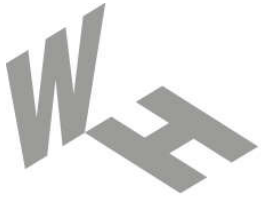


Westfälische Hochschule
Gelsenkirchen Bocholt Recklinghausen
University of Applied Sciences

Kryptographische Verfahren

Prof. Dr. (TU NN)
Norbert Pohlmann

Institut für Internet-Sicherheit – if(is)
Westfälische Hochschule, Gelsenkirchen
http://www.internet-sicherheit.de

Inhalt

- Ziele
- Einführung
- Grundlagen der Verschlüsselung
- Elementarverschlüsselung
- Symmetrische oder Private-Key Verschlüsselungsverfahren
- Asymmetrische oder Public-Key Verschlüsselungsverfahren
- One-Way-Hashfunktionen
- Zusammenfassung

Inhalt

Ziele

- Einführung
- Grundlagen der Verschlüsselung
- Elementarverschlüsselungen
- Symmetrische oder Private-Key Verschlüsselungsverfahren
- Asymmetrische oder Public-Key Verschlüsselungsverfahren
- One-Way-Hashfunktionen
- Zusammenfassung

Ziele

- Gutes Verständnis für kryptographische Verfahren und ihre Anwendungen.
- Erlangen der Kenntnisse über den Aufbau, die Prinzipien, die Architektur und die Funktionsweise von kryptographischen Verfahren.

Inhalt

- Ziele

Einführung

- Grundlagen der Verschlüsselung
- Elementarverschlüsselungen
- Symmetrische oder Private-Key Verschlüsselungsverfahren
- Asymmetrische oder Public-Key Verschlüsselungsverfahren
- One-Way-Hashfunktionen
- Zusammenfassung

Einführung
→ Motivation: Verschlüsselungstechniken

- **Wunsch nach Vertraulichkeit**
  - Transformation einer
    verständlichen Informationsdarstellung
    in eine
    nicht verständliche Informationsdarstellung

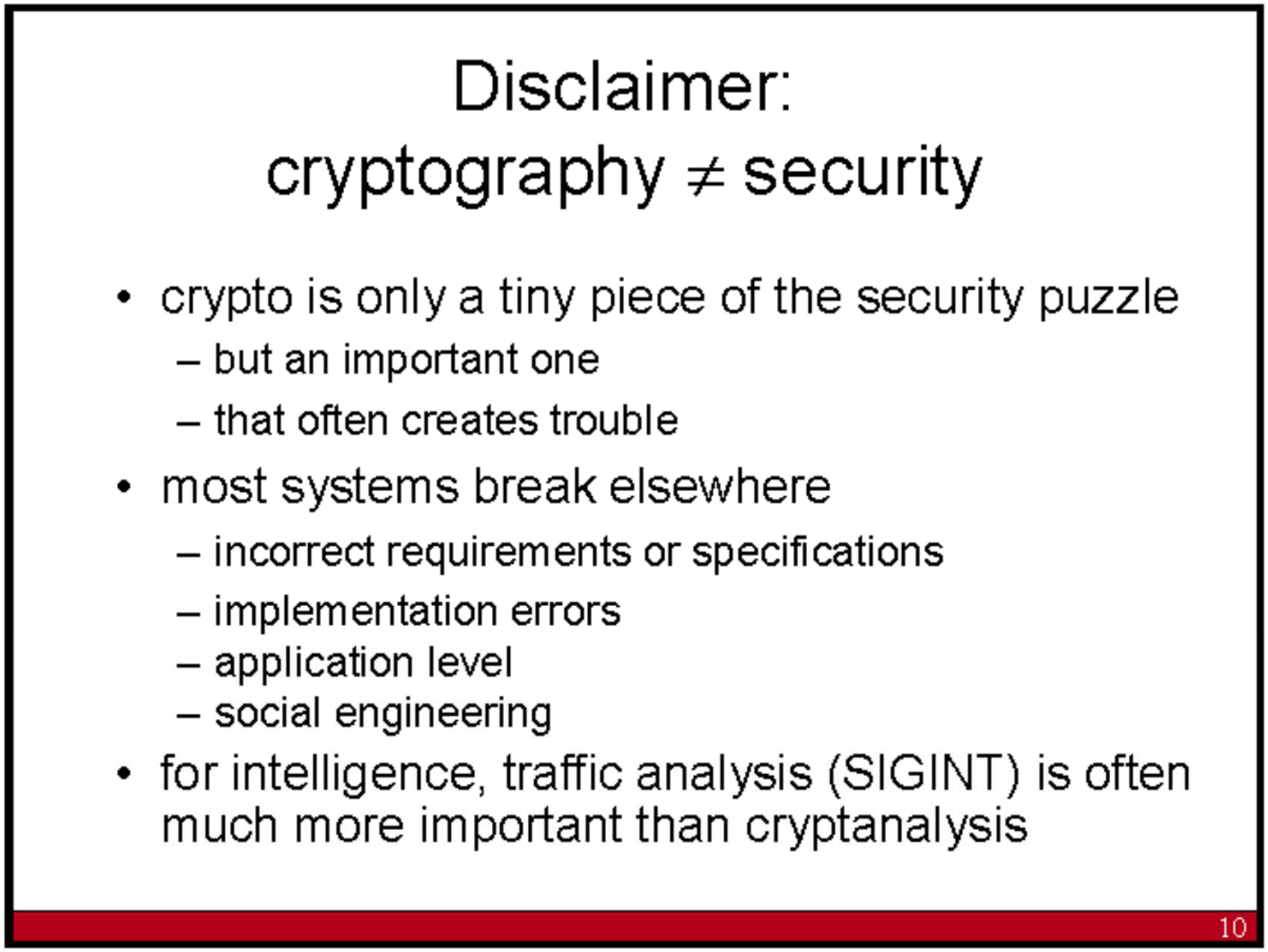
- seit 6000 Jahren gibt es Schrift,
  seit rund 3000 Jahren Verschlüsselung und
  seitdem auch den Versuch, die Verschlüsselung zu knacken !
- Romeo und Julia
- Maria Stuart und das Babington-Komplott 1586
- Eintritt der USA in den 1. Weltkrieg: Zimmermann-Telegramm 1917
- Enigma-Entschlüsselung
- Nutzung offener Netze

Einführung
→ Motivation: Kryptographie im Alltag

- Telefonkarten
- Fernbedienungen
- Mobilfunk (Handys, SIM-Karte) -> Authentikation, Verschlüsselung
- Nummerncodierung der Geldscheine
- Electronic cash (ec), HBCI, SET, ec-Karte, Bankenkarte
- Geldautomaten
- Wegfahrsperre im Auto (Autoschlüssel)
- Electronic games (Lotto, virtual casino)
- Online trading and marketplace (secure authentication)
- Multimedia services (video on demand)
- (Wireless) communication (high speed data encryption, SSL, WEP, WPA)

- **Kryptographie und Geheimsprachen sind nicht nur etwas für Agenten.**
- **Kryptographie ist eine moderne, mathematisch geprägte Wissenschaft.**

Einführung
→ Kryptographie != Sicherheit



Cryptographic Algorithms and Protocols for Network Security – Bart Preneel 2008

Inhalt

- Ziele
- Einführung
- **Grundlagen der Verschlüsselung**
- Elementarverschlüsselungen
- Symmetrische oder Private-Key Verschlüsselungsverfahren
- Asymmetrische oder Public-Key Verschlüsselungsverfahren
- One-Way-Hashfunktionen
- Zusammenfassung

Grundlagen der Verschlüsselung
→ Begrifflichkeiten

- Ziel der Verschlüsselung ist es, Daten in einer solchen Weise einer **mathematischen Transformation** zu unterwerfen, dass es einem Angreifer nicht möglich ist, die Originaldaten aus den transformierten Daten zu rekonstruieren.
- Damit die verschlüsselten Daten für ihren legalen Benutzer noch verwendbar bleiben, muss es diesem jedoch möglich sein, durch Anwendungen einer inversen Transformation aus ihnen wieder die Originaldaten zu regenerieren.
- Die Originaldaten werden mit **„Klartext“** (clear text, plain text, message) bezeichnet.
- Die transformierten Daten werden **„Schlüsseltext“**(Chiffretext, Chiffrat, Kryptogramm, cipher text) genannt.
- Die Tranformation selbst wird als **„Verschlüsselung“**, ihre Inverse als **„Entschlüsselung“** bezeichnet.

Grundlagen der Verschlüsselung
→ Definition eines kryptographischen Systems

- Beschreibbar als 6-Tupel (M; C; K_E; K_D; E; D) :
  - M = Menge der Klartext-Nachrichten m (messages, plain text)
    - z.B. M = {0, 1}, also die Menge der endlichen 0,1-Folgen
  - C = Menge der Kryptogramme c (verschlüsselte Nachrichten, cipher text)
    - z.B. C = {0, 1}
  - K_E = endliche, nicht-leere Menge der Verschlüsselungs-Schlüssel
    - z.B. $K_E = \{0, 1\}^{128}$ (128 Bit)
  - K_D = endliche, nicht-leere Menge der Entschlüsselungs-Schlüssel
    - mit: $k_d = f(k_e)$, $k_d \in K_D$, $k_e \in K_E$
  - E = Verschlüsselungsverfahren E: $M \times K_E \rightarrow C$ (umkehrbar)
  - D = Entschlüsselungsverfahren D: $C \times K_D \rightarrow M$ mit
    - für $m \in M$: $D(E(m, k_e), k_d) = m$ mit $k_e \in K_E$, $k_d \in K_D$ und $f(k_e) = k_d$

Grundlagen der Verschlüsselung
→ Überblick (1/3)

- Das generelle Ziel der Verschlüsselung kann folgendermaßen formuliert werden:
  - Die Entschlüsselung darf nur dem legalen Empfänger/Besitzer der übermittelten/gespeicherten Informationen möglich sein, nicht jedoch anderen Personen - im Extremfall nicht einmal dem Absender, der die Information selbst verschlüsselt hat.
- Dieses Ziel lässt sich offensichtlich genau dann erreichen, wenn nur der legale Empfänger/Besitzer der Information die Entschlüsselung kennt, und wenn es ohne dessen Kenntnis auch nicht möglich ist, diese aus dem Schlüsseltext zu bestimmen.
- Es wäre also auf den ersten Blick ausreichend, wenn Sender und Empfänger eine nur ihnen bekannte Transformation untereinander absprechen und die Kenntnisse darüber geheim halten.

Grundlagen der Verschlüsselung
→ Überblick (2/3)

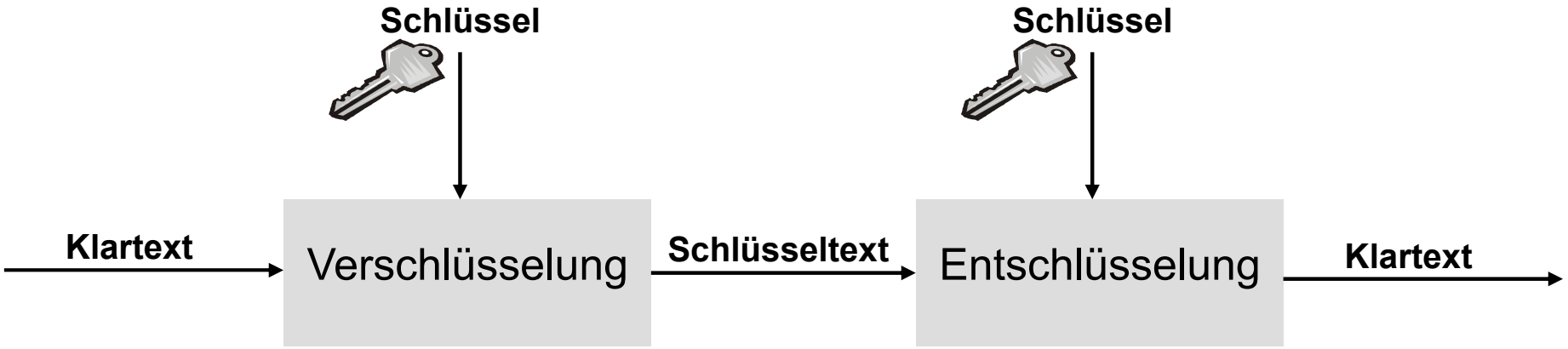
- Dieser Ansatz ist jedoch aus drei Gründen nicht verwendbar:
- **1.)** Der **Aufwand zur Definition und Realisierung** eines Verschlüsselungs-Algorithmus ist nicht zu vernachlässigen.
  Dieses Argument ist um so schwerwiegender, als dass es von Zeit zu Zeit notwendig werden kann, die Verschlüsselung zu wechseln. In diesem Fall müßte ein neuer Algorithmus eingesetzt werden.

- **2.)** Es besteht das Risiko, dass es einem Angreifer möglich ist, aus der Struktur der verschlüsselten Daten den Klartext oder die zur Verschlüsselung bzw. Entschlüsselung verwendete Transformation abzuleiten, also die Verschlüsselung zu „brechen“.
  Da es sehr aufwendig ist, den Nachweis zu führen, dass ein **bestimmtes Verschlüsselungsverfahren gegen derartige Angriffe durch „Kryptoanalysis“** sicher ist, und da ad hoc bestimmte Algorithmen mit hoher Wahrscheinlichkeit unsicher sind, ist der Einsatz eigener Verfahren für jede einzelne Kommunikation praktisch unmöglich.

- **3.)** Als letztes ist der untragbare Aufwand bei wechselnden Kommunikationspartnern zu nennen, da **für jeweils zwei Partner ein separater Verschlüsselungs-Algorithmus** zur Verfügung stehen muss.

Grundlagen der Verschlüsselung
→ Überblick (3/3)

- Als Lösung dieser Probleme bietet es sich an, zur Verschlüsselung einige wenige Algorithmen einzusetzen, deren Sicherheit erwiesen ist.
- Um die Forderung nach einer Vielzahl von Verschlüsselungsverfahren erfüllen zu können, kann man diese Algorithmen zusätzlich von einem Parameter abhängig machen, dem sogenannten **„Schlüssel“**, der den Ablauf der Transformation so stark beeinflußt, dass ohne seine Kenntnis keine Entschlüsselung möglich ist.



- Wird der Schlüssel geheim gehalten, so kann der **Verschlüsselungs-Algorithums selbst durchaus öffentlich bekannt sein**; er soll es sogar, da er nur so einer öffentlichen Diskussion preisgegeben wird.

Grundlagen der Verschlüsselung
→ Kerckhoff-Prinzip (formuliert 1883)

- in „Philosophie der modernen Kryptoanalyse“
- niederländischer Philologe Auguste Kerckhoffs von Nieuwenhof
- Die Sicherheit des **Kryptosystems darf nicht von der Geheimhaltung des Algorithmus abhängen**!
- Sie darf sich nur auf die **Geheimhaltung des Schlüssels** gründen!
- **„No security through obscurity“**
  - siehe GSM (Handy) Beispiel (A5/1-Algorithmus)!

Grundlagen der Verschlüsselung
→ Definition einiger Begriffe

- **Kryptographie**
  ist die Wissenschaft von den Methoden der Ver- und Entschlüsselung
- **Kryptoanalysis**
  ist die Wissenschaft von den Methoden der unbefugten Entschlüsselung von Daten zum Zweck der Rückführung der ursprünglichen Information.
- **Kryptosystem**
  dient zur Geheimhaltung von übertragenen oder gespeicherten Informationen gegenüber Dritten.
- **Kryptoanalyse**
  ist die Analyse eines Kryptosystems zum Zwecke der Bewertung seiner kryptographischen Stärke.
- **Kryptologie**
  ist die Wissenschaft der Verheimlichung von Informationen durch Transformation der Daten. Sie umfaßt Kryptographie und Kryptoanalysis.
- **Steganographie**
  ist eine Methode zum Verbergen der Existenz einer Information (auch digitale Wasserzeichen, Copyright-Schutz)

Grundlagen der Verschlüsselung
→ Begriffe aus der Kryptoanalysis

- Angriffe gegen Kryptosysteme können folgendermaßen unterschieden werden:
- **Cipertext-only attack**
  - Der Kryptoanalytiker kennt außer dem verwendeten Kryptoverfahren nur den Schlüsseltext.
- **Know-plaintext attack**
  - Hier stehen dem Kryptoanalytiker Klartext/Schlüsseltext Paare zur Verfügung.
  - Diese Paare können z.B. dadurch erlangt werden, dass man bestimmte Zeichenfolgen kennt, die im Klartext vorkommen (z.B. HTTP-Header).
- **Chosen-plaintext attack**
  - Der Kryptoanalytiker hat Zugang zum Verschlüsselungsgerät, nicht aber zum Schlüssel und kann somit beliebige Klartexte verschlüsseln.
  - Durch gezielte Wahl des Klartextes läßt sich unter Umständen der Schlüssel mit wesentlich niedrigerem Aufwand als bei den beiden anderen Verfahren bestimmen, so dass der Angreifer mit ausgewähltem Klartext die höchsten Anforderungen an die Sicherheit des Verschlüsselungsverfahrens stellt!

Grundlagen der Verschlüsselung
→ Strategien der Analyse eines Kryptosystems

- **Vollständige Suche**
  - Diese Methode besteht im wesentlichen im Ausprobieren aller möglichen Schlüssel.
  - Bei einem Know-Plaintext-Angriff verschlüsselt man den bekannten Klartext mit allen möglichen Schlüsseln und vergleicht den entstehenden Schlüsseltext mit dem bekannten Schlüsseltext.
- **Trial and Error Methode**
  - Bei dieser Methode wird die vollständige Suche dadurch reduziert, dass man nicht mehr den gesamten Schlüsselraum zu untersuchen braucht, sondern nur noch Teilräume, in denen der gesuchte Schlüssel vermutet wird.
  - Dieses mag z.B. der Fall sein, wenn es viel äquivalente Schlüssel gibt.
  - Schlüssel mit denselben Eigenschaften:
    - z.B. Vornamen; Spitznammen, ...
    - Darstellbare ASCII-Zeichen
      → meistens 0...9 (10), A..Z (26) und a...z (26) → 62 verschiedene ASCII-Zeichen
      → pro Byte ca. 5-Bit → z.B. von 2^{64} auf 2^{40}

Eine qualitative Schlüsselgenerierung ist sehr wichtig !

Grundlagen der Verschlüsselung
→ Strategien der Analyse eines Kryptosystems

- **Statistische Methoden**
  - Hierbei versucht der Kryptoanalytiker die statistischen Strukturen des Klartexts, das sind z.B. Buchstaben oder Worthäufigkeiten, im Schlüsseltext wiederzufinden, um dadurch an den Klartext zu gelangen.

    → Häufigkeitsanalysen von Alphabeten

- **Strukturanalyse des Kryptosystems (Short-Cut Methode)**
  - Ein solches Verfahren kann immer nur auf ein spezielles Kryptosystem zugeschnitten sein.
  - Sind z.B. alle Parameter außer dem Schlüssel bekannt, so wird man mit den gegebenen Parametern versuchen,eine Funktion aufzustellen, mit der sich der Klartext berechnen läßt.
    - **m = Kryptoanalyse (c, Design, Struktur, ...) ?**

Ein Verschlüsselungsverfahren muss mind. 5 Jahre öffentlich diskutiert werden !

Grundlagen der Verschlüsselung
→ Vollständige Suche

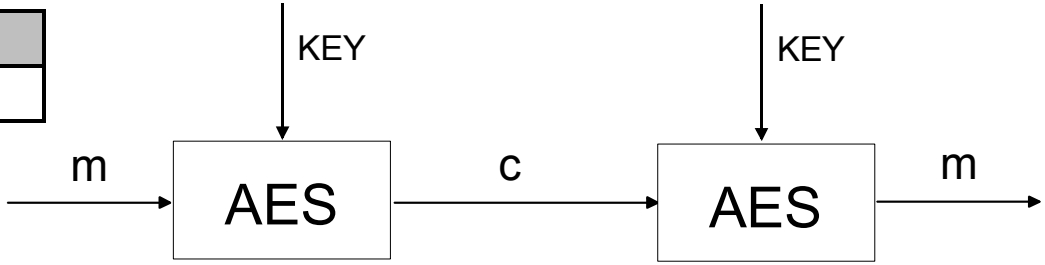
- Prinzipiell läßt sich die vollständige Schlüsselsuche (Brute-Force-Methode) gegen jedes Kryptoverfahren einsetzen.
- Sie führt aber nur dann zum Erfolg, wenn genügend Rechnerzeit und Speicherplatz zur Verfügung stehen.
- Daher läßt sich ein Kryptosystem in zwei Sicherheitskategorien einteilen.
- **Absolute Sicherheit**
  - Absolute Sicherheit liegt vor, wenn es theoretisch unmöglich ist, das System zu brechen (nur Einmal-Schlüssel mit definierten Eigenschaften).
- **Rechnerische, praktische Sicherheit**
  - Bei der rechnerischen oder praktischen Sicherheit ist es zwar theoretisch möglich, das Kryptosystem zu brechen, praktisch wird dazu jedoch so enorm viel Rechnerzeit bzw. Speicherplatz benötigt, dass dieser Weg einem jeden Kryptoanalytiker aussichtslos erscheinen muss.
  - Die meisten Informationen werden sowieso nach einer längeren Zeit wertlos.
  - Die rechnerische, praktische Sicherheit kann durch eine mathematische **Analyse der Komplexität** festgestellt werden.

Grundlagen der Verschlüsselung
→ Vollständige Suche: Beispiel (1/2)

m = Entschlüsselungsfunktion (c, KEY)

Aufwand für den „guten“ Teilnehmer

Schlüssellänge in Bits	Aufwand in s
128	6,12E-7



m = Brute-Force-Funktion (c) -> $O(2^n)$

Aufwand für den Angreifer
(Durchprobieren aller möglichen Schlüssel)

Key Länge in Bits	Anzahl der möglichen Schlüssel	Aufwand, den richtigen Schlüssel zu finden in Jahren (Annahme 1.000.000.000 Versuche in der Sekunde)
8	256	0,00000
40	1.099.511.627.776	0,00002
56	72.057.594.037.927.900	1,14
64	1.84E+19	292,47
128	3.40E+38	5.391.448.762.278.160.000.000
192	6,27E+57	9,95E+40
256	1.16E+77	1.83E+60

bedeutet praktische Sicherheit

Grundlagen der Verschlüsselung
→ Vollständige Suche: Beispiel (2/2)

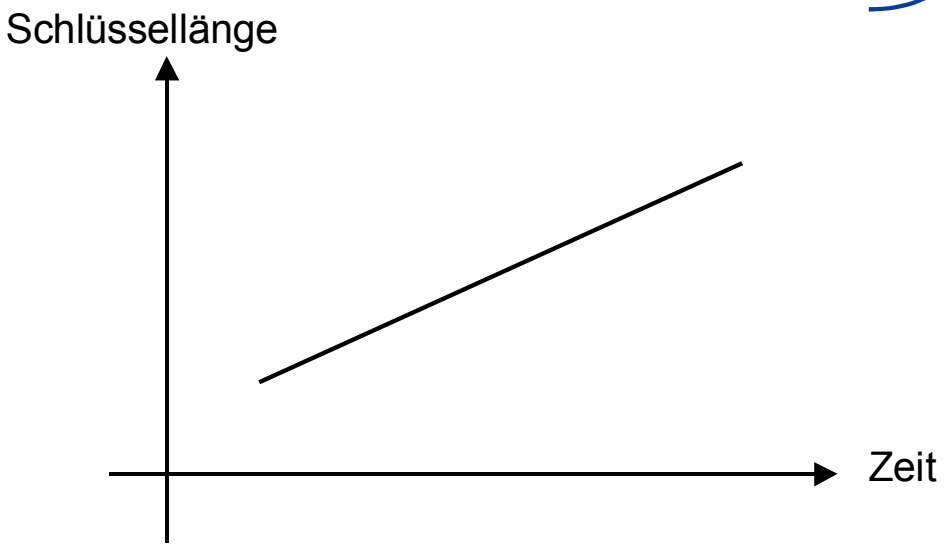
- Neuere Erkenntnisse zeigen allerdings, dass bei dem Beispiel AES eine so genannte „realted-key weakness“ bei AES-256 und AES-192 vorliegt.
- Das reduziert den Schlüsselraum auf 2^{119} bzw. 2^{176}
- Gilt immer noch als ausreichend.
- Solche Fehler werden im laufe der Zeit mehr, was die Algorithmen insgesamt schwächt, ohne Rechenzeit in Betracht zu ziehen

Grundlagen der Verschlüsselung
→ Geschwindigkeit von Computern

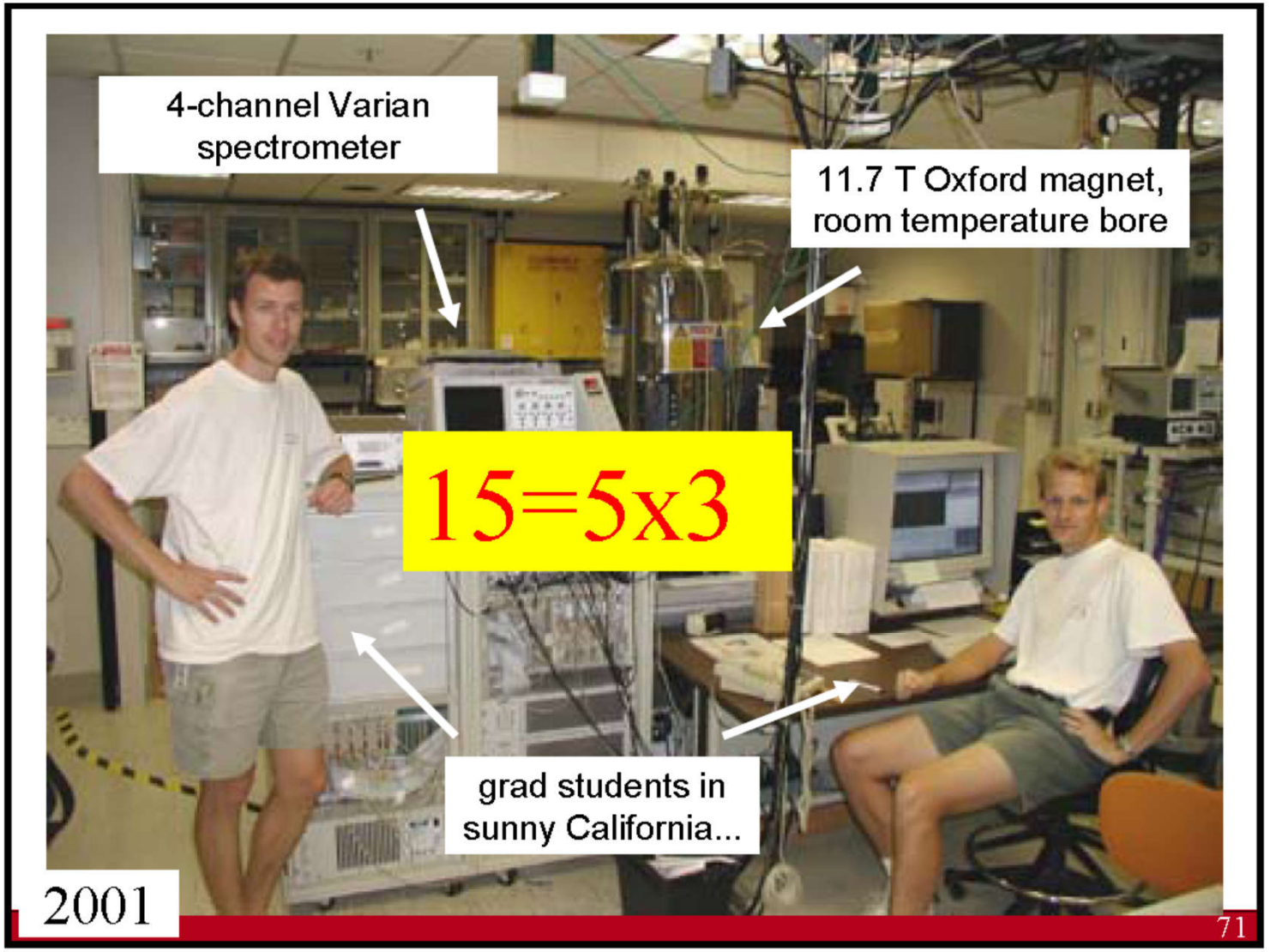
- Der Zeitfaktor und die Innovationen (z.B. Quantenrechner) müssen berücksichtigt werden

- **Praktische Sicherheit**
  - vor 20 Jahren eine Schlüssellänge von 64 Bit (*DES*)
  - heute 128 Bit (*Triple DES, AES*)
  - für die nächsten 20 Jahre 256 Bit (*AES*)

Alle 10 bis 15 Jahre ist ein Wechsel der Algorithmen notwendig!

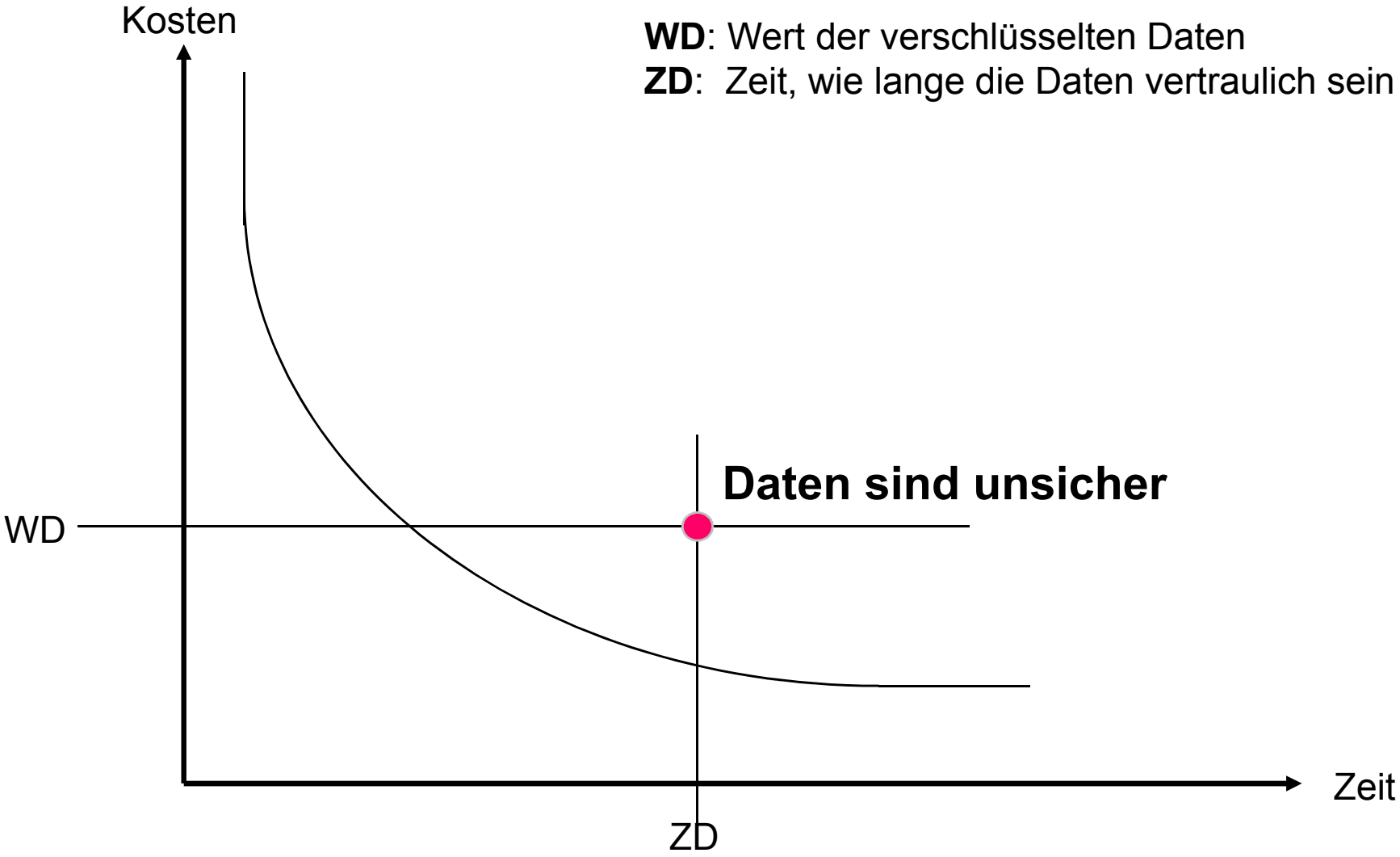


Quantencomputer



Cryptographic Algorithms and Protocols for Network Security – Bart Preneel 2008

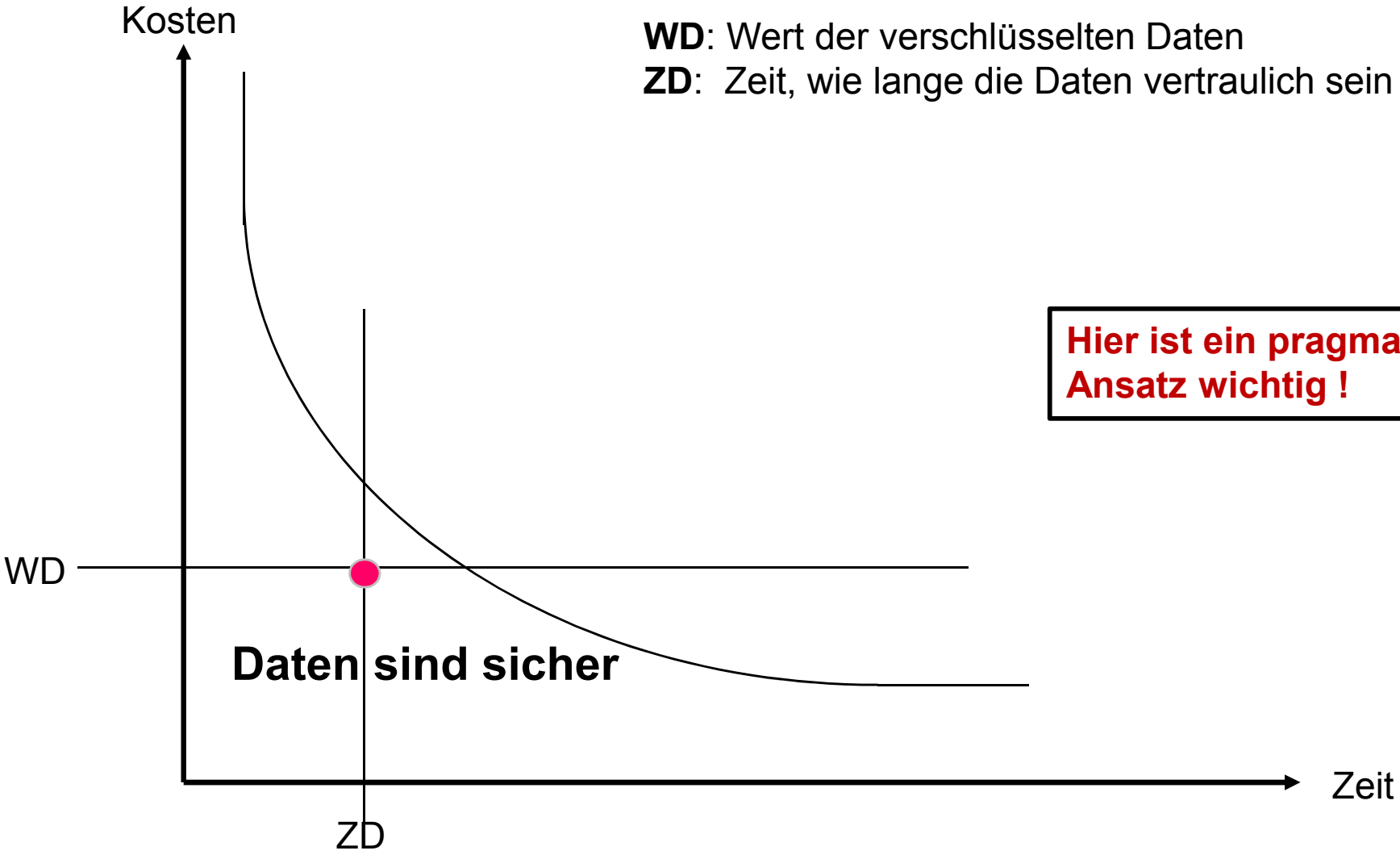
Grundlagen der Verschlüsselung
→ Zeit/Kosten, die Daten zu entschlüsseln (1/2)



WD: Wert der verschlüsselten Daten
ZD: Zeit, wie lange die Daten vertraulich sein müssen

Source: RSA Labs, NetworkWorld, June 2000

Grundlagen der Verschlüsselung
→ Zeit/Kosten, die Daten zu entschlüsseln (2/2)



WD: Wert der verschlüsselten Daten
ZD: Zeit, wie lange die Daten vertraulich sein müssen

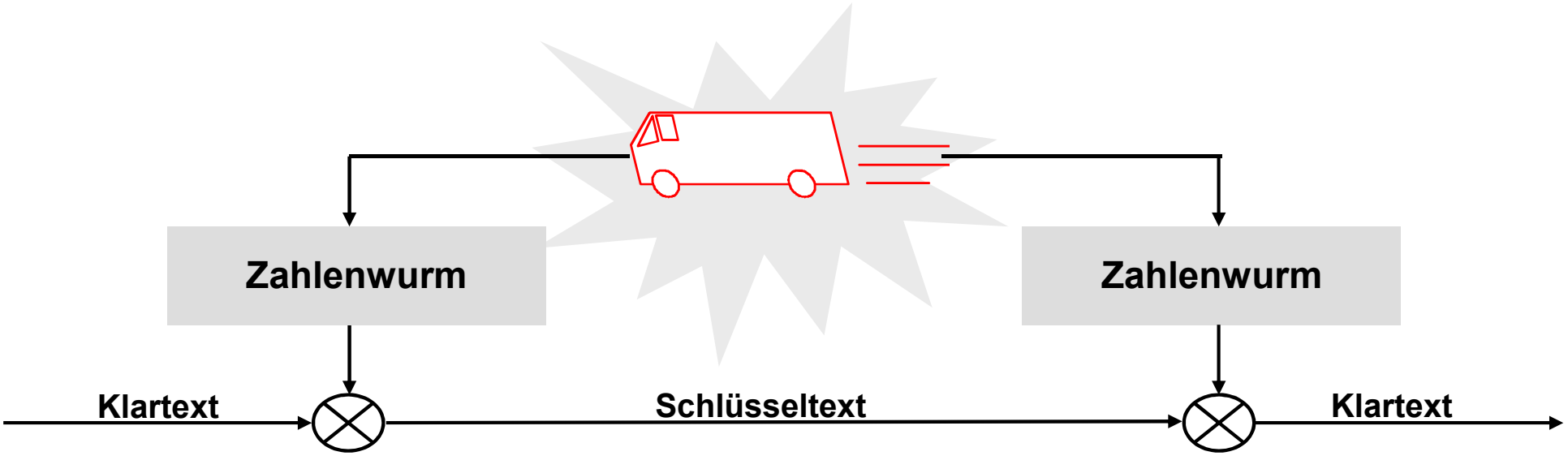
Hier ist ein pragmatischer Ansatz wichtig !

Source: RSA Labs, NetworkWorld, June 2000

Inhalt

- Ziele
- Einführung
- Grundlagen der Verschlüsselung
- **Elementarverschlüsselungen**
- Symmetrische oder Private-Key Verschlüsselungsverfahren
- Asymmetrische oder Public-Key Verschlüsselungsverfahren
- One-Way-Hashfunktionen
- Zusammenfassung

Grundlagen: Elementarverschlüsselungen
→ Der Einmal-Schlüssel



- Dieses Verfahren wird auch individuelle Wurmverschlüsselung, Zahlenwurm oder One-Time-Pad genannt.
- Der Einmal-Schlüssel zählt zu den „absolut sicheren“ Verschlüsselungsverfahren.
- Das Verfahren benötigt für jede Nachricht einen Zahlenwurm d.h. einen Schlüssel, der mindestens die Länge des zu übermittelnden Klartext haben muss.

Grundlagen: Elementarverschlüsselungen
→ Der Einmal-Schlüssel

- Der Zahlenwurm/Schlüssel muss für jede Nachricht neu durch **Zufallskriterien** erzeugt werden und **sicher** zwischen den Kommunkationspartnern verteilt werden.
- Der Schlüssel und die Nachricht werden bitweise modulo 2 addiert, d.h. XOR verknüpft.
- Da jede Nachricht mit einem gleichlangen Schlüssel verknüpft wird, geht im Schlüsseltext jede Struktur verloren, sodass sich für die Kryptoanalysis keinerlei Ansatzpunkte bieten.
- **Wichtig ist die Qualität der Zufallszahlen!**
- Obwohl dieses Verfahren für den „heißen Draht“ zwischen Washington und Moskau genutzt wurde (wird?), ist dieses Verfahren für den kommerziellen Einsatz nicht geeignet, da anstelle des absolut geheimen Schlüssels ebensogut die zu übertragende Nachricht selbst auf dem sicheren Weg übermittelt werden könnte.

Grundlagen: Elementarverschlüsselungen
→ Monoalphabetische Substitution: Verfahren

- Eine recht einfache Methode, einen Klartext zu verschlüsseln, besteht darin, nach einem bestimmten Schema jedes Zeichen des Klartextes durch ein anderes, dem Klartext fest zugeordnetes Zeichen zu ersetzen d.h. zu substituieren.

- **Verschlüsselungsvorschrift:**

  (1) A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  (2) G W X V L O A K U B C N D R M F H Y P Q T Z E I J S

- Beispiel:
  - Schlüssel: Verschlüsselungsvorschrift
  - **Klartext: K R Y P T O L O G I E**
  - **Schlüsseltext: C Y J F Q M N M A U L**

- Es können auch verschiedenartige Alphabete verwendet werden (z.B. lateinische Buchstaben und 26 Buchstaben des chinesischen Alphabetes).

Grundlagen: Elementarverschlüsselungen
→ Monoalphabetische Substitution: Kryptoanalyse

- Verfahren dieser Art sind durch Häufigkeitsanalysen leicht zu brechen.
- In jeder natürlichen Sprache kommen die Buchstaben nicht gleich häufig vor, vielmehr hat jeder Buchstabe charakteristische Häufigkeiten.

Buchstabe	Häufigkeit (in %)	Buchstabe	Häufigkeit (in %)
A	6,51	N	9,78
B	1,89	O	2,51
C	3,06	P	0,79
D	5,08	Q	0,02
E	17,40	R	7,00
F	1,66	S	7,27
G	3,01	T	6,15
H	4,76	U	4,35
I	7,55	V	0,67
J	0,27	W	1,89
K	1,21	X	0,03
L	3,44	Y	0,04
M	2,53	Z	1,13

Häufigkeit der Buchstaben der deutschen Sprache

- Was passiert nun, wenn ein deutscher Klartext mit der monoalpabetischen Substitution verschlüsselt wird?
  - **Die Häufigkeitsverteilung der Buchstaben bleibt erhalten!**

Grundlagen: Elementarverschlüsselungen
→ Homophone Substitution: Verfahren

- Die homophone Substitution ist eine Verbesserung der monoalphabetischen Substitution.
- Die Verbesserung wird durch einer Verschleierung der Häufigkeit erreicht.
- Bei diesem Verfahren wird die Verschlüsselungsvorschrift so gestaltet, dass alle Schlüsseltextzeichen mit der gleichen Wahrscheinlichkeit auftreten.
- Dazu wird jedem Buchstaben eine Menge von Zeichen zugeordnet, und zwar so, dass die Anzahl der Schlüsseltextzeichen, die zu einem Buchstaben gehören, seiner Häufigkeit entsprechen.
- Bei der Verschlüsselung wird ein Klartextbuchstabe zufällig einem dazugehörigen Schlüsseltextzeichen zugeordnet.
- Da die Zeichen zufällig gewählt werden, kommt jedes Zeichen gleich häufig vor.

Grundlagen: Elementarverschlüsselungen
→ Homophone Substitution: Verschlüsselungsvorschrift

Klartext	Schlüsseltext
A	(10,21,52,59,71)
B	(20,34)
C	(28,06,80)
D	(19,58,70,81,87)
E	(09,18,29,33,38,40,42,54,55,60,66,75,85,86,92,93,99)
F	(00,41)
G	(08,12,97)
H	(01,07,24)
I	(14,39,50,65,76,88,94)
J	(57)
K	(23)
L	(02,05,82)
M	(27,11,49)
N	(30,35,43,62,67,68,72,77,79)
O	(26,53)
P	(31)
Q	(25)
R	(17,36,51,69,74,78,83)
S	(15,16,45,56,61,73,96)
T	(13,32,90,91,95,98)
U	(03,04,47)
V	(37)
W	(22)
X	(44)
Y	(48)
Z	(64)

- Schlüssel: Verschlüsselungsvorschrift
- **Klartext:** **K R Y P T O L O G I E**
- **Schlüsseltext:** **23 69 48 31 90 26 05 53 08 94 33**

Grundlagen: Elementarverschlüsselungen
→ Homophone Substitution: Kryptoanalyse (1/2)

- Die Analyse basiert auf der Beobachtung, dass zwar die Häufigkeit der Schlüsseltextzeichen gleich ist, dass aber aus der Betrachtung von Paaren von Schlüsseltextzeichen sehr wohl Informationen gewonnen werden kann.

Buchstabenpaar	Häufigkeit (in %)
en	3,88
er	3,75
ch	2,75
te	2,26
de	2,00
nd	1,99
ei	1,88
ie	1,79
in	1,67
es	1,52

Häufigkeit der Buchstabenpaare

Grundlagen: Elementarverschlüsselungen
→ Homophone Substitution: Kryptoanalyse (2/2)

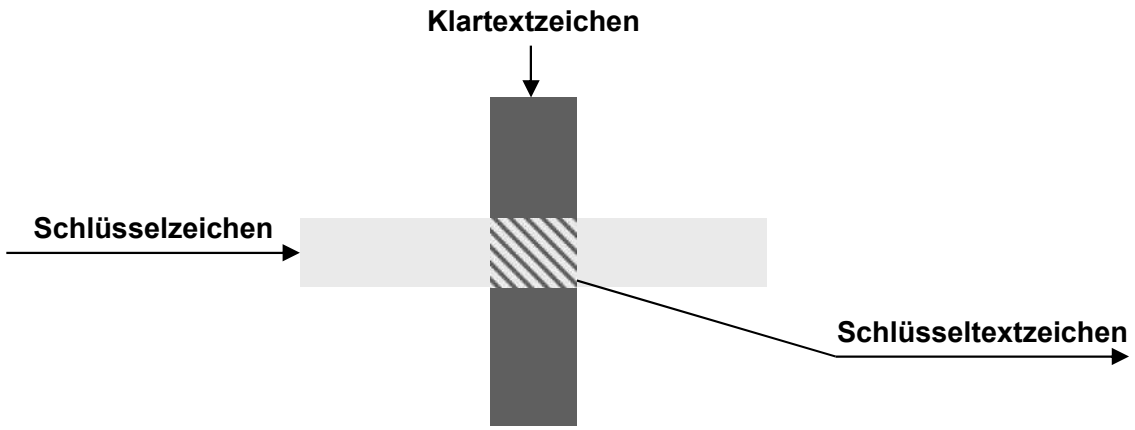
- Betrachtet man ein Schlüsseltextäquivalent des Buchstaben C, also etwa 28, so wird man feststellen, dass nur bestimmte Schlüsseltextzeichen als unmittelbare Nachfolger von 28 in Frage kommen.
- Das sind für C gleich 01, 07, 24, 23 also die Schlüsseltextäquivalente der Buchstaben H und K.
- Damit weiß man bereits, welche Zeichen dem Buchstaben H und K entsprechen.

Klartext	Schlüsseltext
A	(10,21,52,59,71)
B	(20,34)
C	(28,06,80)
D	(19,58,70,81,87)
E	(09,18,29,33,38,40,42,54,55,60,66,75,85,86,92,93,99)
F	(00,41)
G	(08,12,97)
H	(01,07,24)
I	(14,39,50,65,76,88,94)
J	(57)
K	(23)
L	(02,05,82)
M	(27,11,49)
N	(30,35,43,62,67,68,72,77,79)
O	(26,53)
P	(31)
Q	(25)
R	(17,36,51,69,74,78,83)
S	(15,16,45,56,61,73,96)
T	(13,32,90,91,95,98)
U	(03,04,47)
V	(37)
W	(22)
X	(44)
Y	(48)
Z	(64)

- Diese Andeutung ist natürlich noch längst keine Kryptoanalyse, sie soll nur zeigen, dass auf den ersten Blick „praktisch unknackbare“ Verfahren doch angreifbar sind.

Grundlagen: Elementarverschlüsselungen
→ Polyalphabetische Substitution: Verfahren

- Substitutionsverfahren, die die Häufigkeitsanalyse stärker verschleiern, sind z.B. die „polyalphabetischen Substitutionsverfahren“.
- Das bekannteste Verfahren ist die **Vigenére-Verschlüsselung**
- Diese Verfahren arbeiten mit einem Schlüssel, der aus einer Zeichenfolge besteht, von der jedes Zeichen eine bestimmte Zeile der Tabelle auswählt, und jedes Klartextzeichen wählt eine bestimmte Spalte der Tabelle aus.



- Der Kreuzungspunkt der Zeile und Spalte enthält dann das zugehörige Schlüsseltext-Zeichen.
- Der Schlüssel wird wiederholt, wenn er kürzer als der Klartext ist.
- Die Entschlüsselung erfolgt in der umgekehrten Weise.

Grundlagen: Elementarverschlüsselungen
→ Polyalphabetische Substitution: Beispiel des Verfahrens

- **Verschlüsselungsvorschrift:**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
3	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
4	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
5	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
.													.													
.													.													
.													.													

- Beispiel:
  - **Klartext:** **K R Y P T O L O G I E**
  - Schlüssel: 5 3 2
  - **Schlüsseltext:** **O T Z T V P P Q H M G**

Grundlagen: Elementarverschlüsselungen
→ Polyalphabetische Substitution: Kryptoanalyse

- Obwohl es aufwendiger statistischer Analyse bedarf, können auch polyalphabetische Verfahren gebrochen werden.
- Ein genügend langer Schlüsseltext weist viele statistisch erfassbare Regelmäßigkeiten auf, die es einem ermöglichen, den Schlüssel zu erhalten.
- Methoden, die die Länge des benutzten Schlüssels bestimmen.
  - Abstand der beiden Klartextbuchstaben ist ein Vielfaches der Schlüssellänge.
    - Gleicher Klartext = gleicher Schlüsseltext
  - Wenn der Klartext genauso lang wie der Schlüssel ist, arbeitet das Verfahren wie eine monoalphabetische Substitution.

Grundlagen: Elementarverschlüsselungen
→ Transpositions-Verfahren: Zick-Zack-Verfahren

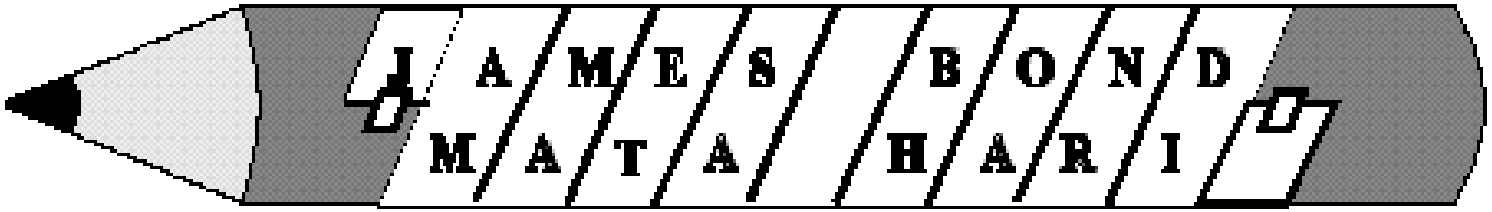
- Transpositionsverfahren sind Verschlüsselungsverfahren, bei denen die einzelnen Zeichen des Klartextes nach einer bestimmten Regel permutiert d.h. vertauscht werden.
- **Verschlüsselungsvorschrift des Zick-Zack-Verfahrens:**
  Der Klartext wird in einer Zick-Zack-Kurve z.B. mit einer Tiefe von fünf (Schlüssel-Wert) aufgeschrieben, und anschließend wird der Schlüsseltext zeilenweise von oben nach unten ausgelesen.
- Schlüssel: Tiefe der Zick-Zack-Kurve (hier 5)
- **Klartext:** **LABOR-FÜR-VERTEILTE-SYSTEME-UND ...**
-

```
L       R       L       E
 A     Ü  -     I  T     T  M     ...
  B   F    V   E    E   S    E   D
   O  -     E  T     -  Y     -  N
    R       R       S       U
```

- **Schlüsseltext:** **LRLEAÜ-ITTMBFVEESEDO-ET-Y-NRRSU**

Grundlagen: Elementarverschlüsselungen
→ Transpositions-Verfahren: Sparta (500 v. Chr.)

- Zwei Zylinder (Holzstäbe) mit genau demselben Radius
  - Sender wickelte ein schmales Band aus Pergament spiralförmig um seinen Zylinder und schrieb dann der Länge nach seine Nachricht auf das Band.
  - Die Nachricht auf dem abgewickelten Band konnte nur von einer Person gelesen werden, die einen Zylinder genau desselben Umfangs hatte.



Inhalt

- Ziele
- Einführung
- Grundlagen der Verschlüsselung
- Elementarverschlüsselungen
- **Symmetrische oder Private-Key Verschlüsselungsverfahren**
- Asymmetrische oder Public-Key Verschlüsselungsverfahren
- One-Way-Hashfunktionen
- Zusammenfassung

Symmetrische Verschlüsselungsverfahren
→ Produktverschlüsselung

- Verknüpft man Elementarverschlüsselungen mit verschiedenen kryptographischen Eigenschaften, so spricht man von einer Produktverschlüsselung.
- Ziel der **Produktverschlüsselung** ist es, **kryptographisch stärker** d.h. schwerer zu brechen zu sein, als jede ihrer Einzelverschlüsselung.
- Eine der häufigsten Produktverschlüsselungen ist die **iterative Verknüpfung von nichtlinearen Substitutionen und Permutationen**.
- Vertreter der Produktverschlüsselungen sind z.B.:
  - **Data Encryption Standard (DES)**
  - **Advanced Encryption Standard (AES)**
  - **International Data Encryption Algorithmus (IDEA)**

DES-Algorithmen
→ Fakten

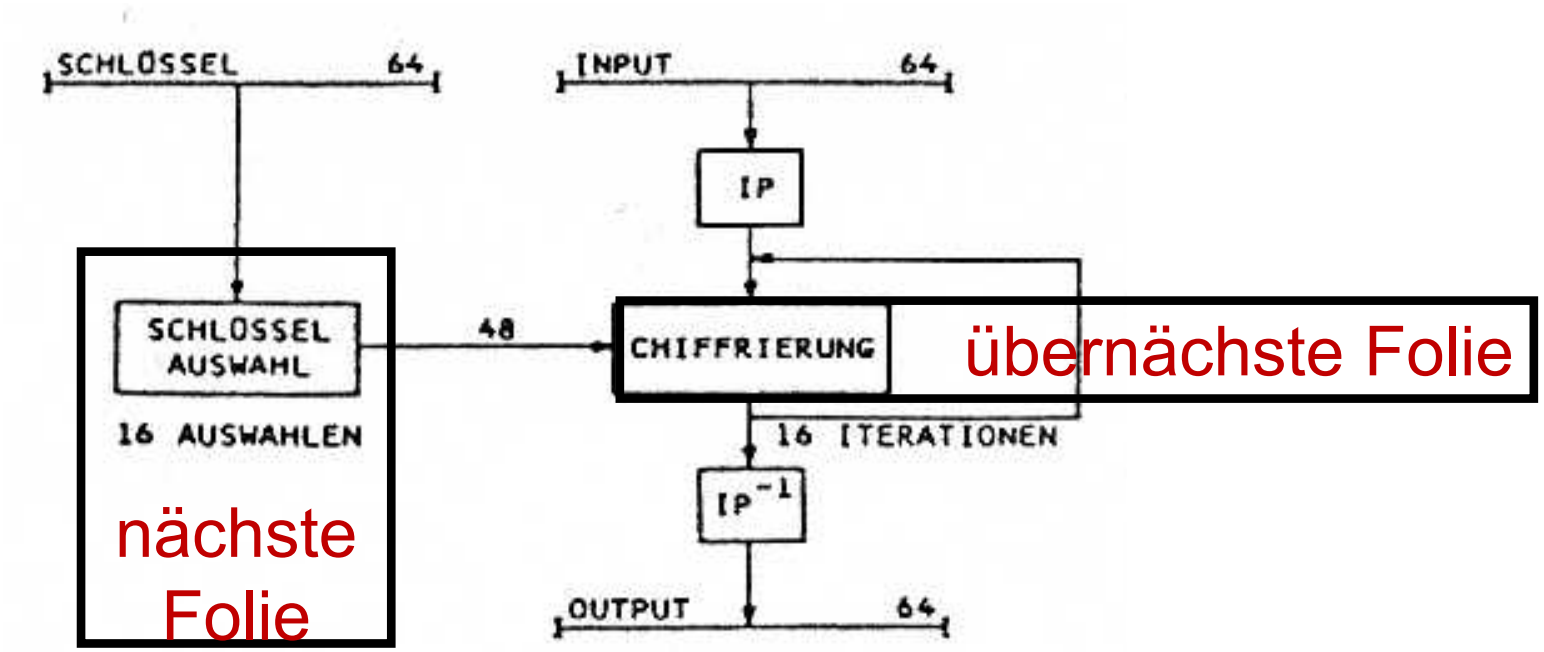
- DES = Data Encryption Standard (1976)
- Weltweiter Standard für 25 Jahren
  - ist von IBM entwickelt worden (undurchsichtiger Vorgang)
  - Lizenz- und rechtefreie Verwendung
- Block Cipher
  - Blocklänge 64 bit (8 Byte)
- Schlüssellänge 56 Bit
  - 8 Bit Paritäts Überprüfung (8 Byte)
- Ideal zur Implementierung in Hardware
- Sehr viel Strategie für SW (CPU, RAM, Schlüssel-Vorbereitung, ...)
- **Sicherheit:**
  - **Schlüssellänge wird als nicht mehr ausreichend eingestuft**
  - Eine Abhilfe für ein paar Jahre war die Verwendung des Triple DES

Data Encryption Standard (DES)
→ Übersicht des Verfahrens (1/2)

- Der Data Encryption Standard ist der Verschlüsselungsalgorithmus, der am weitesten verbreitet war, aber vom AES abgelöst wurde.
- Der Data Encryption Standard stellt eine Block-Produkt-Verschlüsselung aus nichtlinearer Substitution und Permutation dar, die schlüsselgesteuert in einer Iterationsschleife 16 mal durchlaufen werden.
- Dazu wird der Klartext in 64-Blöcke zerlegt.
- Der Schlüssel besteht aus 64-Bit, wovon 56-Bit beliebig wählbar sind, während die restlichen 8-Bit die Funktion von Paritätsbits haben.
- Jeder Input wird einer Eingangspermutation unterzogen.
- Anschließend durchläuft der Block 16 schlüsselabhängige aber funktional identische Iterationen.

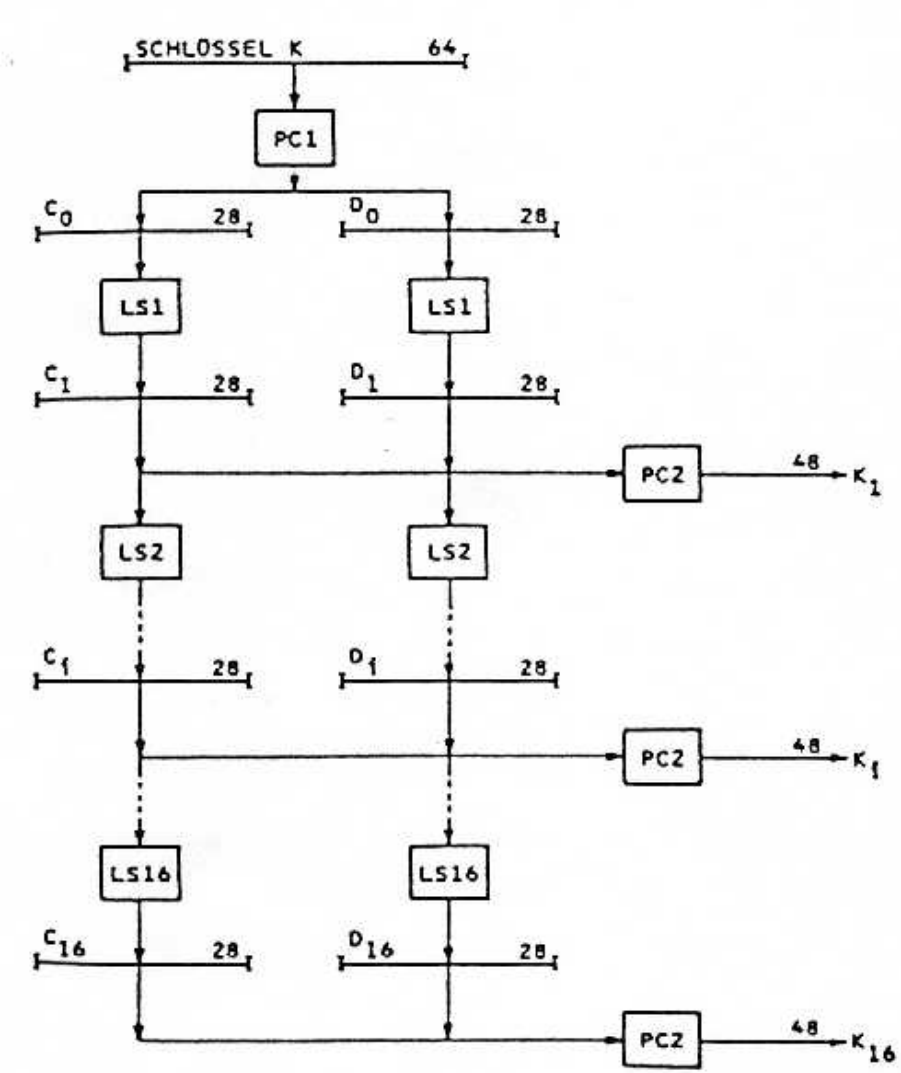
Data Encryption Standard (DES)
→ Übersicht des Verfahrens (2/2)

- Jede der Iteration benutzt einen unterschiedlichen 48-Bit Teilschlüssel.
- Auf dem Ausgang der letzten Iteration wird die inverse Eingangspermutation durchgeführt, und man erhält den Output, d.h. den Schlüsseltext-Block.

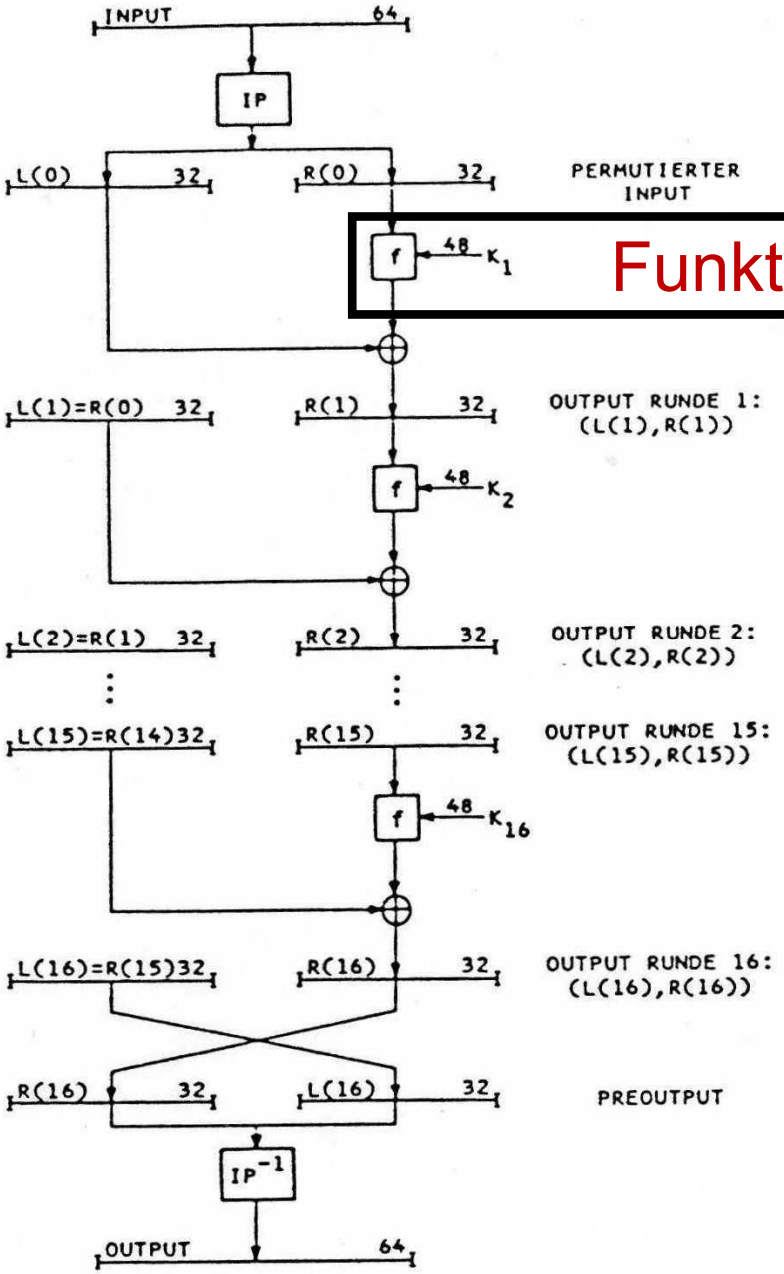


Data Encryption Standard (DES)
→ Schlüsselauswahlfunktion

- Auf den 56-Bit großen Schlüssel wird zuerst eine Permutation angewendet.
- Der resultierende 56-Bit Vektor wird in eine linke und rechte Hälfte aufgeteilt.
- Die beiden Hälften werden dann jeweils getrennt um eine oder zwei Bitpositionen, zirkulär links geschiftet.
- Ob eine oder zwei Bitpositionen zirkulär links geschiftet werden, hängt vom jeweiligen Iterationsschritt ab.
- Anschließend werden in jedem Iterationsschritt 48-Bit aus dem 56-Bit-Vektor ausgewählt und einer Permutation unterworfen.
- Das Ergebnis ist dann der jeweilige Teilschlüssel eines Iterationsschrittes.



Data Encryption Standard (DES)
→ Verschlüsselungsalgorithmus (Chiffrierung)

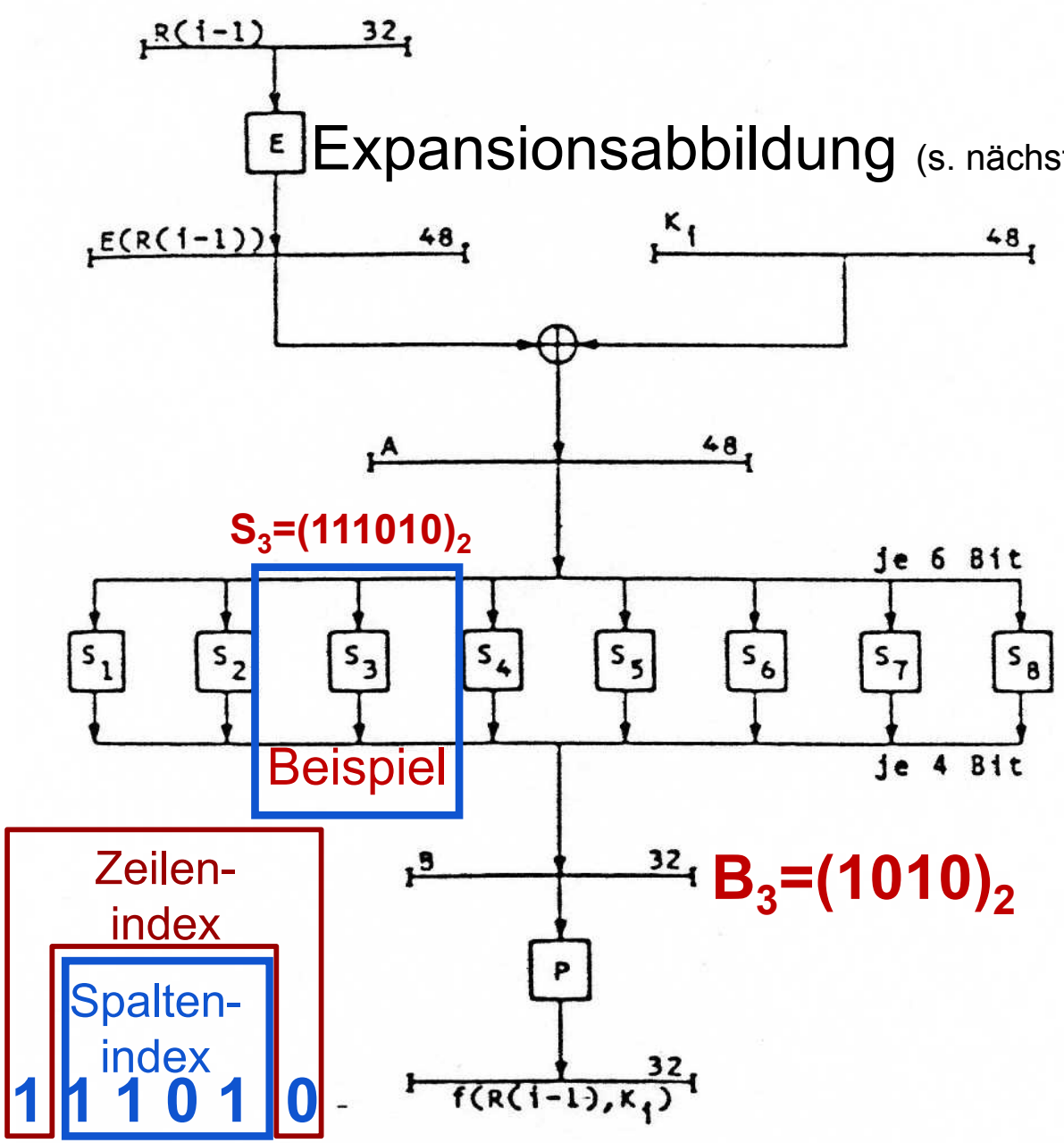


Funktion ab nächster Folie

Data Encryption Standard (DES)
→ Die Funktion f (1/3)

- Der 32-Bit Input-Block der Funktion wird mit der Expansionsabbildung E auf einen 48-Bit-Block erweitert.
- Dies geschieht durch die Duplizierung ausgewählter Bit vom Input-Block mit gleichzeitiger Permutation.
- Ki ist der durch die Schlüsselauswahlfunktion bestimmte 48-Bit Teilschlüssel.
- Der Teilschlüssel ki und der erweiterte Input-Block werden bitweise modulo 2 addiert, d.h. XOR verknüpft.
- Das Ergebnis wird in acht 6-Bit Blöcke aufgeteilt und in acht verschiedenen Substitution-Boxen einer nichtlinearen Substitution unterworfen.
- Die **nichtlineare Substitution stellt den kryptologisch wichtigsten Teil des DES** dar.
- Das Ergebnis der acht getrennten Substitutionen ist ein 32-Bit Vektor, auf den eine Permutation angewendet wird.

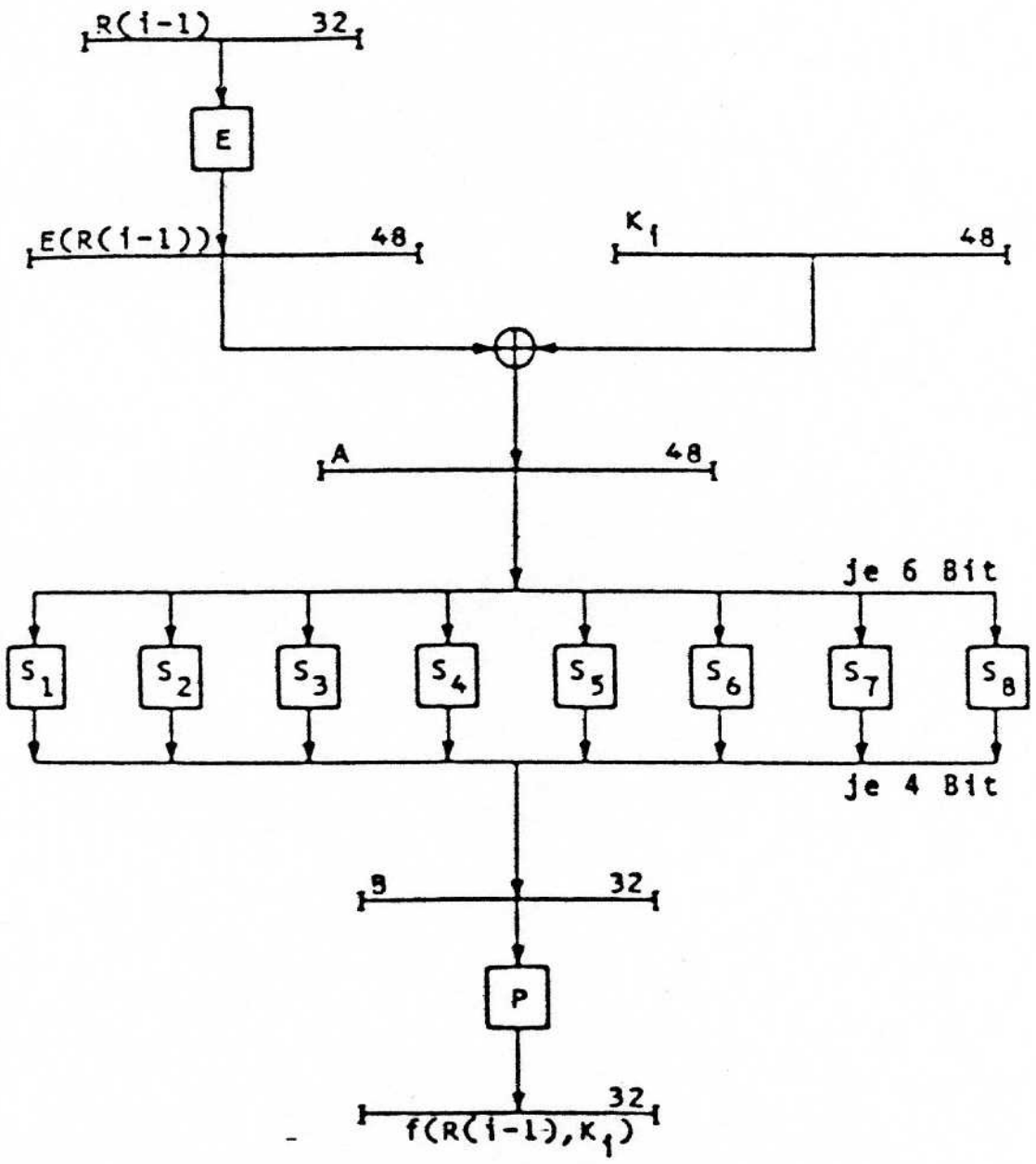
Data Encryption Standard (DES)
→ Die Funktion f (2/3)



Tafel der S-Boxen:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	S_6
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S_7
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S_8
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Data Encryption Standard (DES)
→ Die Funktion f (3/3)



Expansionsabbildung

Erweiterungspermutation

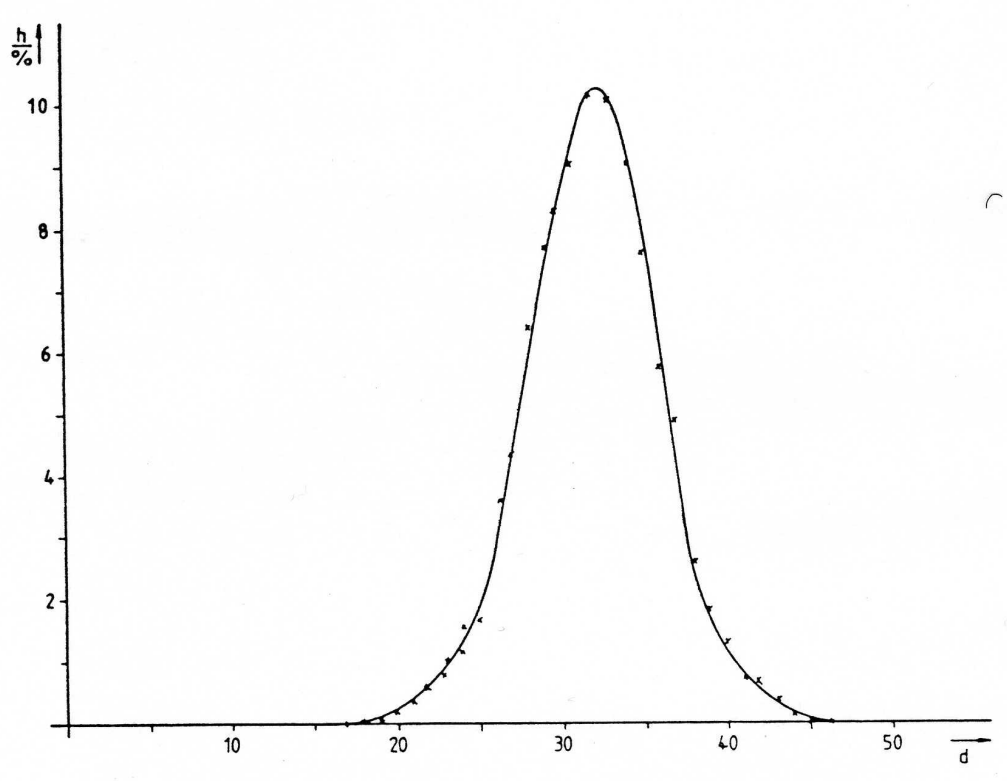
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Permutation P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Data Encryption Standard (DES)
→ Fehlerfortpflanzung

- Jede S-Box des DES-Verfahrens besitzt eine 2:1 Fehlerfortpflanzung, d.h. durch die Veränderung eines S-Box Inputbits verändern sich zwei S-Box Outputbits.
  - Wird also im Klartext ein Bit verändert, so ändern sich im Schlüsseltext etwa 32-Bit also 50 %
  - Das gleiche Verhalten tritt auch beim Schlüssel auf



- Kleine Änderungen im Klartext oder im Schlüssel führen also zu großen Änderungen im Schlüsseltext und üben somit einen lawinenartigen Vorgang aus.
- Dieses Verhalten müssen alle kryptographischen Verfahren haben!

Data Encryption Standard (DES)
→ Kryptoanalyse

- **Vollständige Suche**
  - 2^{56} ist heute zu klein!
- **Schwache Schlüssel**
  - Der DES besitzt vier schwache Schlüssel
  - DES (DES (m, k), k) = m
  - **Diese müssen bei der Schlüsselgenerierung berücksichtigt werden**

```
0000 0000 0000 0000
fefe fefe fefe fefe
lele lele 0e0e 0e0e
e0e0 e0e0 f0f0 f0f0
```

- **Statistische Methoden**
  - Der DES besitzt sehr gute statistische Eigenschaften.
  - Es sind keine Abhängigkeiten zwischen Klartext und Schlüsseltext nachgewiesen worden.

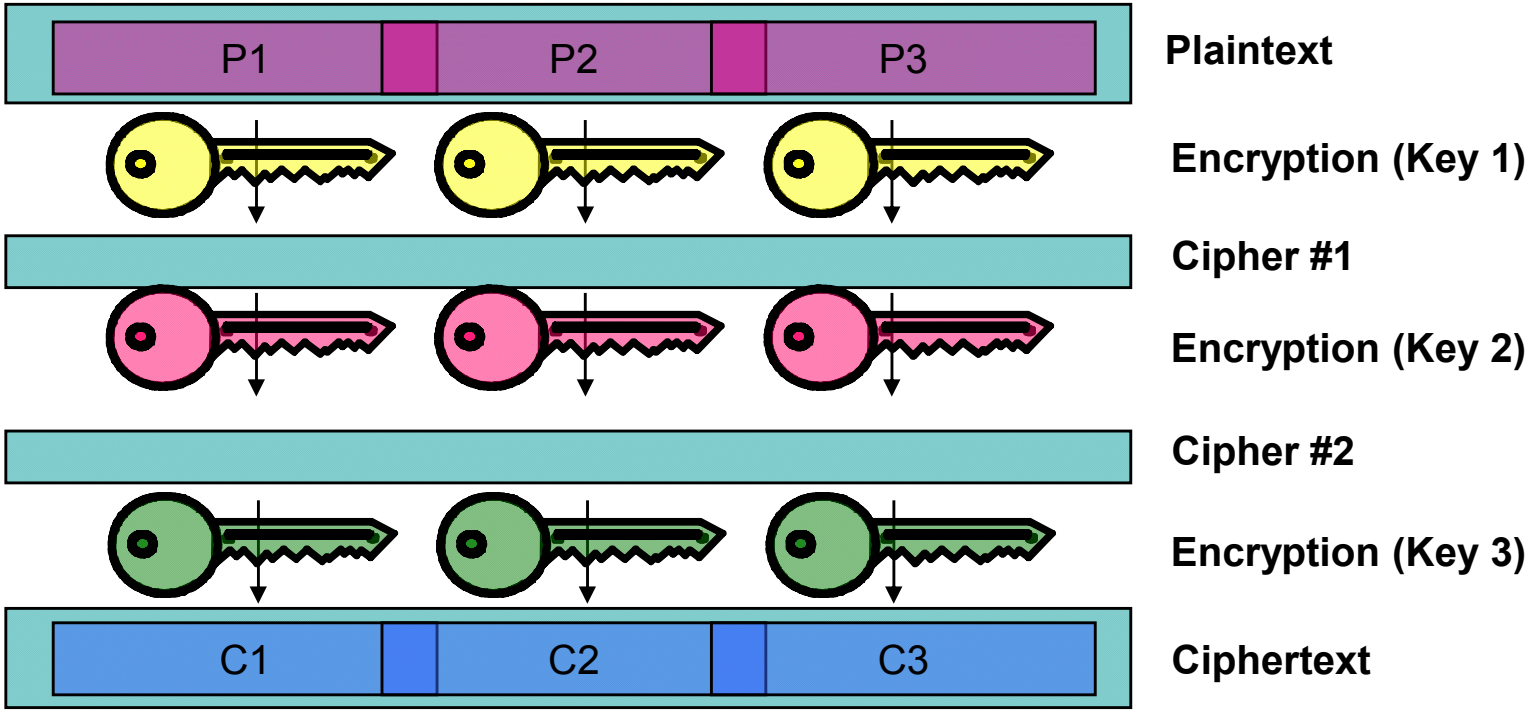
Data Encryption Standard (DES)
→ Triple DES

- Die Schlüssellänge des DESs mit **56 Bit reicht seit einigen Jahren nicht mehr aus**.
- Die rechnerische, praktische Sicherheit ist nicht mehr gegeben!

- Eine Lösung dieses Problems ist die **mehrfache Verschlüsselung eines Inputblocks mit unterschiedlichen Schlüsseln**.
- Damit kann die Schlüssellänge für die vollständige Suche vergrößert werden.
- Diese Methode wird mit **Triple-DES** bezeichnet.
- Es gibt drei verschiedene Varianten des Triple-DES.

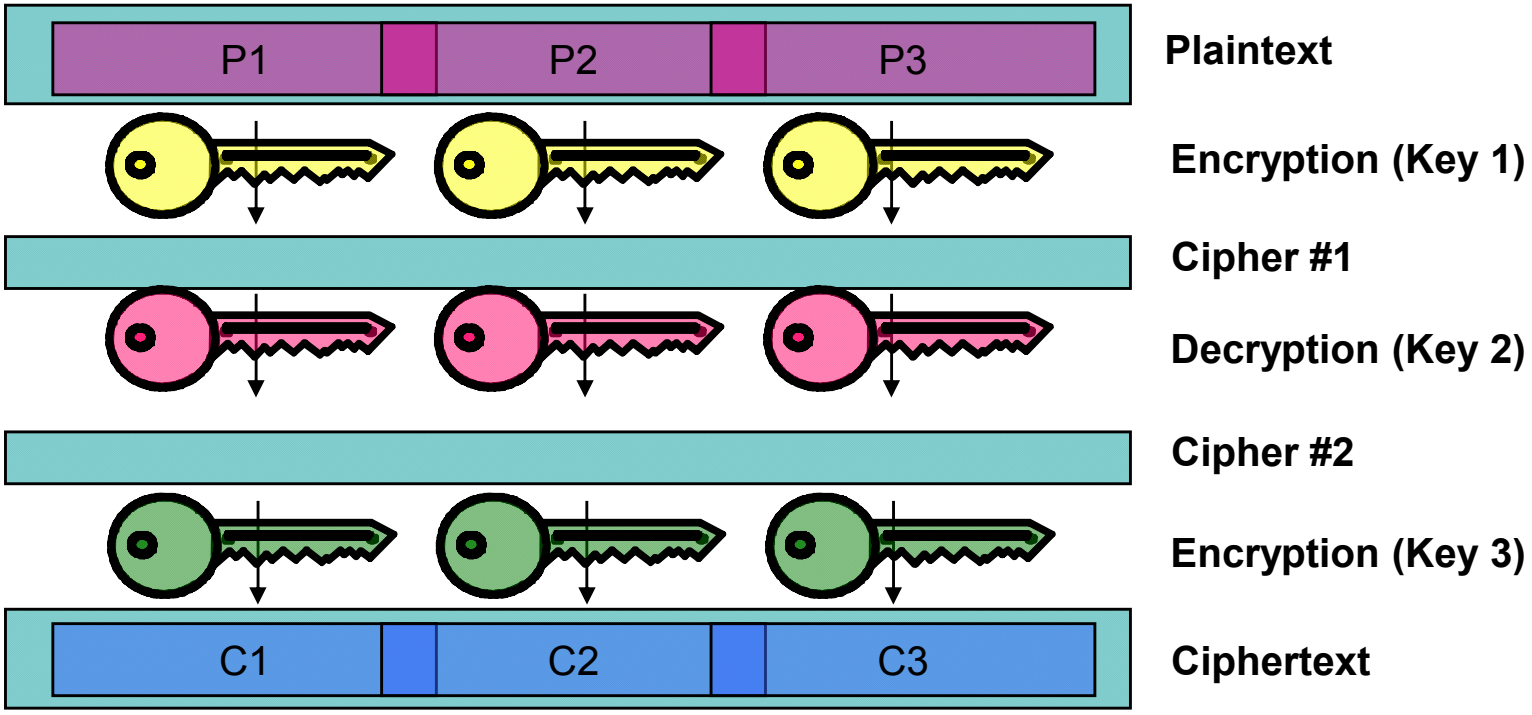
Data Encryption Standard (DES)
→ Triple DES (EEE3)

- DES Verschlüsselung wird dreimal durchlaufen (DES-EEE3)
  - 3-DES Verschlüsselung mit **3 unterschiedlichen Schlüsseln**
  - Schlüssellänge 168 Bit (3 X 56 Bit)



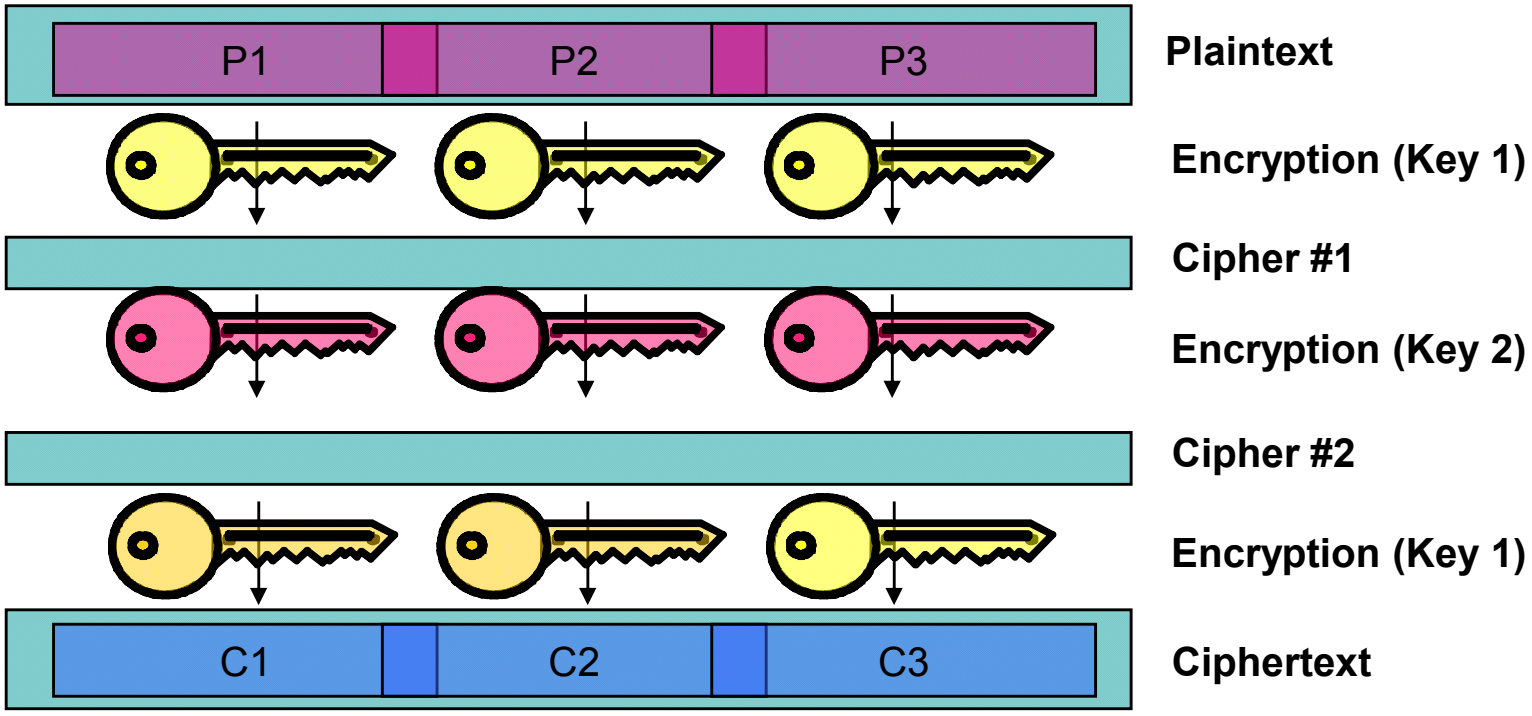
Data Encryption Standard (DES)
→ Triple DES (EDE3)

- DES Verschlüsselung wird dreimal durchlaufen (DES-EDE3)
  - 3-DES Verschlüsselung mit **3 unterschiedlichen Schlüsseln**
  - Encrypt-**Decrypt**-Encrypt
  - Schlüssellänge 168 Bit (3 X 56 Bit)



Data Encryption Standard (DES)
→ Triple DES (EEE2)

- DES Verschlüsselung wird dreimal durchlaufen (DES-EEE2)
  - 3-DES Verschlüsselung mit **2 unterschiedlichen Schlüsseln**
  - Erste und dritte Operation verwenden den gleichen Schlüssel
  - Schlüssellänge 112 Bit (2 X 56 Bit)
  - Vorteil, wenn Schlüssel für einen längeren Zeitraum gleich sind!

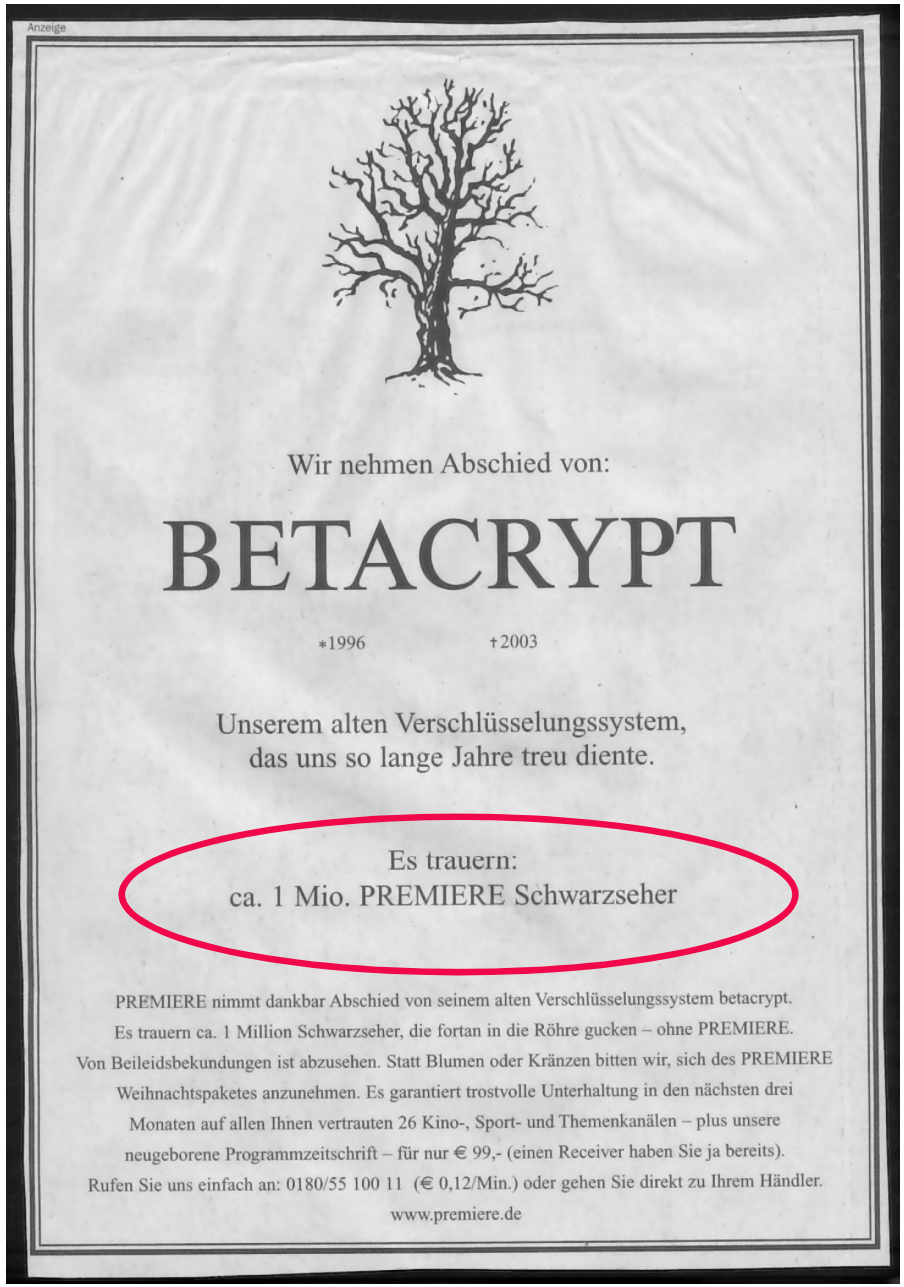


Symmetrische Verschlüsselungsverfahren
→ Bewertung der kryptographischen Stärke

- In der Regel werden die kryptographischen Verfahren, die entwickelt werden, der Kryptologen Gemeinde (mehrere 100 Mathematiker in der Welt) zur Verfügung gestellt, damit die Kryptoanalyse beginnen kann.
- Vorstellung der Verfahren mit allen Design-Aspekten auf öffentlichen Konferenzen, wie z.B. Eurocrypt, Crypto, Asiacrypt, ...
- Erst, wenn **nach ca. 5 Jahren** keiner es geschafft hat, dass Verfahren zu brechen, gilt ein Verfahren als **praktisch sicher**.
- Allerdings werden auch solche Verfahren regelmäßig von Mathematischer Seite gebrochen nach den 5 Jahren
- Eine Garantie für die Sicherheit des Verfahrens ist damit nicht gegeben
- Allerdings minimiert sich so die Gefahr

Symmetrische Verschlüsselungsverfahren
→ Bewertung: Beispiel (1/3)

FAZ, 03.11.03



Anzeige

Wir nehmen Abschied von:

BETACRYPT

*1996 †2003

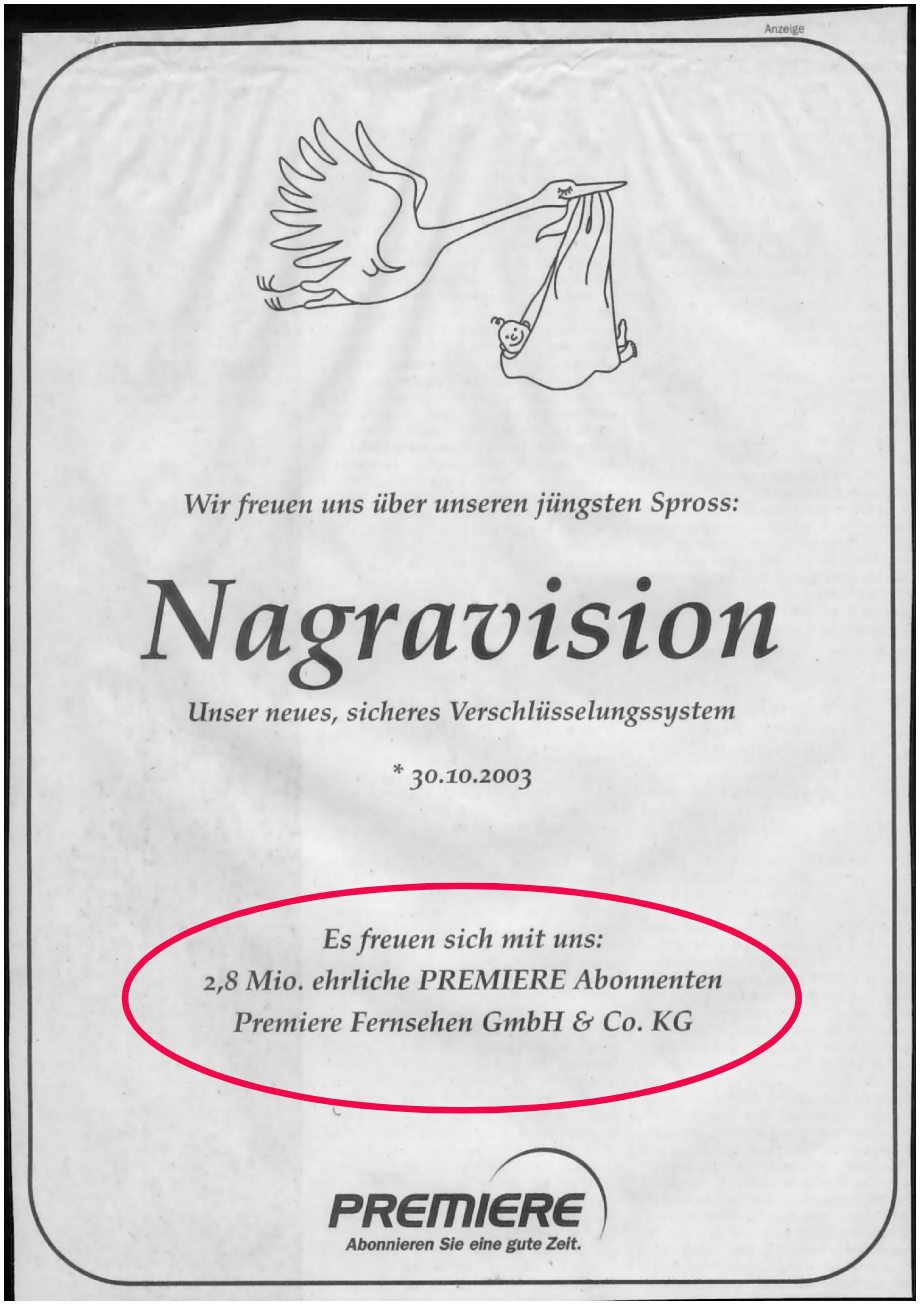
Unserem alten Verschlüsselungssystem,
das uns so lange Jahre treu diente.

Es trauern:
ca. 1 Mio. PREMIERE Schwarzseher

PREMIERE nimmt dankbar Abschied von seinem alten Verschlüsselungssystem betacrypt. Es trauern ca. 1 Million Schwarzseher, die fortan in die Röhre gucken – ohne PREMIERE. Von Beileidsbekundungen ist abzusehen. Statt Blumen oder Kränzen bitten wir, sich des PREMIERE Weihnachtspaketes anzunehmen. Es garantiert trostvolle Unterhaltung in den nächsten drei Monaten auf allen Ihnen vertrauten 26 Kino-, Sport- und Themenkanälen – plus unsere neugeborene Programmzeitschrift – für nur € 99,- (einen Receiver haben Sie ja bereits). Rufen Sie uns einfach an: 0180/55 100 11 (€ 0,12/Min.) oder gehen Sie direkt zu Ihrem Händler.
www.premiere.de

Symmetrische Verschlüsselungsverfahren
→ Bewertung: Beispiel (2/3)

FAZ, 03.11.03



Symmetrische Verschlüsselungsverfahren
→ Bewertung: Beispiel (3/3)

Neue Premiere Verschlüsselung: Kriminelle Hacker sehen schwarz

- **Umstellung auf NDS Videoguard® bei Satellitenkunden**
- **Neues Nagravisionsystem parallel im Einsatz**
- **Smartcardtausch bei allen Sat-Abonnenten startet im 2. Quartal 2008**
- **Wechsel Verschlüsselungssystem für Abonnenten denkbar einfach**
- **Keine neuen Receiver notwendig**

München, 15. April 2008. Premiere fährt eine Doppelstrategie im Kampf gegen kriminelle Hacker. Ab dem 2. Quartal 2008 wird neben einer neuen Version von Nagravision als Verschlüsselungstechnologie NDS Videoguard® für die Ausstrahlung der Premiere Abo-Programme über Satellit eingeführt. Damit schützt Premiere seine exklusiven Film- und Sport-Programme mit den neuesten Entwicklungen der beiden weltweit führenden Verschlüsselungsdienstleister. Alle neuen Receiver werden mit NDS Videoguard® ausgestattet.

NDS Videoguard® ist das Verschlüsselungssystem der NDS Group, einer Tochtergesellschaft der News Corporation. Bis 2012 wird der Abo-Sender seine Satellitenkunden und Haushalte, die ihr Programm von kleineren, privaten

Quelle: http://info.premiere.de/inhalt/de/medienzentrum_news_uk_15042008.jsp

Das Auswahlverfahren des AES (1/3)
→ National Institute of Standards and Technology (NIST)

- Ziel war die Suche nach einem neuen Standard Verschlüsselungsalgorithmus für die nächsten 20 Jahre.
- Im September 1997 hat NIST einen Wettbewerb ausgeschrieben
  - 21 Algorithmen sind eingereicht worden
  - 15 Algorithmen wurden evaluiert (haben den Anforderungen genügt)
  - davon 5 besonders in der letzten Runde
- Die beste Kombination aus
  - Sicherheit
  - Leistungsfähigkeit
  - Effizienz
  - Implementierbarkeit
  - Flexibilität
- Wichtige Voraussetzungen
  - patentfrei und lizenzfrei

  damit der AES eine möglichst große Verbreitung findet.

Das Auswahlverfahren des AES (2/3)
→ National Institute of Standards and Technology (NIST)

Country	Candidate Algorithm	Submittor(s)
Australia	LOKI97	Lawrie Brown, Josef Pieprzyk, Jennifer Seberry
Belgium	RIJNDAEL	Joan Daemen, Vincent Rijmen
Canada	CAST-256	Entrust Technologies, Inc.
	DEAL	Outerbridge, Knudsen
Costa Rica	FROG	TecApro Internacional S.A.
France	DFC	Centre National pour la Recherche Scientifique (CNRS)
German	MAGENTA	Deutsche Telekom AG
Japan	E2	Nippon Telegraph and Telephone Corporation (NTT)
Korea	CRYPTON	Future Systems, Inc.
USA	HPC	Rich Schroeppel
	MARS	IBM
	RC6	RSA Laboratories
	SAFER+	Cylink Corporation
	TWOFISH	Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson
UK/Israel/Norway	SERPENT	Ross Anderson, Eli Biham, Lars Knudsen

Das Auswahlverfahren des AES (3/3)
→ National Institute of Standards and Technology (NIST)

- Am 2. Oktober 2000 entschied sich NIST für den **Rijndael** vom Forschungslabor der belgischen Universität Leuven (Computer Security and Industrial Cryptography - COSIC)
  - **ein europäischer Algorithmus für einen US-Standard!**
  - **ist für die amerikanische Sicherheit verantwortlich!**
- In 2000 wurde der Rijndael, AES FIPS - Standard (FIPS - Federal Information Processing Standard)
- **Der AES hat den DES ersetzen !**
- Das **Wechseln von Algorithmen** in großen Anwendungen wie z.B. Electronic cash ist **sehr komplex**!

Advanced Encryption Standard (AES)
→ Fakten

- **Entwickelt von Joan Daemen und Vincent Rijmen**
  - Rijndael (Aussprache von Rijndael: Reign Dahl)
  - Patentfrei
- **Block Cipher mit variabler Blocklänge**
  - 128, 192 und 256 Bit
- **Variabler Schlüssellänge**
  - 128, 192 und 256 Bit
- **DES Ersatz als Advanced Encryption Standard (AES)**
  - FIPS Standard

Advanced Encryption Standard (AES)
→ Übersicht

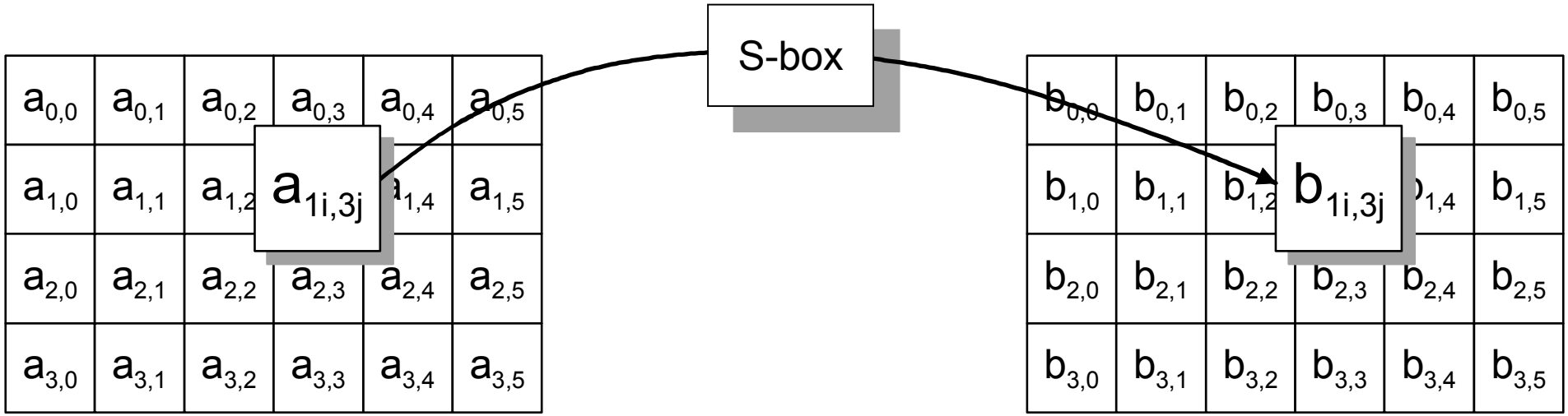
- Der AES ist ein Produktverschlüsselungsverfahren, welches aus mehreren Runden besteht.
- Die Länge der zu verschlüsselnden Blöcke und die Länge des Schlüssels kann 128, 192 oder 256 Bit betragen.
- Die Anzahl der Runden hängt von der Blocklänge und der Schlüssellänge ab und beträgt 10,12 oder 14.

Schlüssellänge (Bit)	**Blocklänge (Bit)**		
	128	192	256
128	10	12	14
192	12	12	14
256	14	14	14

- Jede der Runden vom AES besteht aus einer Reihe von Byte-orientierten Transformationen, in denen die Stärken vieler anderer Verschlüsselungs-Algorithmen kombiniert wurden.
- Diese eingesetzten Operationen haben sich bei anderen Verschlüsselungsverfahren in der Vergangenheit als widerstandsfähig gegenüber Angriffen erwiesen.

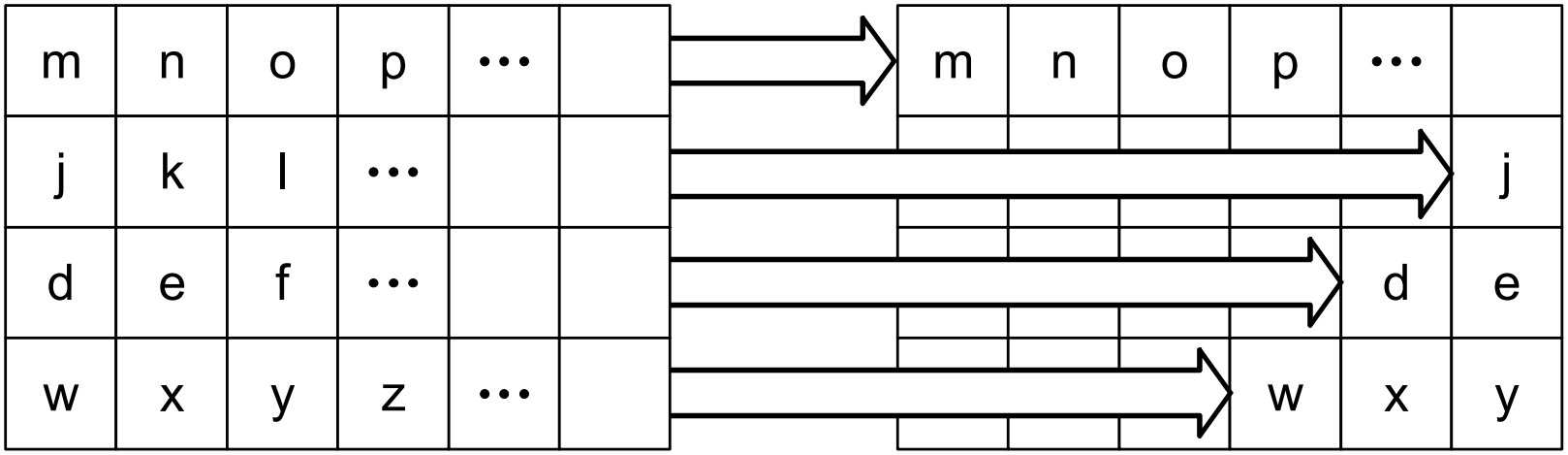
Advanced Encryption Standard (AES)
→ ByteSub-Transformation

- Die in einem zweidimensionalen Array abgelegten Zeichen des Klartext-Blocks werden zunächst der sogenannten ByteSub-Transformation unterworfen.
- Es handelt sich um eine nichtlineare Substitution der einzelnen Bytes, die über eine Tabelle (S-Box) festgelegt wird.
- Die folgende Abbildung zeigt die Transformation für den Fall einer Blocklänge von 192 Bit, bei denen der Block in einem Array von 6 x 4 Bytes abgelegt ist.



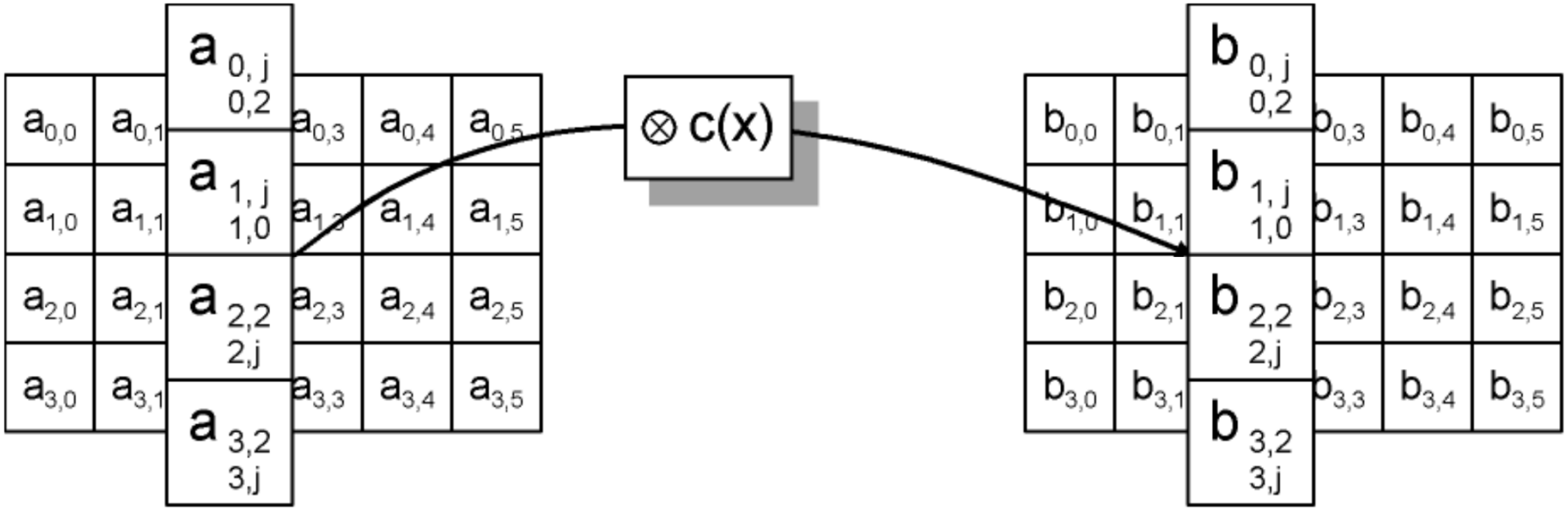
Advanced Encryption Standard (AES)
→ ShiftRow-Transformation

- Die Bytes werden anschließend der ShiftRow-Transformation unterworfen, bei der die Zeilen des Arrays bis auf die ersten zyklisch geshiftet werden.
- Jede Zeile wird um eine andere Anzahl von Bytes geshiftet.



Advanced Encryption Standard (AES)
→ MixColumn-Transformation

- Die MixColumn-Transformation unterwirft jeder Spalte des Arrays einer Multiplikation mit einem festen Polynom.



Advanced Encryption Standard (AES)
→ AddRoundKey-Transformation

- In der abschließenden AddRoundKey-Transformation wird der aus dem geheimen Schlüssel ermittelte Rundenschlüssel mit dem Array durch ein bitweises XOR verknüpft.

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$

$\oplus$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$	$k_{0,4}$	$k_{0,5}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$	$k_{1,4}$	$k_{1,5}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$	$k_{2,4}$	$k_{2,5}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$	$k_{3,4}$	$k_{3,5}$

$=$

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$	$b_{0,4}$	$b_{0,5}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	$b_{1,4}$	$b_{1,5}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$	$b_{2,4}$	$b_{2,5}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$	$b_{3,4}$	$b_{3,5}$

- In der letzen Runde vom AES wird die MixColumn-Transformation überschlagen und direkt in die AddRoundKey-Transformation verzweigt.

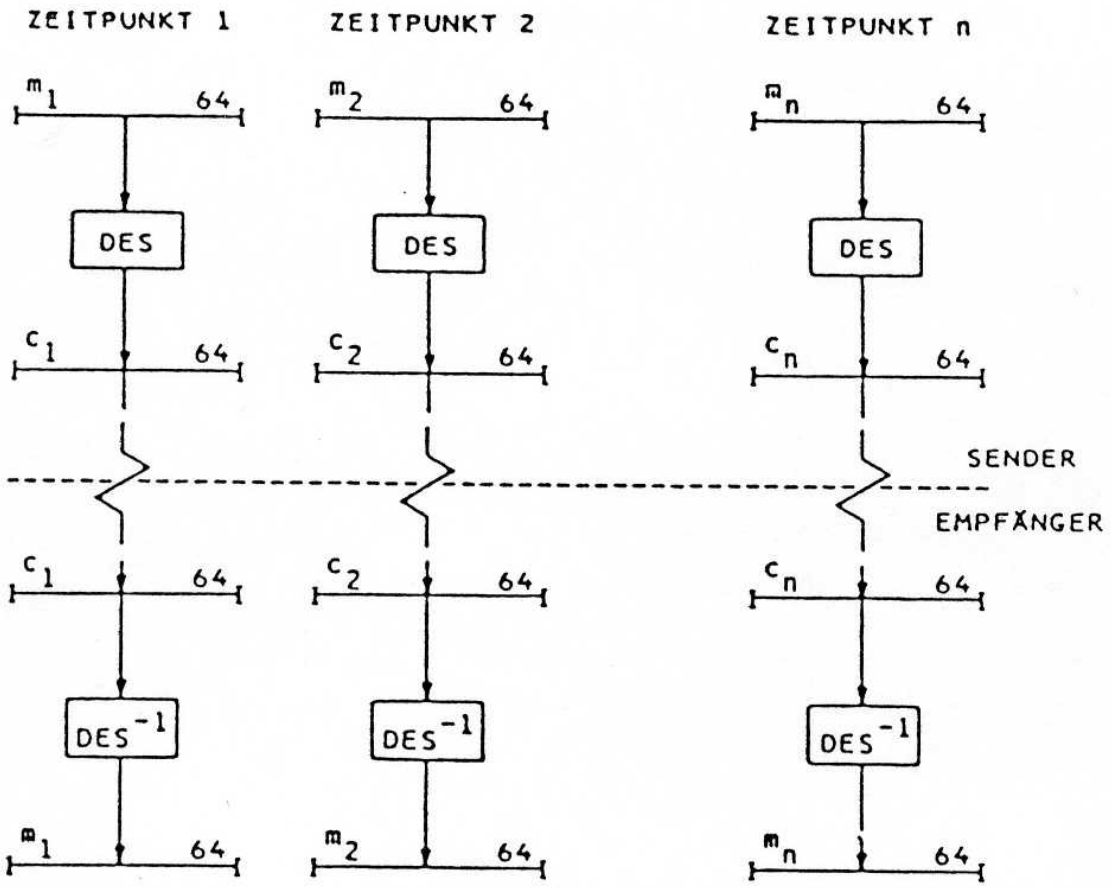
Advanced Encryption Standard (AES)
→ Rundenschlüssel

- Die in den einzelnen Runden benutzten Rundenschlüssel werden aus dem originalen Schlüssel durch eine Expansions-Funktion berechnet.
- Über XOR, zyklische Shifts und einen Tabellen-Lookup werden vor Beginn der Ver- bzw. Entschlüsselung alle Rundenschlüssel berechnet.
- Dabei wird ein Puffer der Länge (Blocklänge in Bit) * (Anzahl der Runden + 1) gefüllt, aus dem die jeweiligen Rundenschlüssel dann entnommen werden.
- Die ersten Ns Bit (Ns = Schlüssellänge) des Puffers entsprechen dem Schlüssel in unverfälschter Form, alle anderen jeweils Ns Bits entstehen aus dem vorherigen Ns Bits durch eine zyklische Permutation und einer Substitution, die der ByteSub-Transformation ähnelt.
- Vor dem Beginn der ersten Runde wird eine initiale AddRountKey-Transformation durchgeführt, die den Klartext mit dem ersten Rundenschlüssel verknüpft.
- Die Entschlüsselung erfolgt analog zur Verschlüsselung.

Blockverschlüsselung
→ Einleitung

- Die Algorithmen DES, AES und IDEA gehören zur Familie der Blockverschlüsselungen, bei denen in einem Ver- bzw. Entschlüsselungsvorgang jeweils ein Block von 64 Bits (192/256) verändert wird.
- Diese Blockverschlüsselungs-Algorithmen können in verschiedenen **Betriebsarten** oder **Modes of Operation** ausgeführt werden.
- Die verschiedenen Betriebsarten bieten eine **unterschiedliche Sicherheit**, die auf der anderen Seite aber auch **verschiedenen Aufwand** erforderlich macht.
- Es gibt vier verschiedene Betriebsarten:
  - ECB-Mode (Electronic Code Book Mode)
  - CBC-Mode (Cipher Block Chaining Mode)
  - CFB-Mode (Cipher Feedback Mode)
  - OFB-Mode (Output Feedback Mode)
- Die vier Betriebsarten werden anhand des DESs erläutert.

Modes of Operation
→ Electronic Code Book Mode (ECB): Verfahren



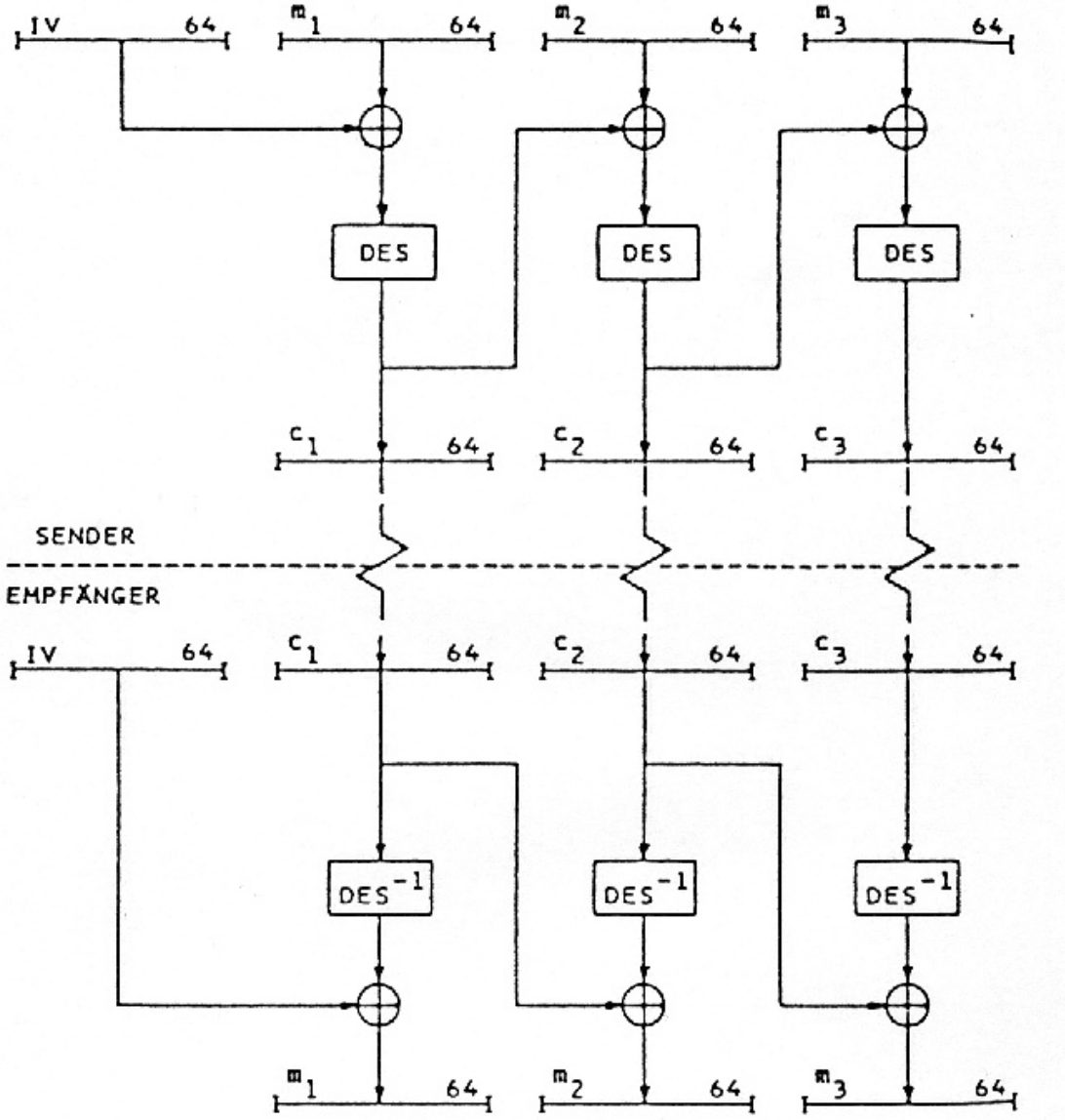
- Der **ECB-Mode stellt die Standardverschlüsselung** dar, die jeweils auf einem Block von 64 Bits operiert und diesen unabhängig von anderen Blöcken verschlüsselt.
- Die Nachricht wird in z.B. 64-Bit Blöcke zerlegt, die dann einzeln hintereinander verschlüsselt werden.

Modes of Operation
→ Electronic Code Book Mode (ECB): Eigenschaften

- Falls innerhalb einer Nachricht ein **gleicher 64-Bit Klartext-Block** auftritt, ergibt dies auch einen **gleichen 64-Bit Schlüsseltext-Block!**
- Aufgrund dieser Eigenschaft ist die ECB-Betriebsart nur für spezielle Anwendungen sinnvoll, bei denen Wiederholungen oder häufig auftretende Folgen nicht vorkommen!
- Falls die Blockgrenze zwischen Ver- und Entschlüsselung verloren geht (z.B. durch den Verlust eines Bits), so geht die Synchronisation zwischen Ver- bzw. Entschlüsselung verloren, bis die richtige Blockgrenze wiederhergestellt wird.
- Die Ergebnisse aller Entschlüsselungsoperationen sind dann fehlerhaft.

Modes of Operation
→ Cipher Block Chaining Mode (CBC): Verfahren

- Der Cipher Block Chaining Mode verschlüsselt einen Block, der vor der Verschlüsselung jeweils mit dem verschlüsselten Vorgängerblock verknüpft wird.
- Diese Art der Verschlüsselung heißt **Blockverkettung**.
- Dies erfordert für den ersten Datenblock einer Nachricht, einen bei Sender und Empfänger verfügbaren **Startwert oder Initialisierungsvektor**.

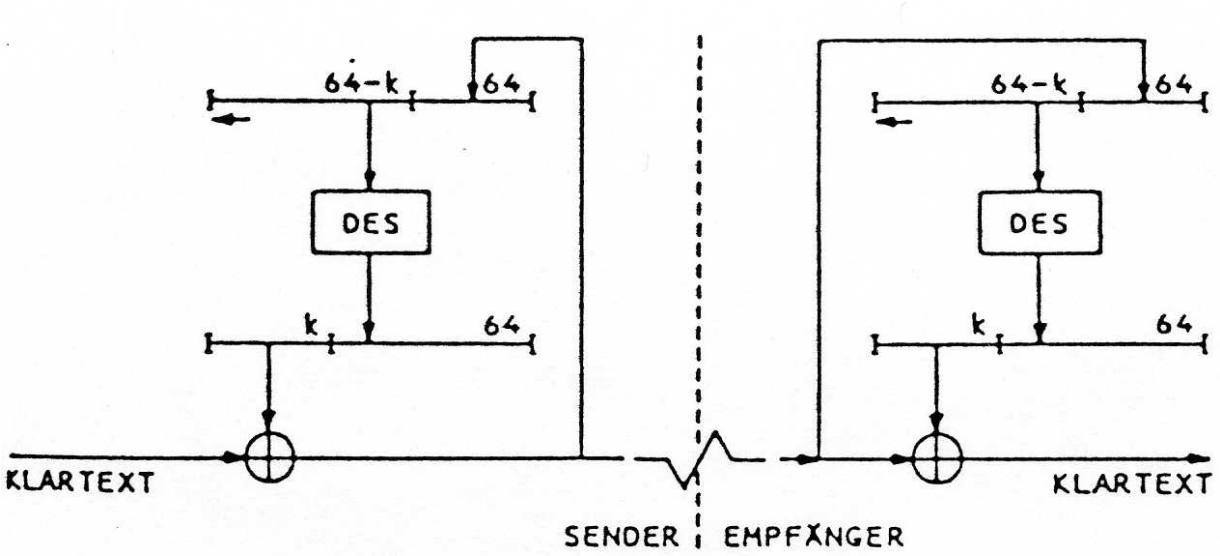


Modes of Operation
→ Cipher Block Chaining Mode (CBC): Eigenschaften

- Der CBC-Mode erzeugt **denselben Schlüsseltext, wenn derselbe Klartext mit gleichem Schlüssel und Initialisierungswert verschlüsselt wird.**
- Mit Hilfe eines variablen Initialisierungsvektors z.B. Zählnummern oder ausgehandelte Zufallszahlen kann dieses verhindert werden.
- **Identische Klartext-Blöcke innerhalb einer Nachricht führen zu verschiedenen Schlüsseltext-Blöcken (Blockverkettung).**
- Beim CBC-Mode beeinflussen ein oder mehrerer Bitfehler in einem einzigen Schlüsseltext-Block die Entschlüsselung von zwei Blöcken und zwar in dem Block, in dem der Fehler auftritt und in den folgenden.
- Wenn die Fehler im i-ten Schlüsseltextblock auftreten, beträgt die durchschnittliche Bitfehlerrate im i-ten Klartextblock 50 %.
- Im (i+1)-ten Klartextblock sind nur die Bit fehlerhaft, die direkt den fehlerhaften Bitpositionen im i-ten Schlüsseltext-Block entsprechen.
- Wie bei ECB-Mode: Falls die Blockgrenze verloren geht, geht auch die Synchronisation verloren, bis die richtige Blockgrenze wiederhergestellt wird. Die Ergebnisse aller Entschlüsselungsoperationen sind dann fehlerhaft.

Modes of Operation
→ Cipher Feedback Mode (CFB): Verfahren

- Eine bevorzugte Methode, eine Folge von Zeichen oder Bits einzeln zu verschlüsseln, ist der Cipher Feedback Mode.
- Durch die Betriebsart wird eine Blockverschlüsselung zu einer kontinuierlichen Verschlüsselung, die auf Klartexteinheiten k-Bit Länge operiert.

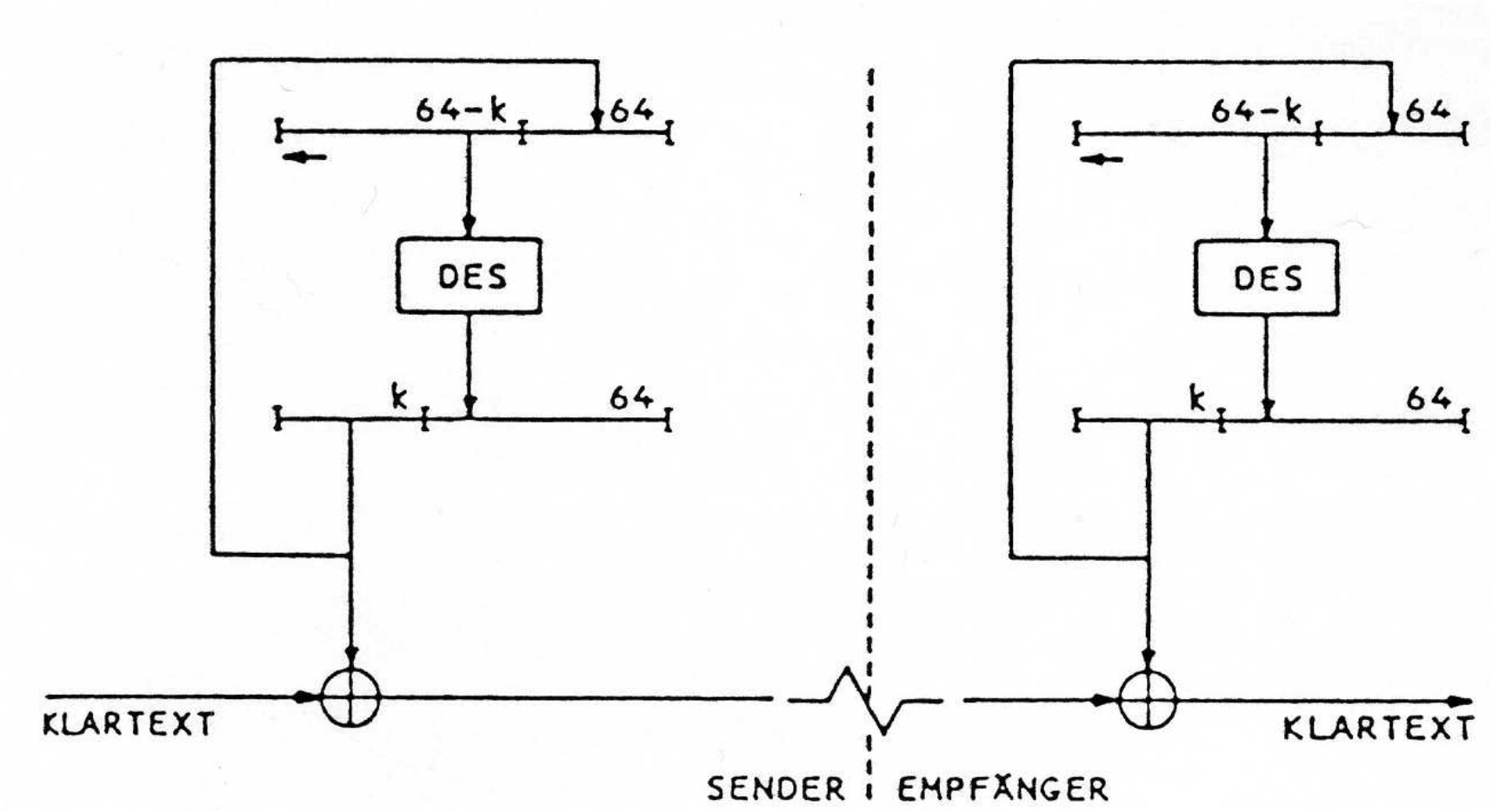


- Sowohl sender- als auch empfängerseitig arbeitet die Blockverschlüsselung im Verschlüsselungsmodus und erzeugt eine **pseudozufällige Bitfolge**, die modulo 2 (XOR) zu dem Klartextzeichen bzw. emfängerseitig zu den Schlüsseltextzeichen addiert wird.
- Zu Beginn einer Verschlüsselung muss der Input der Blockverschlüsselung mit einem Initialisierungsvektor sender- und empfängerseitig geladen werden.
- Für jedes zu verschlüsselnde Zeichen ist eine Blockverschlüsselung erforderlich, so dass diese Betriebsart nicht so effizient ist.

Modes of Operation
→ Cipher Feedback Mode (CFB): Eigenschaften

- Beim CFB-Mode beeinflussen Fehler in einem k-Bit-Block des Schlüsseltextes die Entschlüsselung des unmittelbaren verstümmelten und des folgenden Schlüsseltextes solange, bis die fehlerbehafteten Bits aus dem CFB-Eingabeblock herausgeschoben sind.
- Der erste betroffene k-Bit-Block des Klartextes ist in genau den Bitpositionen fehlerhaft, in denen der Schlüsseltext fehlerhaft ist.
- Der nachfolgende entschlüsselte Klartext hat eine durchschnittliche Bitfehlerrate von 50% solange, bis alle Fehler aus dem Eingangsblock herausgeschoben sind.
- Sind bis dahin keine zusätzlichen Fehler aufgetreten, so erscheint danach wieder der richtige Klartext.
- Diese Eigenschaft wird mit **„begrenzte Fehlerfortpflanzung“** oder mit **„selbst Synchronisation“** bezeichnet.
- Wenn die Grenzen der k-Bit-Blöcke während der Entschlüsselung verloren gehen, so geht auch die kryptographische Synchronisation verloren, bis eine erneute Initialisierung (Reinitialisierung) durchgeführt wird.
- Nach Wiederherstellung der richtigen Grenzen der k-Bit-Blöcke sind höchstens noch die folgenden 64-Bit fehlerhaft.

Modes of Operation
→ Output Feedback Mode (OFB): Verfahren



- Der Output Feedback Mode arbeitet ähnlich wie der CFB Mode, nur mit dem Unterschied, dass hier nicht das Schlüsseltextzeichen, sondern das Outputzeichen der Blockverschlüsselung in das Inputregister zurückgeführt wird.

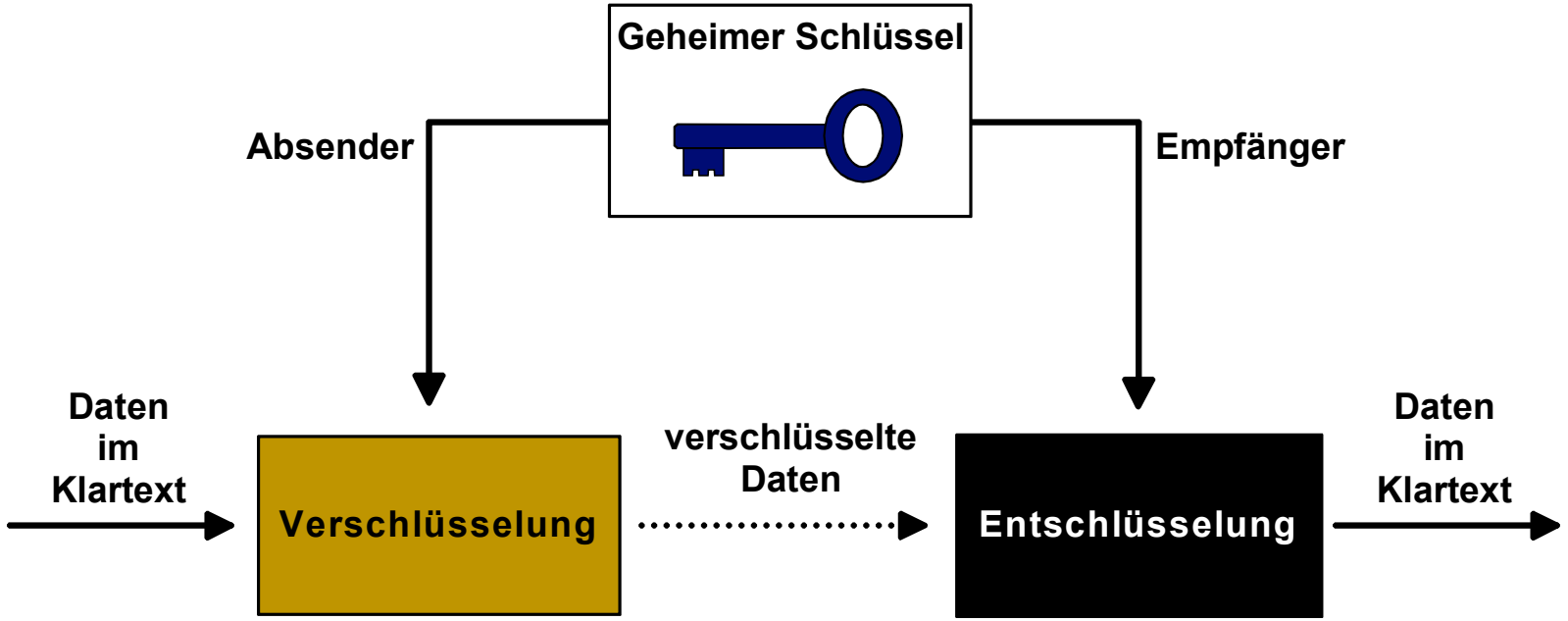
Modes of Operation
→ Output Feedback Mode (OFB): Eigenschaften

- Der OFB-Mode führt zur keiner Fehlerfortpflanzung in der resultierenden Klartextausgabe.
- Ein fehlerhaftes Bit im Schlüsseltext hat nur ein fehlerhaftes Bit im entschlüsselten Klartext zur Folge.
- Der OFB-Mode ist **nicht selbstsynchronisierend**.
  - Wenn die beiden Operationen Verschlüsselung und Entschlüsselung aus der Synchronisation geraten, muss das System wieder neu initialisiert werden.
  - Eine Reinitialisierung kann mit einem neuen Startwert bei gleichem Schlüssel durchgeführt werden.
- Dieser Mode ist für störungsanfällige Übertragungswege (z.B. Satellitenverbindung) gedacht, wo eine Fehlerfortpflanzung nicht erwünscht ist.

Modes of Operation
→ Zusammenfassung

- Auswahl des Modes of Operations in Abhängigkeit von
  - Performance, die benötigt wird und vorhanden ist (SW/HW)
  - Fehlerfortpflanzung, die gewünscht oder ungewünscht ist
  - Selbstsynchronisation, die evtl. notwendig ist
- Diese Anforderungen können unterschiedlich sein von
  - der Kommunikationsebene auf der verschlüsselt werden soll (1 oder 7 OSI)
  - die Qualität des Übertragungskanals
  - ...
- Die Modes of Operation werden typischerweise in den entsprechenden Standards festgelegt.

Symmetrische Verschlüsselungsverfahren



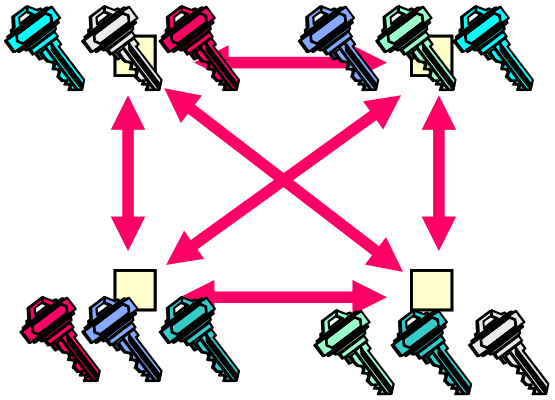
Symmetrische Verschlüsselungsverfahren
→ Überblick

	Schlüssellänge	Als stark betrachtet
DES	56	
Triple DES (2-keys)	112	
Triple DES (3-keys)	168	
IDEA	128	
RC2, RC 4, RC 5	variable	
Blowfish	variable	◆
CAST	128	
AES (Rijndael)	variable	◆

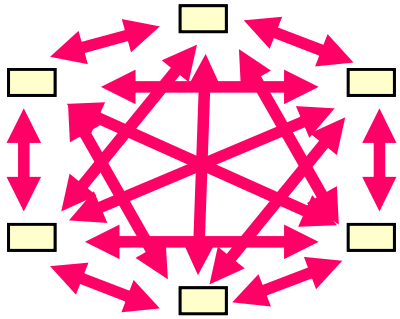
Symmetrische Verschlüsselungsverfahren
→ Nachteil

- **Key Management**
  - gesicherter Schlüsselaustausch
  - regelmäßiger Schlüsselwechsel

- **Beispiel**
  - n Partner benötigen **n*(n-1)/2 keys**
  - n=12: 66 keys
  - n=1000: 499500 keys



n=4 6 keys



n=6 12 keys

Symmetrische Verschlüsselungsverfahren
→ Anwendungen

- **Vertraulichkeit**
  - Dokumentenverschlüsselung (z.B. S/MIME)
  - Daten Verschlüsselung innerhalb von Netzwerken (IPSec, SSL, ...)
  - Datei-, Verzeichnis- oder Plattenverschlüsselung
  - Anwendungen (ec-Cash, usw.)

- **Integrität**
  - CBC MAC, HMAC (siehe One-Way Hashfunktionen)

Verschlüsselung
→ Weitere Eigenschaften

- **Verschlüsselte Daten sind wie Zufallszahlen**
  - kann leicht festgestellt werden
  - aus diesem Grund Steganographie
- **Verschlüsselung und Kompression**
  - Erst komprimieren, dann verschlüsseln!
  - Zufallszahlen können nicht komprimiert werden, da sie keine Redundanz besitzen!

Steganographie
→ Abgrenzung (1/2)

- **Kryptographie:**
  Nachrichten werden unverständlich gemacht, aber offen übermittelt.

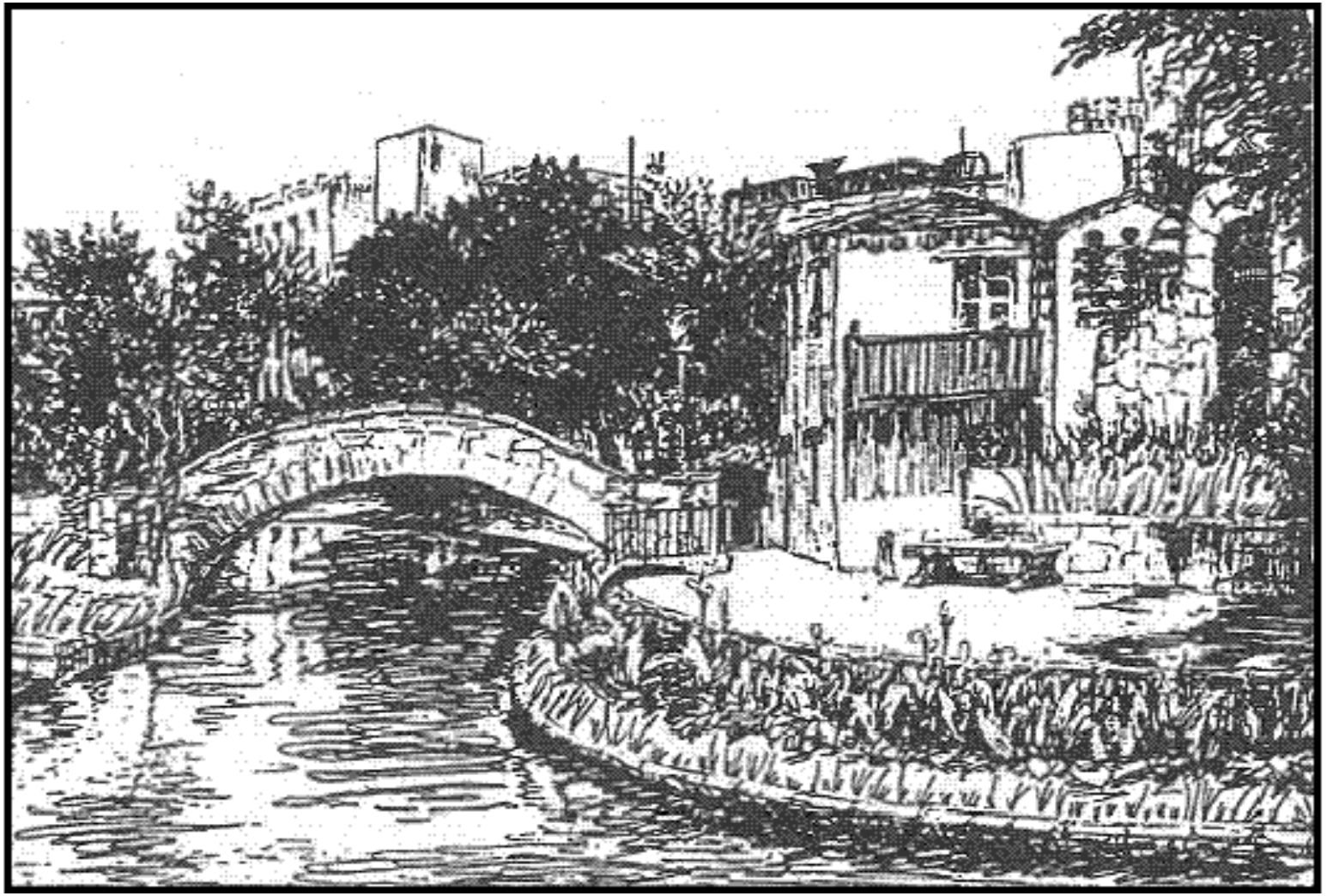
- **Steganographie:**
  Die Existenz von Nachrichten wird verborgen.

  - Klassische Beispiele
    - Unsichtbare Tinte
    - Mikrofilm
    - Semagramme: Informationen in Bildern verstecken
    - verdeckte Kommunikationskanäle

  - Moderne Beispiele
    - Verstecken von Bits in
      - Textdateien
      - Bilddateien
      - Audiodateien
      - Videokonferenzen
      - …

Steganographie
→ Abgrenzung (2/2)

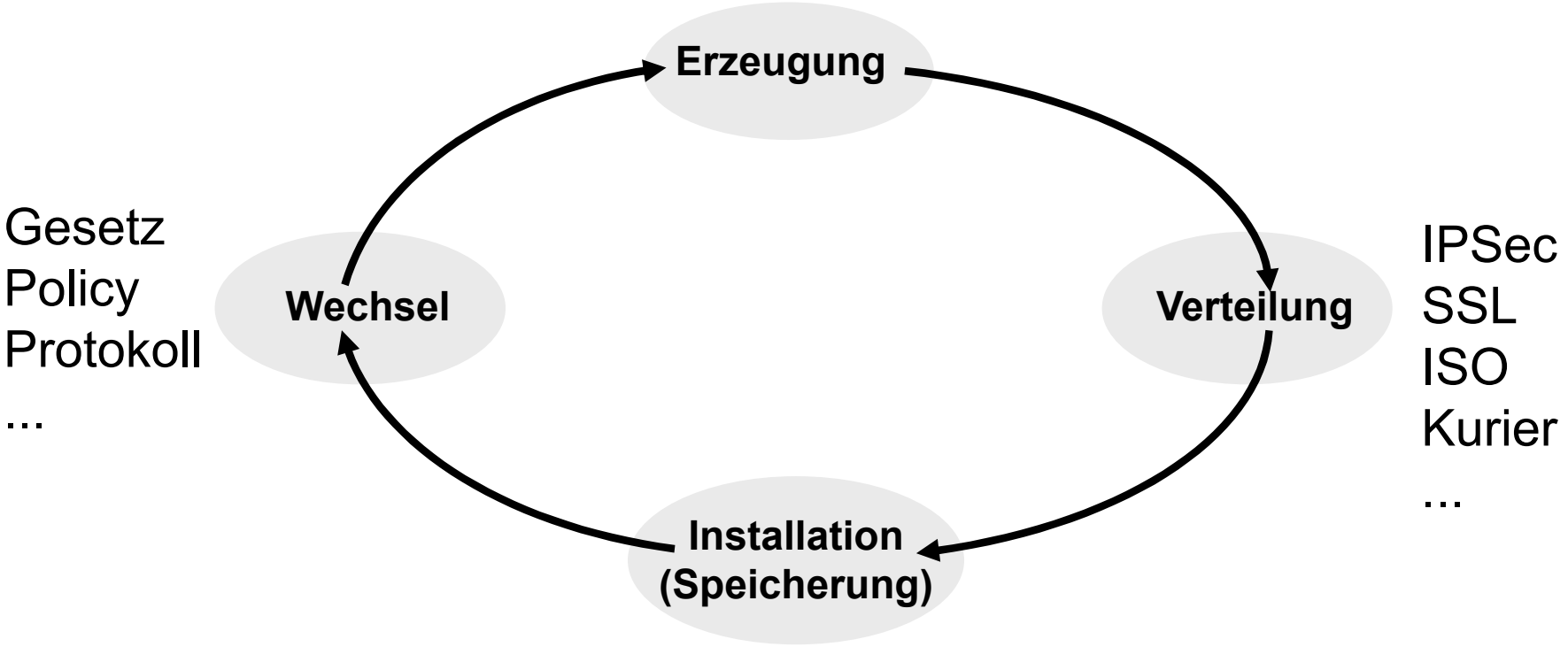
- Semagramm:
  Die Nachricht steht im Morsecode, der aus kurzen und langen Grashalmen von der Brücke entlang des Flusses und auf der kleinen Mauer gebildet wird.



[aus F.L. Bauer (Buchtitel s. Folie 70); er wiederum hat es aus D. Kahn, The Codebreakers, S. 523]

Lebenslauf eines Schlüssels
→ Grundlagen

Zufallszahlengenerierung



Gesetz
Policy
Protokoll
...

IPSec
SSL
ISO
Kurier
...

Software (verschlüsselt)
SmartCards
Sicherheits-Module
...

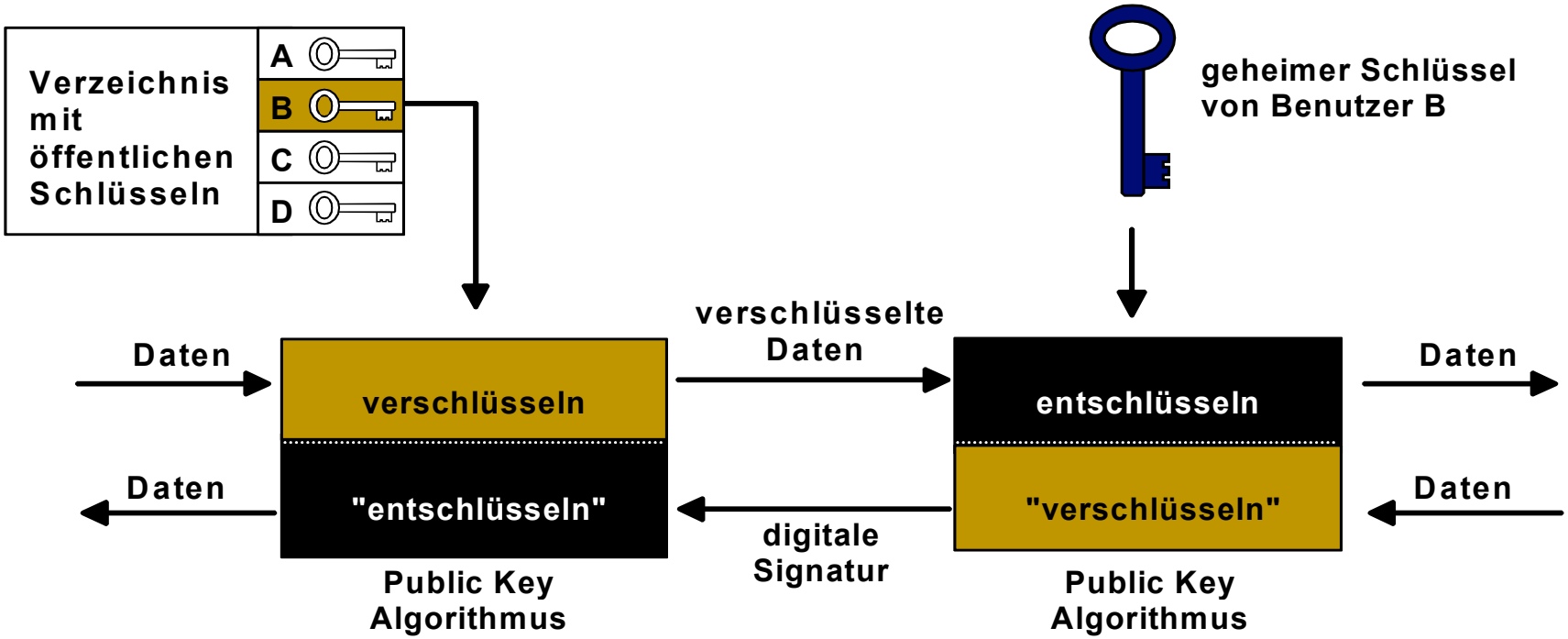
Die Stärke eines kryptographischen Verfahrens
→ Grundlagen

- **Verfahren:** anerkanntes, sicheres Verfahren (z.B. AES)
  Schlüssellänge: groß genug (>160, 192, 256)
- **richtige Implementierung** (Standard)
- **Schlüsselgenerierung:** Gütekriterien, Streuung, Periodizität, Gleichverteilung
- **sichere Schlüsselspeicherung**
  - in verschlüsselter Form *(siehe Angriffe: ISSE)*
  - auf einer SmartCard
  - in einem Security Modul
    - sicheres Verfahren zur Aktivierung
- **sichere Distribution** (Key Management)

Inhalt

- Ziele
- Einführung
- Grundlagen der Verschlüsselung
- Elementarverschlüsselungen
- Symmetrische oder Private-Key Verschlüsselungsverfahren
- **Asymmetrische oder Public-Key Verschlüsselungsverfahren**
- One-Way-Hashfunktionen
- Zusammenfassung

Asymmetrische Verschlüsselungsverfahren
→ Idee



Asymmetrische Verschlüsselungsverfahren
→ Einführung (1/4)

- Um das klassische Problem der Kryptographie, die Schlüsselverteilung zu erleichtern, wurden Verfahren entwickelt, die mit sogenannten „öffentlichen Schlüsseln“ oder Public-Keys arbeiten.
- Diese Verfahren werden vielfach auch als asymmetrische Verfahren bezeichnet.
- Man geht dabei von Verschlüsselungsverfahren aus, bei denen zur Entschlüsselung ein anderer Schlüssel als zur Verschlüsselung verwendet wird, wobei die folgenden zusätzlichen Forderungen erhoben werden:
  - Der Schlüssel zur Entschlüsselung ist nicht aus dem Schlüssel zur Verschlüsselung ableitbar.
  - Die Verschlüsselung kann nicht durch einen Angriff mit ausgewählten Klartext gebrochen werden.
- Wenn diese beiden Bedingungen erfüllt sind, gibt es keinen Grund mehr, den Schlüssel zur Verschlüsselung geheimzuhalten.

Asymmetrische Verschlüsselungsverfahren
→ Einführung (2/4)

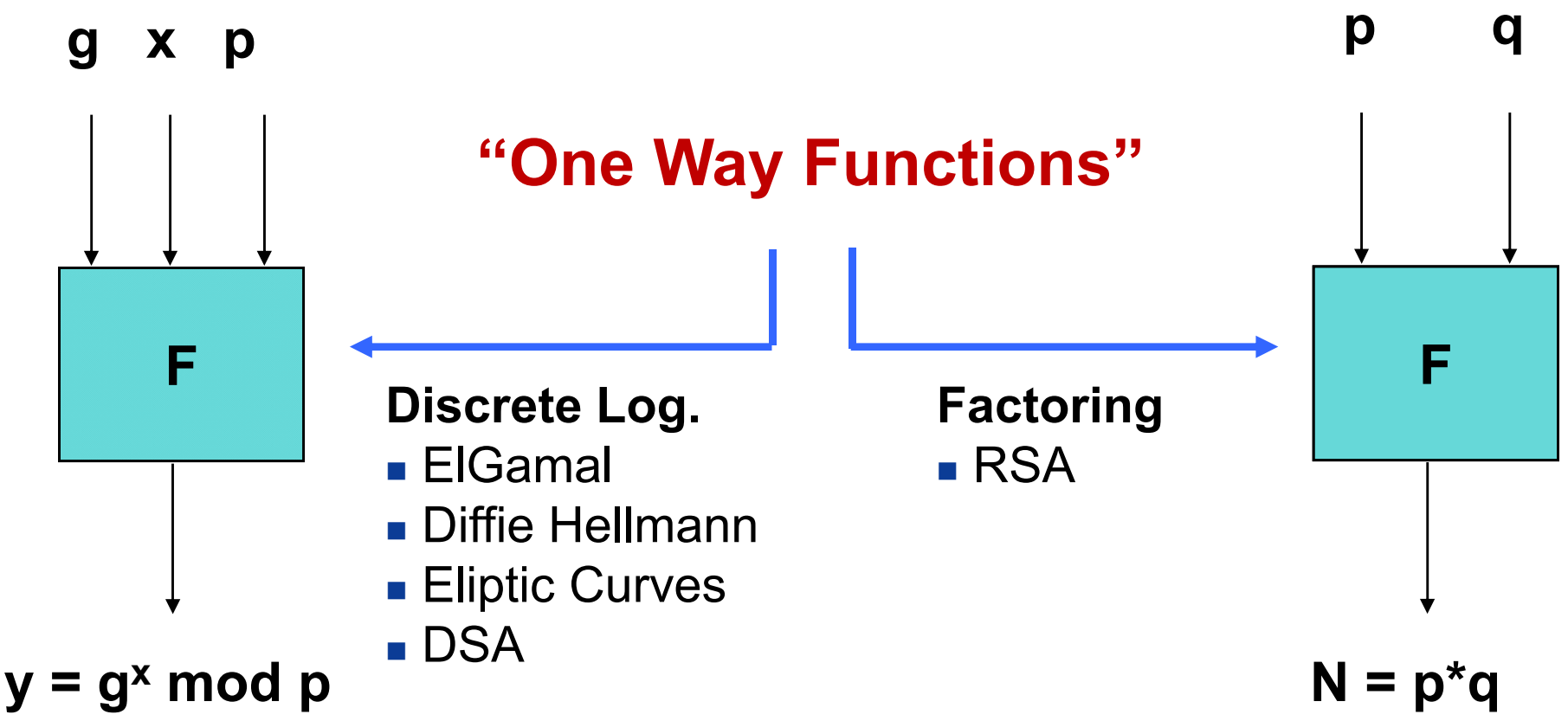
- Man kann sogar den für jeden Teilnehmer gültigen Schlüssel veröffentlichen.
- Daher nennen sich diese Verfahren **Public-Key-Verfahren**.
- Der Schlüssel zur Verschlüsselung wird als „öffentlicher Schlüssel“ bezeichnet und der Schlüssel zur Entschlüsselung „privater“ oder „geheimer Schlüssel“ genannt.
- Will A eine Nachricht an B senden, so entnimmt er den öffentlichen Schlüssel von B einem öffentlichen Verzeichnis, verschlüsselt die Nachricht damit und sendet sie an B.
- Da nur B den zugehörigen geheimen Schlüssel kennt, und da dieser Schlüssel weder aus dem öffentlichen Schlüssel noch aus der verschlüsselten Nachricht bestimmt werden kann, ist B tatsächlich der Einzige, der die Nachricht wieder entschlüsseln kann.
- Damit ist also eine sichere Kommunikation möglich, ohne dass dazu vorher eine geheime Schlüsselübermittlung zwischen A und B oder von dritter Seite an beide notwendig wäre.

Asymmetrische Verschlüsselungsverfahren
→ Einführung (3/4)

- Da vom Grundsatz her der geheime Schlüssel immer aus dem öffentlichen Schlüssel ableitbar ist, werden für Public-Key-Verfahren Algorithmen gewählt, die auf der Lösung von Problemen der Komplexitätstheorie beruhen.
- Derartige Funktionen werden auch mit **„one-way trap door“-Funktionen** bezeichnet.
- Bei „one-way“-Funktionen oder Einwegfunktionen handelt es sich um Funktionen, deren Funktionswert „leicht“ zu berechnen ist, während die Berechnung der inversen „schwierig“ oder sogar „unmöglich“ ist.
- Die Begriffe „leicht“, „schwierig“ und „unmöglich“ sollen den rechnerischen Aufwand beschreiben und hängen somit vom Entwicklungsstand der jeweiligen Computergeneration ab.
- Gibt es zu einer „one-way“-Funktion einen Parameter bzw. Schlüssel, mit dem die inverse Transformation „leicht“ zu berechnen ist, so spricht man von einer „one-way trap-door“-Funktion.

Asymmetrische Verschlüsselungsverfahren
→ Einführung (4/4)

- Ausgehend von dieser Überlegung, die Diffie und Hellman im Jahre 1976 veröffentlichten, wurden verschiedene Vorschläge für Public-Key-Verfahren gemacht.



Asymmetrische Verschlüsselungsverfahren
→ Digitale Signatur

- Eine wichtige Anwendung des Public-Key-Verfahrens ist die digitale Signatur
- Daten, die mit einem bestimmten geheimen Schlüssel verschlüsselt wurden, können nur Mithilfe des dazugehörigen öffentlichen Schlüssels wieder „entschlüsselt“ werden.
- Hat nun eine Person die Daten mit ihrem geheimen Schlüssel digital signiert, kann Mithilfe des öffentlichen Schlüssels überprüft werden, ob es wirklich diese Person war, die die Daten digital signiert hat.
- Die erfolgreich durchgeführte Überprüfung ist der Beweis für die Authentizität der Signatur.
- Mit dem Prinzip der Digitalen Signatur steht somit ein Äquivalent zur handgeschriebenen Unterschrift zur Verfügung.
- Das bekannteste Public-Key-Verfahren ist das RSA-Verfahren, mit dem gleichzeitig signiert und verschlüsselt werden kann.

Asymmetrische Verschlüsselungsverfahren
→ Authentische Schlüssel

- **Problem:**
  - Ein offenes Problem bei Public-Key-Verfahren ist die Frage, wie der öffentliche Schlüssel zum Kommunikationspartner gelangt?
  - Selbst im Fall der Verwendung öffentlicher Schlüssel müssen diese authentisch ausgetauscht werden!

- **Lösung:**
  - Eine elegante Möglichkeit, öffentliche Schlüssel authentisch auszutauschen, ist die Einrichtung eines Zertifizierungs-Systems, eines Trustcenters oder einer Public-Key-Infrastruktur.
  - Der öffentliche Schlüssel jedes Benutzers wird von einer Public-Key-Infrastruktur (Zertifizierungssystem) in Form eines **„digitalen Zertifikates“** zur Verfügung gestellt.
  - Siehe digitale Signatur und Public-Key-Infratstuktur!

Public Key-Algorithmen
→ RSA: Fakten

- 1978 entwickelt von Ron Rivest, Adi Shamir und Leonard Adleman
- Nutzbar zur Verschlüsselung, digitaler Signatur und Key Management
- RSA Patent in den USA lief am 20. September 2.000 ab.
  - Das Patent war bis dahin ein Problem bei der Umsetzung (große Firmen)
- Basiert auf dem Problem, dass das Produkt zweier großer Primzahlen nur schwer in seine Faktoren zu zerlegen ist.
  - Welches sind die Faktoren von 377?
  - Mit bekannten Faktoren (29 x 13) ist die Berechnung des Produkts dagegen einfach: 377
- Je länger die Faktoren (Primzahlen) desto höher die Sicherheit.

Public Key-Algorithmen
→ RSA: Schlüsselgenerierung (1/2)

- Suche zwei große Primzahlen **p** und **q**
- Berechne das Produkt **n**=p*q
- Wähle **e** welches eine relative Primzahl zu (p-1)(q-1) ist und kleiner als p*q → ggT(e,(p-1)(q-1))=1 (Teilerfremd)
  - Eine relative Primzahl liegt vor, wenn kein gemeinsamer Teiler vorhanden ist
  - e muss keine Primzahl sein
  - (p-1)(q-1) kann keine Primzahl sein, da es sich um eine gerade Zahl handelt
- Verwende den erweiterten Euklidischen Algorithmus um **d** zu berechnen
  - ed (mod(p-1)(q-1)) =1

Public Key-Algorithmen
→ RSA: Schlüsselgenerierung (2/2)

- Die Sicherheit des RSA-Verfahrens beruht auf der Schwierigkeit eine große Zahl in ihre Primfaktoren zu zerlegen.
- Es gibt z.B. Faktorisierungsmethoden, die mit Hilfe der Faktoren in p-1 und q-1 zu viel besseren Ergebnissen kommen.
- Aus diesem Grund sollen die Primzahlen für das RSA-Verfahren noch besondere Eigenschaften aufweisen, die mit starken Primzahlen bezeichnet werden.
- Die Eigenschaften sind für die Primzahlen p und q:
  - p ist eine große Zahl (z.B. 768 Bit)
  - p ist eine Primzahl (kann sehr unterschiedlich nachgewiesen werden)
  - p wurde zufällig ausgewählt (Zufallszahlengenerator)
  - p hat eine vorher festgelegte Länge (z.B. zwischen 500 und 520 Bit)
  - p-1 hat einen großen Primteiler r
  - p+1 hat einen großen Primteiler s
  - r-1 hat einen großen Primteiler
  - s-1 hat einen großen Primteiler.

Public Key-Algorithmen
→ RSA: Verschlüsselungsvorschrift

- **M** = Message im Klartext, **C** = Schlüsseltext
- Verschlüsselung ---> $C = M^e \bmod n$
- Entschlüsselung ---> $M = C^d \bmod n$

- "Public key" ist das Zahlenpaar (**e**,**n**)
- "Private key" ist die Nummer **d**

- **pq** (oder **n**) ist der Modulus
- **e** ist der öffentliche Exponent
- **d** ist der geheime Exponent

Public Key-Algorithmen
→ RSA: Beispiel (1/2)

p = 61 1.te Primzahl (Vernichten nachdem e und d berechnet wurden)

q = 53 2.te Primzahl (Vernichten nachdem e und d berechnet wurden)

pq = **3233** Modulus (Teil des öffentlichen Schlüssels)

e = **17** Öffentlicher Exponent (Teil des öffentlichen Schlüssels)

d = **2753** Geheimer Exponent (Geheimhalten!)

Der öffentliche Schlüssel ist (pq=n,e).

Der geheime Schlüssel ist d.

$C = encrypt(M) = (M^{17}) \bmod 3233$

$M = decrypt(C) = (C^{2753}) \bmod 3233$

Public Key-Algorithmen
→ RSA: Beispiel (2/2)

Verschlüssele(123)

= $(123^{17}) \bmod 3233$

= 337587917446653715596592958817679803 mod 3233

= 855

- Es werden effektive Lösungen zur Berechnung benötigt:
  - Algorithmen in Abhängigkeit von der CPU, Speicherplatz und Zeitanforderung
  - Hardware (Sicherheits-Model, SmartCards, TPM, ...)

Public Key-Algorithmen
→ RSA: Challenge (1/2)

- 1991 wurde die RSA-Challenge ausgerufen
- Aufforderung an Mathematiker und Informatiker, Primfaktorzerlegungen von konkreten Zahlen variabler Länger zu finden
- Meilensteine:

Stellen	Bits	Preisgeld	Lösung
129	426	100 $	In 8 Monaten mit 800 Freiwilligen, 1994
174	576	10.000 $	Uni Bonn, Dezember 2003
193	640	20.000 $	Uni Bonn, November 2005
212	704	…	...
232	768	-	Zusammenarbeit von Instituten unter Leitung von T. Kleinjung“ Dezember 2009
309	1024	100.000 $	Offen
463	1536	150.000 $	Offen
617	2048	200.000 $	Offen

Public Key-Algorithmen
→ RSA: Challenge (2/2)

- Im Mai 2007 hat Uni Bonn eine 312 stellige Zahl (1.039 Bit) faktorisiert.
- Daraufhin hat RSA die Challenge zurückgezogen, da laut dem Unternehmen das Ziel, die Darlegung der Sicherheit des Algorithmus, nun ausreichend geklärt worden sei.
- Mittels neuer Faktorisierungsalgorithmen wie dem **Quadratischen Sieb** stellen Zahlen mit 1.024 Bit kein Problem mehr für einen großen Rechnerverbund dar.
- **Viele RSA-Schlüssel benutzen noch standardmäßig 1024 Bit!**

http://www.rsa.com/rsalabs/node.asp?id=2092

Public Key-Algorithmen
→ Diffie-Hellman: Fakten

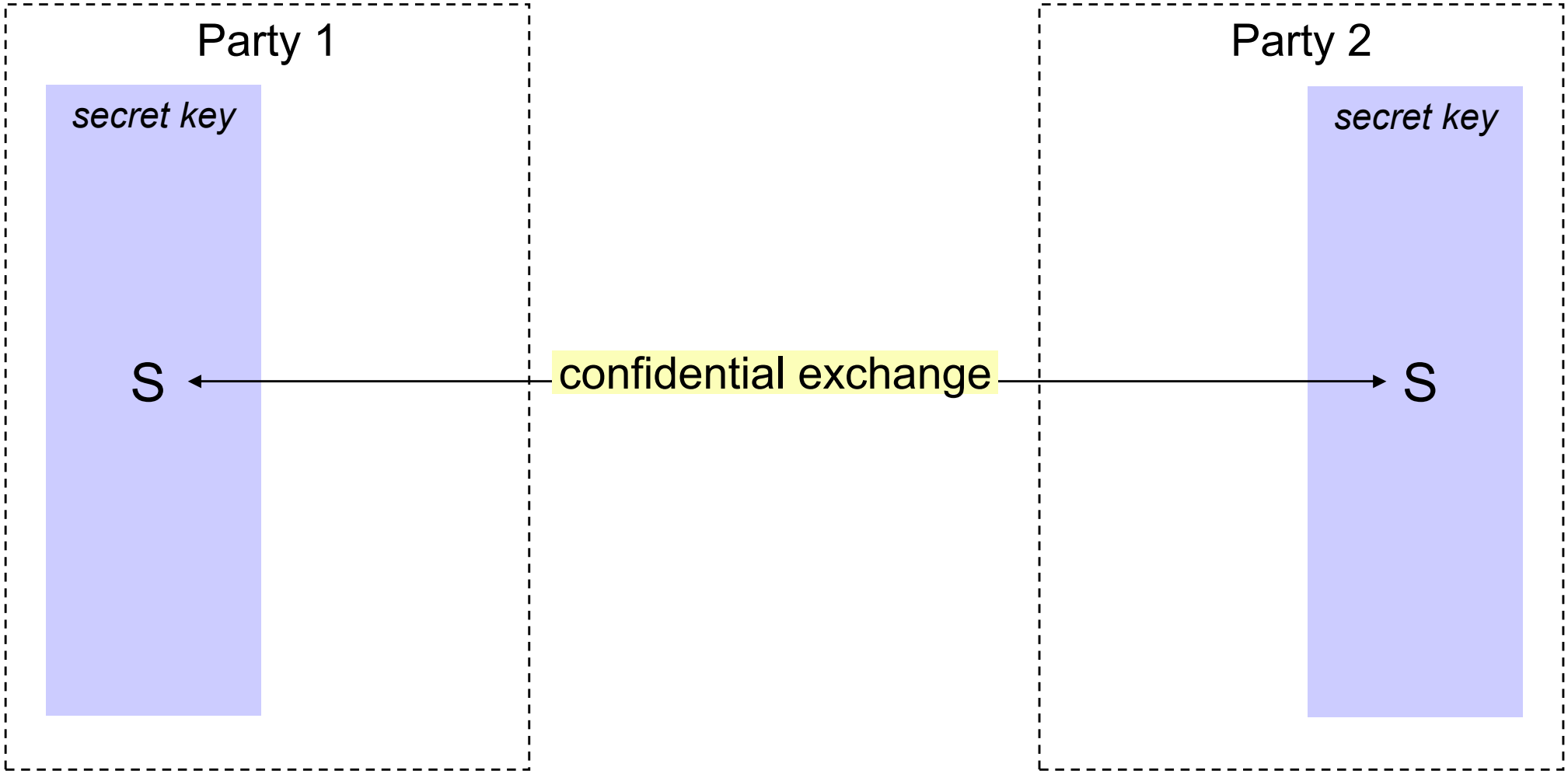
- Erster Public Key Algorithmus in 1976

- Kein Verschlüsselungsalgorithmus
  - Gesicherter Austausch eines geheimen Schlüssels
  - keine Authentisierung der Partner

- Ein Schlüsselpaar besteht aus dem geheimen Schlüssel **a** und dem öffentlichen Schlüssel **A**
  - **A** berechnet sich aus dem geheimen Schlüssel **a** mittels: $\mathbf{A=g^a \bmod n}$
    - **a** ist eine Zufallszahl
    - n eine lange Primzahl
    - g ist teilerfremd zu n → d. h. ggT(g,n) = 1
    - g und n sind öffentlich

Public Key-Algorithmen
→ Diffie-Hellman

The goal is to confidentially exchange a key ‘S’ for a following communication



Public Key-Algorithmen
→ Diffie-Hellman-Verfahren (1976)

- Vereinbarung eines gemeinsamen, geheimen Schlüssels **ohne vorherigen Austausch**!
- ***Kein*** Verschlüsselungsverfahren, ***keine*** Authentifizierung der Partner!
- Einsatz u.a. in SSL und IPSec
- Basis: Problem des diskreten Logarithmus
- **Vorgehen:**
  - Wähle eine allen Teilnehmern bekannte Primzahl q als Modulus; wähle **Einheitswurzel g** mit $g \in \{0, ..., q-1\}$ mit $\{1, ..., q-1\} = \{g^1; ..., g^{q-1}\}$
  - Schritte Teilnehmer 1 und Teilnehmer 2:
  - Wähle **geheimen** Schlüssel: **a** <= q, **b** <= q
  - Berechne **öffentlichen** Schlüssel: **A** = g^{a} mod q, **B** = g^{b} mod q
  - Berechne **gemeinsamen** Schlüssel **S**:
  - **S** = K_{12} = B^{a} mod q = A^{b} mod q = $(g^{b})^{a}$ mod q
- **Angreifer**: Aufwand **S** zu berechnen?

Public Key-Algorithmen
→ Diffie-Hellman-Verfahren (1976)

- Einheitswurzel oder Primitivwurzel ist ein Begriff aus der Zahlentheorie
- Ausgezeichnetes (durch besondere Eigenschaft) Element einer primen Restklassengruppe
  - Jedes Element kann als Potenz der Primitivwurzel dargestellt werden

q	**Primitivwurzeln (Einheitswurzelen g) modulo q**
2	1
3	2
5	2, 3
7	3, 5
11	2, 6, 7, 8
13	2, 6, 7, 11
17	3, 5, 6, 7, 10, 11, 12, 14
19	2, 3, 10, 13, 14, 15
23	5, 7, 10, 11, 14, 15, 17, 19, 20, 21
29	2, 3, 8, 10, 11, 14, 15, 18, 19, 21, 26, 27

Public Key-Algorithmen
→ Diffie-Hellman

- Jede Partei erzeugt eine geheime Zufallszahl
  (Partei 1 → a und Partei 2 → b) und
  sie vereinbaren eine öffentliche Zahl g.
  Alle Rechnungen modulus einer bekannten Primzahl q.

- $A=g^a$ und $B=g^b$ sind One way-Operationen;
  $a=\log_g A$ und $b=\log_g B$ sind für große Zahlen ***nicht*** „leicht“ lösbar.

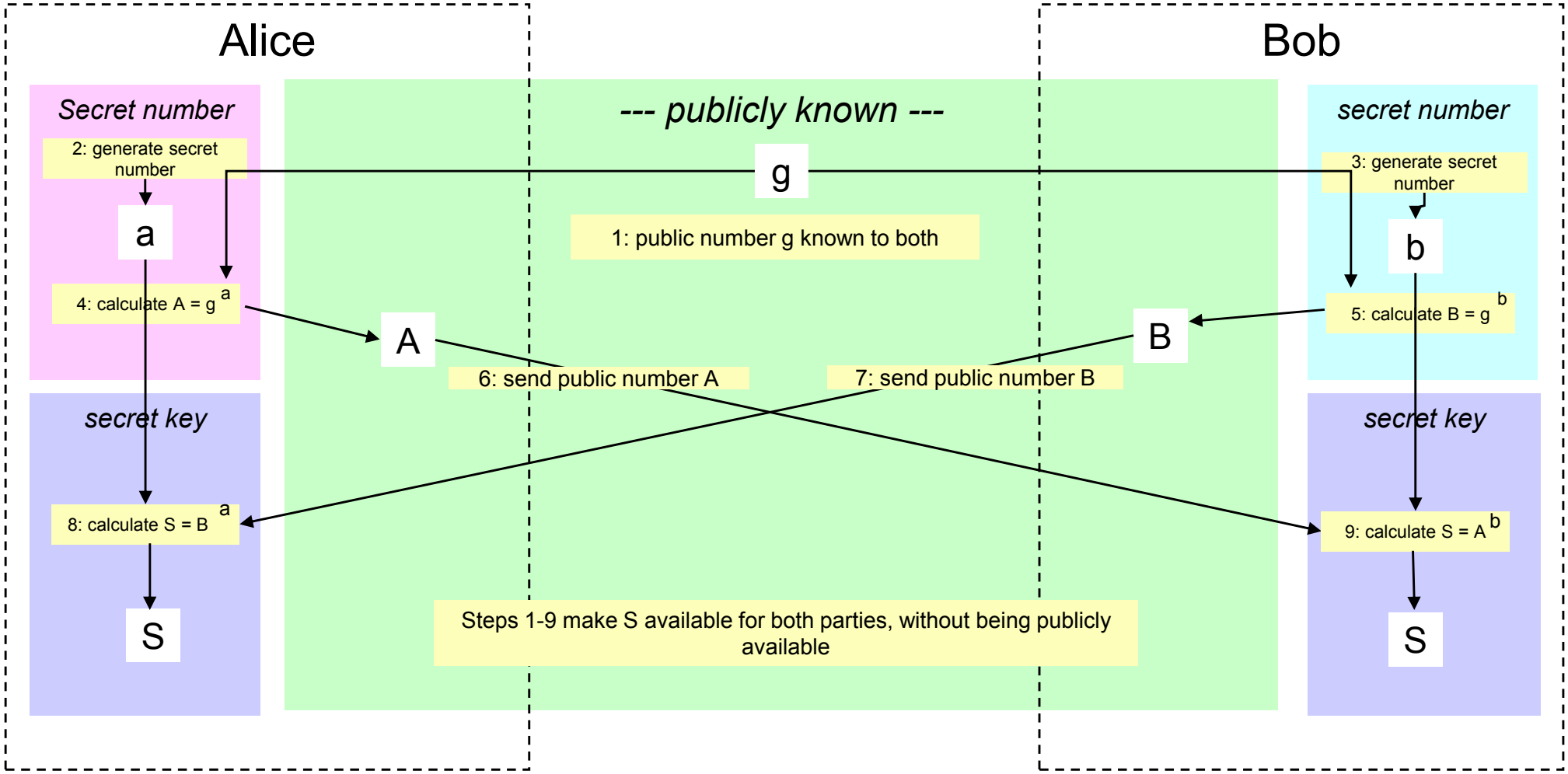
- g, A, und B werden öffentlich kommuniziert;
  diese Information reicht nicht, um S herzuleiten !

- Um S zu berechnen, braucht man beides:
  - die öffentliche Information (g; A; B) und
  - die geheime Information (a oder b) :

$$S = A^b = g^{ab} = g^{ba} = B^a$$

Public Key-Algorithmen
→ Diffie-Hellman

The procedure is based on the exchange of uncritical information!



Public Key-Algorithmen
→ Elliptische Kurven

- Elliptische Kurven basierende Kryptographie (auch bekannt als EC bzw. ECC) wird als Ersatz für RSA, DSA und Diffie-Hellman Schlüsselaustausch angepriesen.

- **Vorteile:**
  - Kürzere Schlüssel
    - Schnellere Verarbeitung
    - Geringere Speicherbedarf
    - Schnellere Kommunikation
  - besonders Interessant für SmartCards (findet Verwendung im nPA)

Public Key-Algorithmen
→ Einsatzgebiet

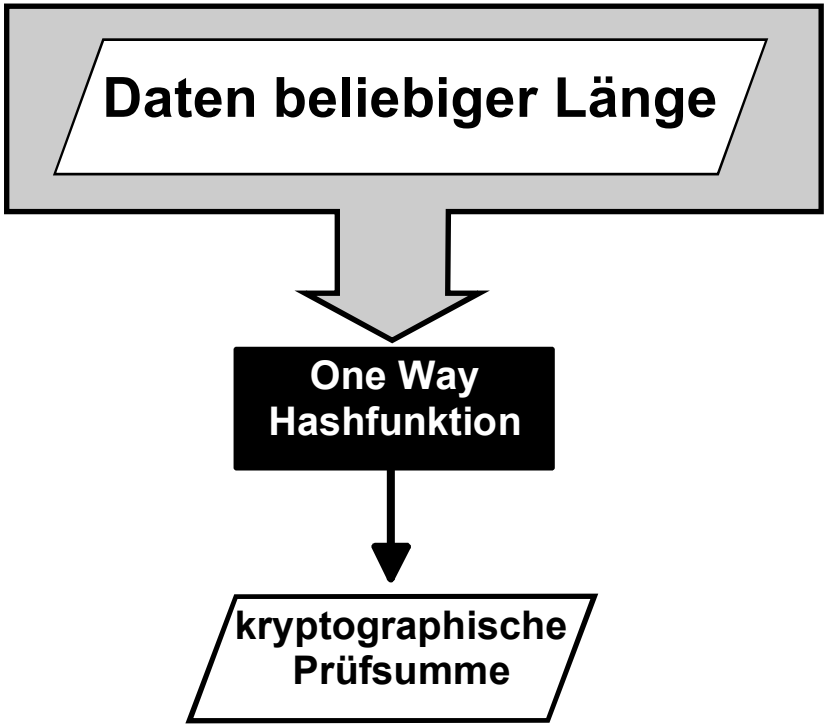
- **Wahrung der Integrität Verbindlichkeit** **(Signatur)**
- **Absenderüberprüfung** **(Zertifikat mit Signatur)**
- **Key-Management** **(Diffie Hellman und weitere Schlüsselaustausch Protokolle)**
- **Vertraulichkeit** **(Ver-/Entschlüsselung (z.B. RSA))**

Inhalt

- Ziele
- Einführung
- Grundlagen der Verschlüsselung
- Elementarverschlüsselungen
- Symmetrische oder Private-Key Verschlüsselungsverfahren
- Asymmetrische oder Public-Key Verschlüsselungsverfahren
- **One-Way-Hashfunktionen**
- Zusammenfassung

One-Way-Hashfunktionen
→ Grundlagen

- Die Digitale Signatur entspricht einer Operation mit einem **Public-Key-Verfahren und ist daher sehr rechenintensiv**.
- Um dem Aufwand zu vermindern, signiert man nicht die gesamte Information mit dem Public-Key-Verfahren, sondern erstellt eine Prüfsumme als **"Konzentrat" der Nachricht**, das digital signiert wird.
- Auf eine Nachricht, deren Länge variabel ist, wird eine sogenannte One-Way-Hashfunktion angewendet, die eine kryptographische Prüfsumme (**Hashwert**) fester kurzer Länge als Ergebnis erzeugt.
- Zu den besonderen Eigenschaften von One-Way-Hashfunktion gehört, dass die Berechnung des Funktionswertes einfach ist, während es aber praktisch unmöglich ist, systematisch einen Wert zu finden, der dieselbe kryptographische Prüfsumme ergibt.



One-Way-Hashfunktionen
→ Eigenschaften (1/2)

- H ist eine **öffentliche bekannte** Einwegfunktion

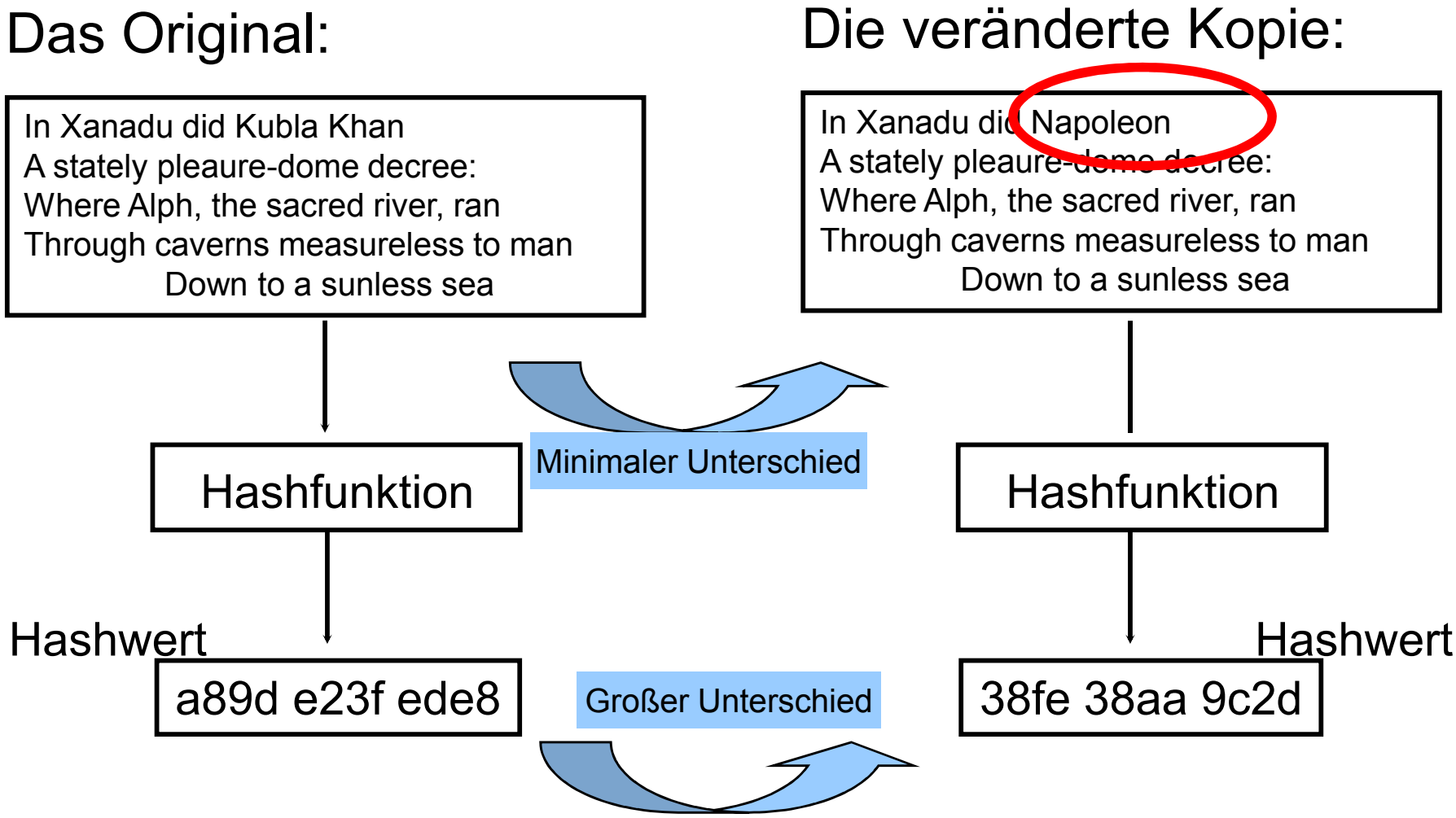
- h = H(M), h ist ein eindeutiger “Fingerabdruck” von M (Hashwert)
  - Eingabe **M kann beliebig lang sein**
  - Hashwert **h hat eine feste Länge**, z.B. 160 Bit

- H(M) ist eine One-Way Funktion (Einwegfunktion)
  - **H(M) ist einfach zu berechnen**, bei gegebenem M
  - Mit gegebenem h, ist es schwer (praktische unmöglich) M zu berechnen, sodass M = f(h) ist !

One-Way-Hashfunktionen
→ Eigenschaften (2/2)

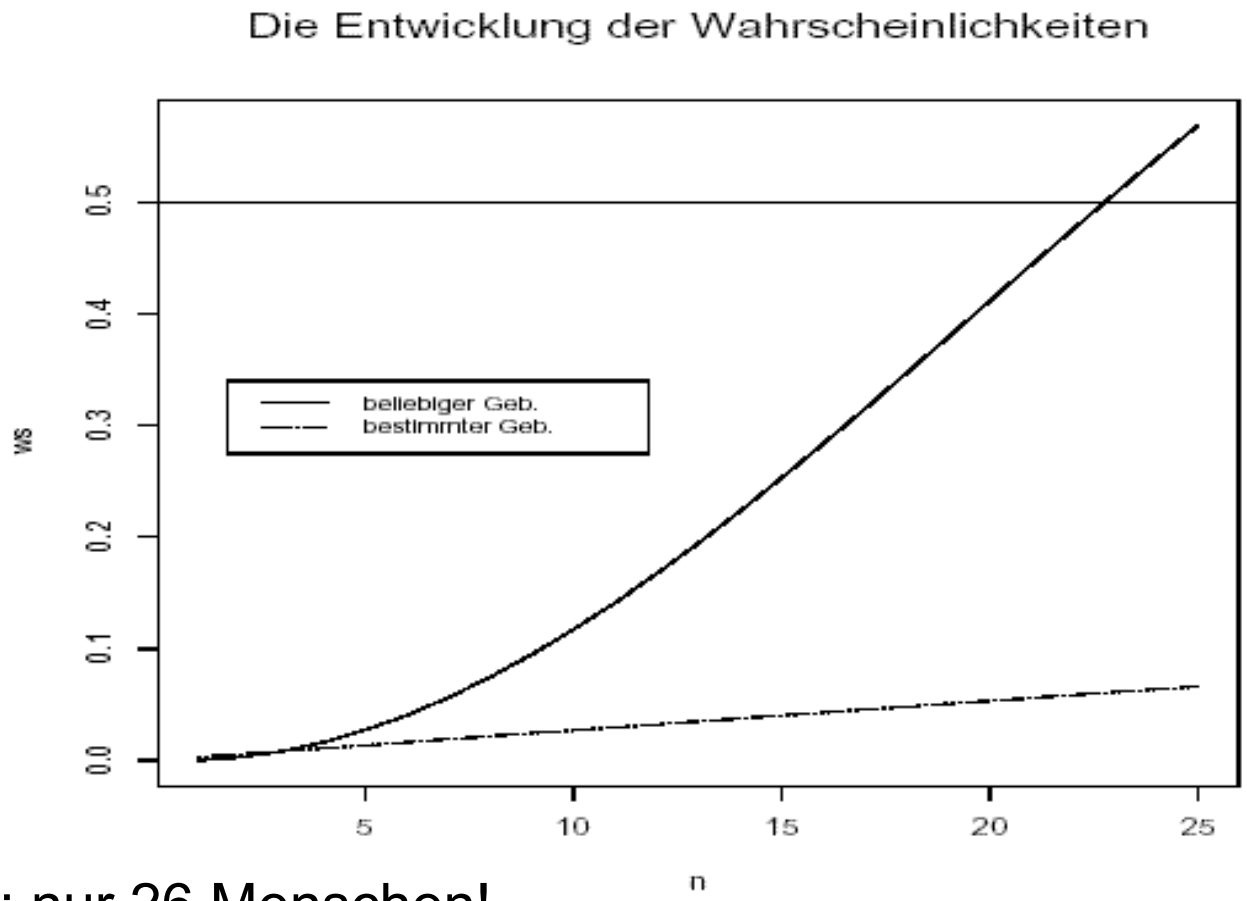
- H(M) ist **kollisionsresistent**
  - Mit gegebenem M, ist es schwer (praktisch unmöglich) eine weitere Nachricht M' zu finden, sodass H(M) = H(M') !
  - Zwei verschiedene digitale Dokumente (Nachrichten), die denselben Hashwert abbilden werden, bilden eine Kollision!
  - Die Existenz von Kollisionen ist unvermeidbar.
  - Diese ist aber nur eine theoretische Aussage.
  - Bei praktischen Anwendungen kommt es nur darauf an, dass es, wie oben verlangt, **praktisch unmöglich** ist, Kollisionen zu finden.

One-Way-Hashfunktionen
→ Arbeitsweise



Das Geburtstags-Paradox
→ Das Paradox

- Wieviele zufällig ausgewählte Menschen braucht man, damit mit p > 0,5 mindestens 2 am gleichen Tag Geburtstag haben?



- Ergebnis: nur 26 Menschen!

Das Geburtstags-Paradox
→ Die Auswirkung auf One-Way-Hashfunktionen

- **Aufgabe**
  - Ein Angreifer will ein geändertes Dokument erzeugen, das denselben Hashwert liefert, da nur der Hashwert bei einer digitalen Signatur von einem elektronischen Dokument signiert wird.

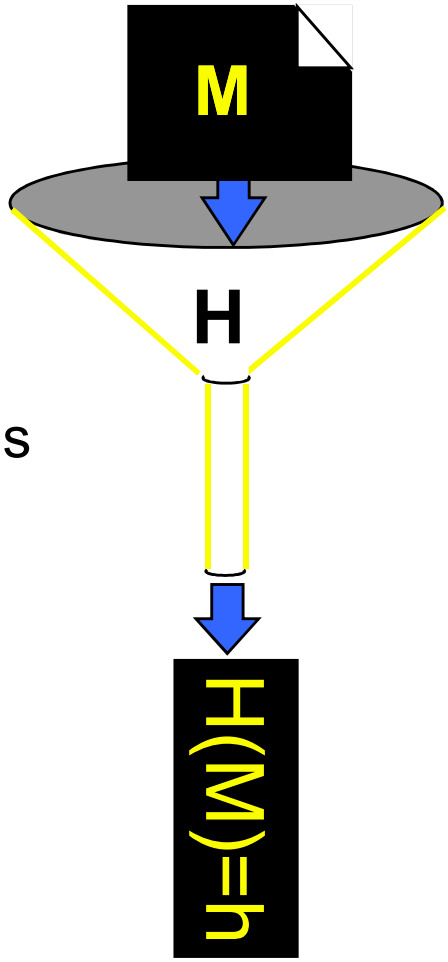
- **Problem**
  - Wenn eine Hashfunktion H Nachrichten auf Komprimate einer Länge von 60 Bit abbildet, dann braucht ein Angreifer rein statistisch „nur“ 2^{30} verschiedene Eingabenachrichten, um eine Kollision zu finden (in Texten z.B. lange Artikelnummern).

- **Anforderung an Hashfunktionen**
  - Hashfunktionen H sollten einen längeren Hashwert h haben als Schlüssellängen bei symmetrischen Verschlüsselungsverfahren, z.T. mindestens 160 Bit!

One-Way-Hashfunktionen
→ Übersicht

- **Hashfunktionen**
  - Sind praktisch unumkehrbare Komprimierungsfunktionen

- **Typische Vertreter:**
  - MD5 = Message Digest #5 1991 von Ronald Rivest als Nachfolger von MD4 vorgestellt:
  - SHA = Secure Hash Algorithm
    1994 von NIST und NSA entwickelt;
  - RIPEMD = RIPE Message Digest
    1992 innerhalb des EU-Projekts RIPE entwickelt;



One-Way-Hashfunktionen
→ MD5

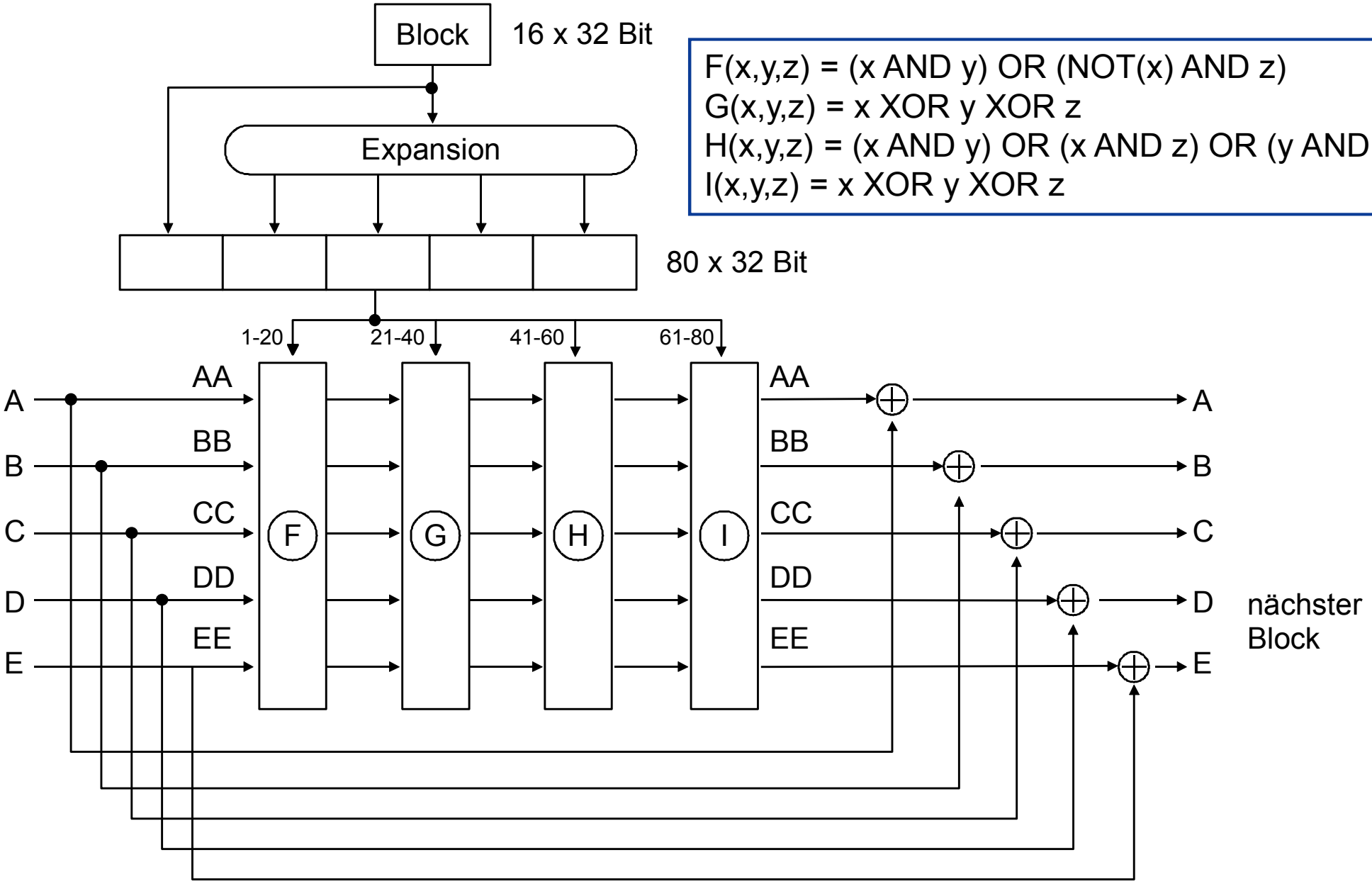
- Message Digest #5, von Ron Rivest
- Eingabe in 512 Bit Blöcken (kürzere Nachrichten werden aufgefüllt)
- Hashwert: 128 Bit (zu klein für die meisten Anwendungen!)
- 4 Runden
- Theoretische Angriffe waren erfolgreich
  - Dies hat keine Auswirkungen auf praktische Anwendungen
- **Dieses Verfahren soll nicht mehr eingesetzt werden!**
  (ist aber noch in vielen Standards vorhanden, z.B. IPSec)

One-Way-Hashfunktionen
→ SHA-1 (Secure Hash Algorithm)

- 1994 entwickelt von NIST und NSA
- Eingabe: 512 Bit Blöcke
- “Variation” von MD4
- Hashwert: **160 Bit**
- 4 Runden

- **Erweiterungen sind:**
  - SHA 256
  - SHA 384
  - SHA 512

One-Way-Hashfunktionen
→ SHA-1 (Secure Hash Algorithm)

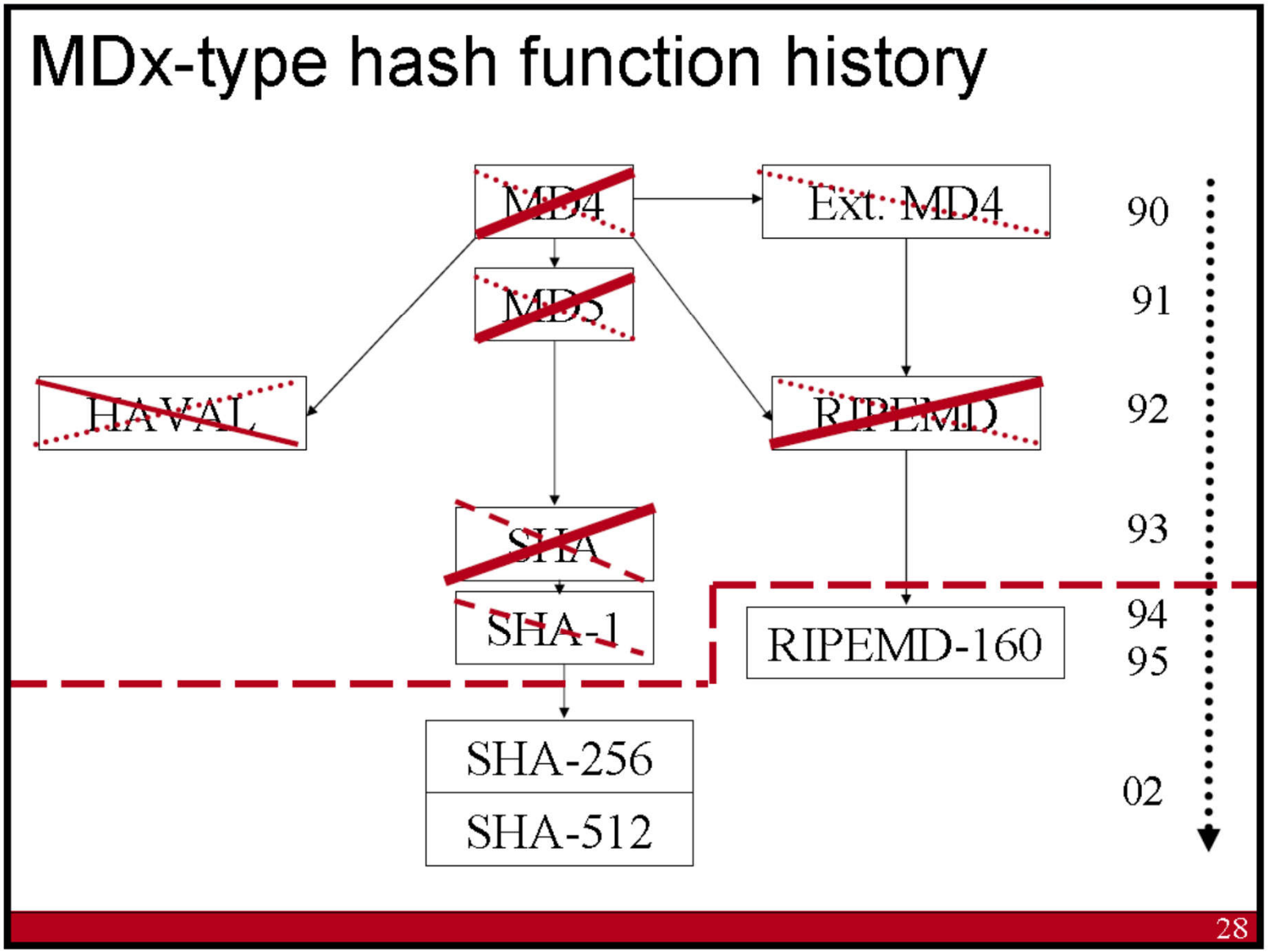


F(x,y,z) = (x AND y) OR (NOT(x) AND z)
G(x,y,z) = x XOR y XOR z
H(x,y,z) = (x AND y) OR (x AND z) OR (y AND z)
I(x,y,z) = x XOR y XOR z

One-Way-Hashfunktionen
→ RIPEMD

- 1988- 92 entwickelt im Rahmen des EU-Projektes RIPE (RACE Integrity Primitives Evaluation)
- RIPEMD = RIPE Message Digest
- Eingabe: 512 Bit Blöcke
- Hashwert: 128 oder **160 Bit (RIPEMD-160)**
- 2 x 5 Runden
- “Variation” von MD4

Geschichte der Hashfunktionen

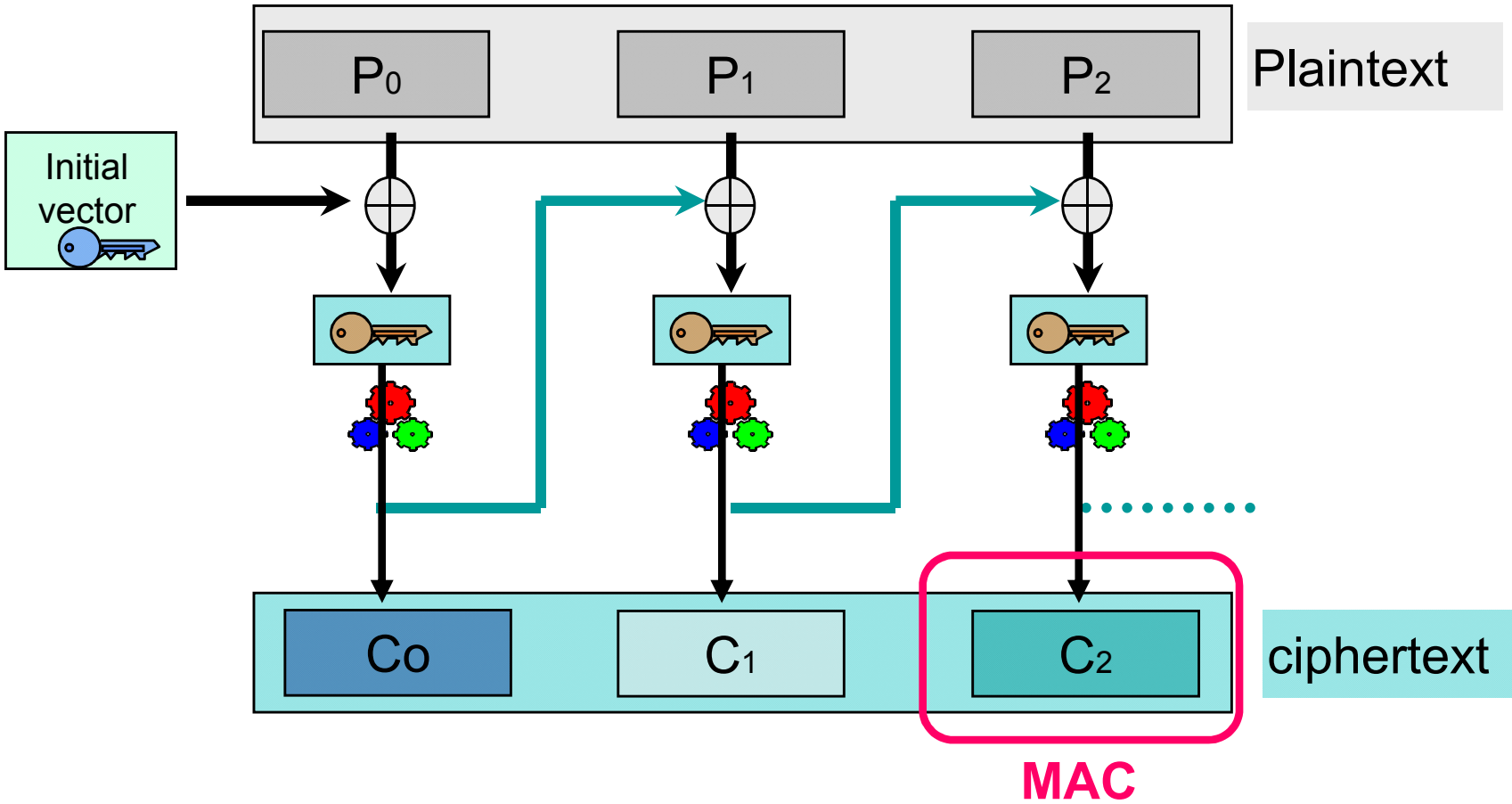


MAC: Message Authentication Code → Übersicht

- Ein **Message Authentication Code** oder MAC ist eine Einweg-**Hashfunktion**, die einen **Schlüssel** enthält, mit nur diesem man den Hashwert verifizieren kann.
- Damit kann man Authentizität ohne Geheimhaltung erreichen.
- Mit Hilfe von MACs können mehrere Benutzer ihre Dateien authentifizieren und einzelne Benutzer können mit MACs überprüfen, ob ihre Dateien verändert wurden, z.B. von Viren.
- Im Gegensatz zu Einweg-Hashfunktionen ist der Schlüssel des MACs zur Berechnung des Hashwertes nur dem Benutzer bekannt und der Hashwert kann nicht unbemerkt verändert werden.
- Eine einfache Möglichkeit zur Umwandlung einer Einweg-Hashfunktion in einen MAC besteht darin, den Hashwert mit einem symmetrischen Algorithmus zu verschlüsseln.
- Jeder MAC kann in eine Einweg-Hashfunktion umgewandelt werden, indem man den Schlüssel veröffentlicht.

MAC: Message Authentication Code → CBC MAC

- CBC MAC, weit verbreiteter Standard [NIS85].
- Die Idee ist die Verwendung von DES (oder auch andere wie AES) im CBC Modus und die anschließende Verwendung des letzten Blocks des Ciphertextes als Prüfsumme.
- Diese Verfahren wird oft in der **Bankenwelt** verwendet.



MAC: Message Authentication Code
→ Performance von kryptographische Verfahren

Typ	Algorithms	MByte/sec @ P90MHz
Symmetric algorithms	DES	0,4 (1,9 in HW)
	Triple-DES	0,16 (0,7 in HW)
	IDEA	1,0
	RC5 (64 Bits)	2,7
	Blowfish-16	3,0
Public Key	RSA (512 Bits)	0,001
Hash	MD5	14,2
	SHA-1	6,1
	RIPEMD-160	5,0

Mögliche Verfahren für die MAC-Berechnung

Hash-Algorithmen sind schneller als Verschlüsselungs-Algorithmen!

MAC: Message Authentication Code
→ HMAC: Übersicht

- **HMAC (Keyed-Hashing for Message Authentication)**
  Randbedingungen: (Internet-Standard (RFC 2104), z.B. IPSec)
  - **Die Geschwindigkeit der Hashfunktion soll nur wesentlich verlangsamt werden!**
  - Das Verfahren soll mit möglichst vielen Hashfunktionen zusammenarbeiten, ohne dass diese modifiziert werden müssen.
  - Die Sicherheit der Hashfunktion darf durch die Manipulation mit geheimen Schlüsseln nicht verringert werden.

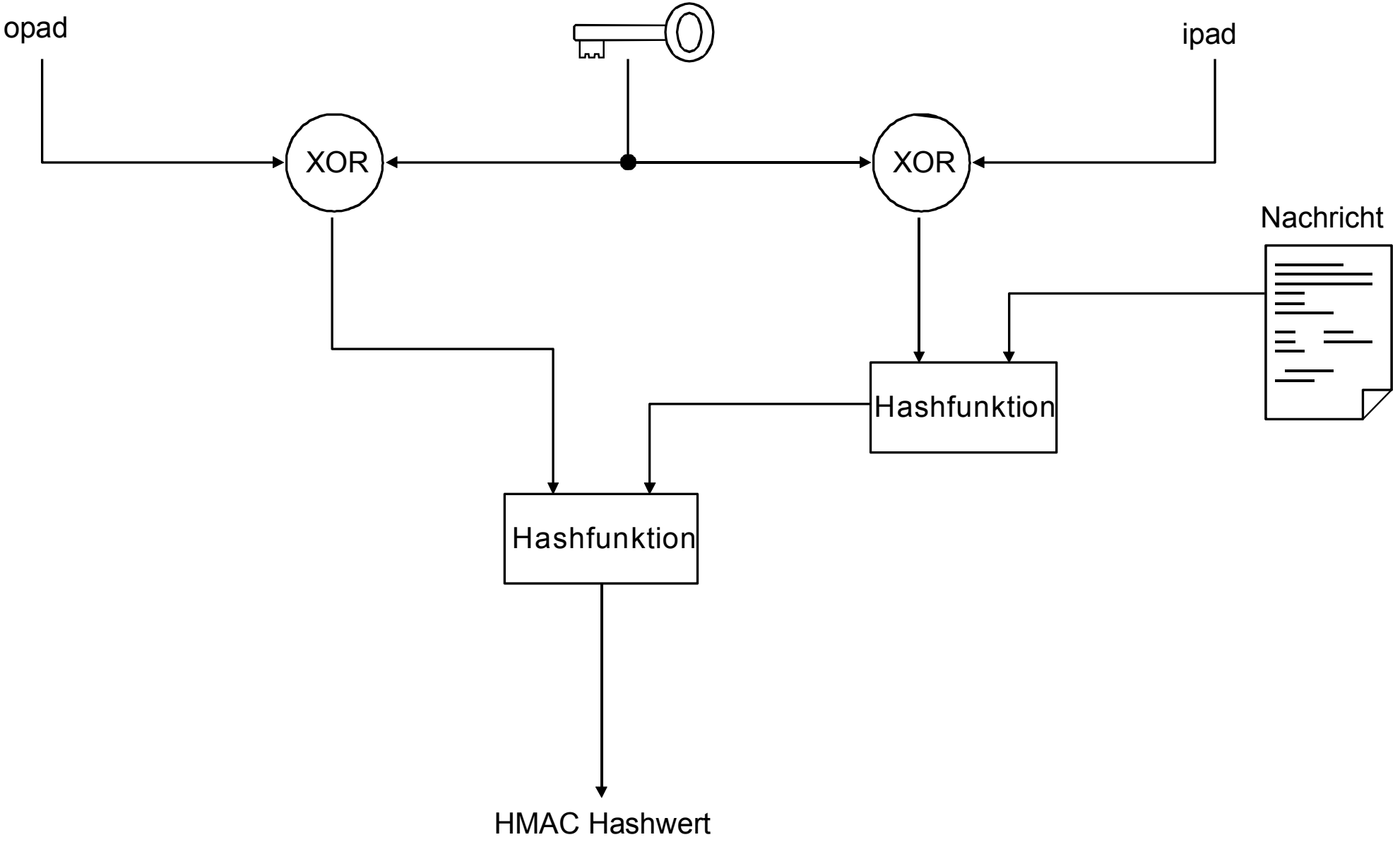
- **HMAC-Verfahren:**
  - HMAC = **H(** K **XOR** opad, **H(** K **XOR** ipad, M**)** **)**

    ipad = 0x36, 0x36, 0x36, ... (gleiche Länge wie die Blocklänge der Hashfunktion)
    opad = 0x5c, 0x5c, 0x5c, ... (gleiche Länge wie die Blocklänge der Hashfunktion)
    K = geheimer Schlüssel
    M = Input (Nachricht)
    XOR = bitweise modulo 2 addieren
    H = One-Way Hashfunktion (SHA-1, RIPEMD, ...)

MAC: Message Authentication Code
→ HMAC: Detailbeschreibung des Verfahrens

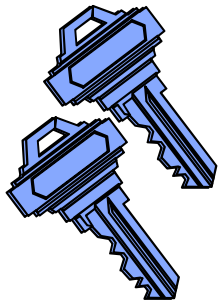
- Die Felder ipad und opad haben eine Länge, die der Blockgröße B der eingesetzten Hashfunktion entspricht (64 Bytes bei SHA-1 und RIPEMD).
- Der Schlüssel K wird durch das Anhängen von Nullen ebenfalls auf die Länge B gebracht.
- Verknüpfe den auf die Länge B gebrachten geheimen Schlüssel K mittels XOR mit dem Feld ipad.
- Stelle das Ergebnis dieser Operation vor die Nachricht und berechne mit der Hashfunktion den Hashwert aus diesem Input.
- Der Hashwert hat die Länge L (16 Byte bei SHA-1 und RIPEMD).
- Verknüpfe den auf die Länge B gebrachten geheimen Schlüssel K mittels XOR mit dem Feld opad.
- Stelle das Ergebnis dieser Operation (Länge B) vor den Hashwert (Länge L) und berechne mit der Hashfunktion den HMAC-Hashwert.
- Der HMAC-Hashwert hat die Länge L (16 Byte bei SHA-1 und RIPEMD).

MAC: Message Authentication Code
→ HMAC: Detailbeschreibung des Verfahrens

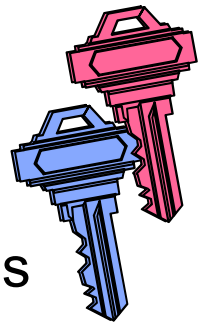


Verschlüsselungs Methoden
→ Probleme

- **Secret Key (symmetrische Verfahren)**
  - Große Anzahl von Schlüsseln, im Falle von vielen Kommunikationspartnern (N*(N-1)/2)
  - Schlüsselaustausch über gesicherte Kanäle

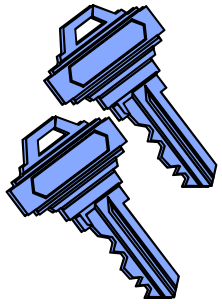


- **Public Key (asymmetrische Verfahren)**
  - Performance ist für reine Verschlüsselung unakzeptabel (RSA: 1000x langsamer gegenüber AES)
  - Es wird eine Public-Key-Infrastruktur benötigt, was in der Praxis oft sehr komplex ist.

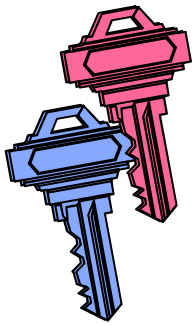


Praktische Anwendung
→ Nutzung mehrerer Methoden

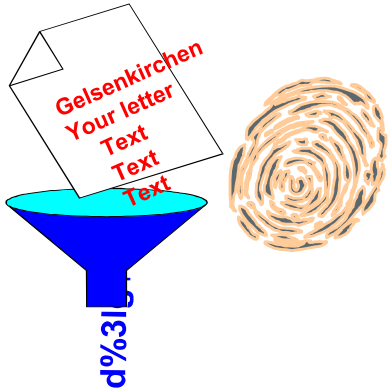
- **Private Key Algorithmen**
  - Verschlüsselung der Daten



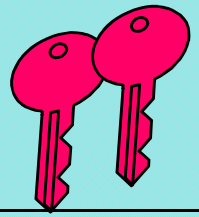
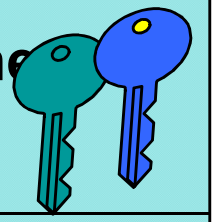
- **Public Key Algorithmen**
  - Gesicherter Schlüsselaustausch von geheimen Schlüsseln
  - elektronische Signaturen
  - Authentikation



- **One-Way Hashfunktionen**
  - Daten Integrität (im Rahmen der digitalen Signatur)



Algorithmen
→ Vorteile/Nachteile

	Symmetrische Algorithmen (z.B. AES) 	Asymmetrische Algorithmen (z.B. RSA) 
Vorteile	Einfache Schlüsselerzeugung Gute Performance	Kein gesicherter Schlüssel-austausch notwendig Nur ein Schlüsselpaar
Nachteile	Gesicherter Schlüsselaus-tausch notwendig Ein Schlüsselpaar für jeden Partner notwendig	Schlechte Performance Komplexe Schlüssel-generierung
Anwendung	→ Vertraulichkeit (Ver-/ Entschlüsselung von Daten) → Integrität (DES-CBC MAC)	→ Schlüssel Management → Digitale Signatur → Authentikation → Integrität

Inhalt

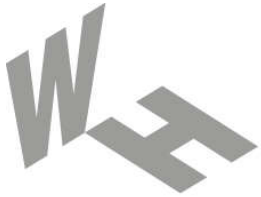
- Ziele
- Einführung
- Grundlagen der Verschlüsselung
- Elementarverschlüsselungen
- Symmetrische oder Private-Key Verschlüsselungsverfahren
- Asymmetrische oder Public-Key Verschlüsselungsverfahren
- One-Way-Hashfunktionen
- **Zusammenfassung**

Kryptographische Verfahren
→ Zusammenfassung

- Kryptographische Verfahren sind die **Basis** der meisten **Sicherheitssysteme**.
- Die **Sicherheit** eines kryptographischen Systems
  - hängt niemals von der Geheimhaltung der Algorithmen ab
  - **basiert ausschließlich auf der Geheimhaltung des privaten Schlüssels**
- Der jeweilige Algorithmus sollte anhand der folgenden Kriterien ausgewählt werden:
  - Die Kosten, um den Algorithmus zu brechen, sollten höher als die damit geschützten Informationen sein.
  - Der zeitliche Aufwand, um den Algorithmus zu knacken, sollte länger als das Interesse an den Informationen sein.
- Es sollten die Empfehlungen der Experten beachtet werden (in D z.B. BSI u. Bundesnetzagentur bezüglich des Signaturgesetztes).

Kryptographische Verfahren
→ Web-Seiten

- www.bsi.de
- www.bundesnetzagentur.de
- www.cryptool.de



Westfälische Hochschule
Gelsenkirchen Bocholt Recklinghausen
University of Applied Sciences

Kryptographische Verfahren

Vielen Dank für Ihre Aufmerksamkeit
Fragen ?

Prof. Dr. (TU NN)
Norbert Pohlmann

Institut für Internet-Sicherheit – if(is)
Westfälische Hochschule, Gelsenkirchen
http://www.internet-sicherheit.de

Lösungshilfe

Buchstabe	Häufigkeit (in %)	Buchstabe	Häufigkeit (in %)
A	6,51	N	9,78
B	1,89	O	2,51
C	3,06	P	0,79
D	5,08	Q	0,02
E	17,40	R	7,00
F	1,66	S	7,27
G	3,01	T	6,15
H	4,76	U	4,35
I	7,55	V	0,67
J	0,27	W	1,89
K	1,21	X	0,03
L	3,44	Y	0,04
M	2,53	Z	1,13

Häufigkeit der Buchstaben der deutschen Sprache

Buchstabenpaar	Häufigkeit (in %)
en	3,88
er	3,75
ch	2,75
te	2,26
de	2,00
nd	1,99
ei	1,88
ie	1,79
in	1,67
es	1,52

Häufigkeit der Buchstabenpaare