptographie-System verwendet.
- Mit Hilfe von CAs oder einem Web of Trust kann die Authentizität eines Zertifikats bestätigt werden.