



---

# Vorlesung

## VPN: Drahtgebunden und drahtlos

Fachbereich Informatik (FB 20)

Lehrstuhl Prof. J. Buchmann

WS-05 / V2 - 20.205.1

In Zusammenarbeit mit dem CAST-Forum

Dr. Wolfgang Böhmer

Skript: <http://www.cdc.informatik.tu-darmstadt.de/~wboehmer/>

Email: [wboehmer@cdc.informatik.tu-darmstadt.de](mailto:wboehmer@cdc.informatik.tu-darmstadt.de)

---





# Vorlesungsinhalte

---

- Verschlüsselung
  - Verschlüsselungstechniken
    - Substitution und Transformation
  - Symmetrische Kryptosysteme
    - Blockchiffre und Stromchiffre
    - DES, Triple DES, IDEA und AES
    - DES-Cracker
  - Asymmetrische Kryptosysteme
    - Modul Arithmetik
  - Schlüsselaustauschverfahren
    - Diffie-Hellmann,
  - Hashfunktionen
    - kryptographisch sichere Funktionen
    - MD5 und SHA1





# Verschlüsselungstechniken

---

- Vertrauliche Nachrichtenübermittlung - ein altes Problem der Menschheit
- Substitutionstechniken (*Tausch- oder Ersatzverfahren*)
- Cäsars-Verschlüsselung
  - Buchstaben-Ersatzverfahren an der Stelle 3 des Alphabets

a b c d e f g h i j ..... t u v w x y z  
D E F G H I J K L M ..... W X Y Z A B C

Klartext: treffe mich nach der Toga Party

Chiffrat: WUHIIIH PLFK QDFK GHU WRJD SDUWB

Für jeden Buchstaben im Klartext ( $p$ ) kann durch Substitution ( $k$ ) das Chiffrat ( $C$ ) erzeugt werden

$$C = E(p) = (p + k) \bmod 26$$

Frage: Kann der Klartext aus dem Chiffrat reproduziert werden?





# Verschlüsselungstechniken

---

Chiffirat:            WUHIIH PLFK QDFK GHU WRJD SDUWB  
**Key 1:**            vtghhg okej pcej fgt vqic rcta  
2:            usfggf ujdi obdi efs uphb qbsz  
3:            treffe mich nach der Toga Party

- Kryptoanalyse (*Kunst des Brechens von Kryptosystemen*)
  - Der Ver- und Entschlüsselungsalgorithmus war bekannt
  - Es gibt aufgrund des Alphabets nur 25 Schlüssel die verwendet werden können
  - Sprache (Klartext) Deutsch und ist leicht erkennbar. (rel. Häufigkeit)
- Das Beispiel zeigt:
  - Chiffirat enthält jede mögliche Permutation von 26 Buchstaben  $26!$  (ca.  $4 \times 10^{26}$ )
  - DES-Algorithmus mit einem 56 Bit Schlüssel und einem Schlüsselraum von  $2^{56}$  entspricht  $7 \times 10^{16}$





# Verschlüsselungstechniken

---

- Relative Häufigkeit (Morse-Alphabet)

(e -> .) (t -> -)

- Playfair-Verschlüsselung (Standard Algorithmus im Ersten Weltkrieg)
- Schlüsselwort z.B.: **Wildfang**
- Chiffrierregeln des Playfair Algorithmus:
  - *Jeder Buchstabe eine Buchstabenkombination im Klartext wird durch den Buchstaben ersetzt, der in der gleichen Reihe steht, jedoch die Position der Spalte des zweiten Buchstaben einnimmt. (Transformation ist Reihen konform). Z.B. (ed) zu (MW) und ns zu (BQ).*

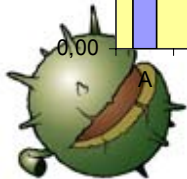
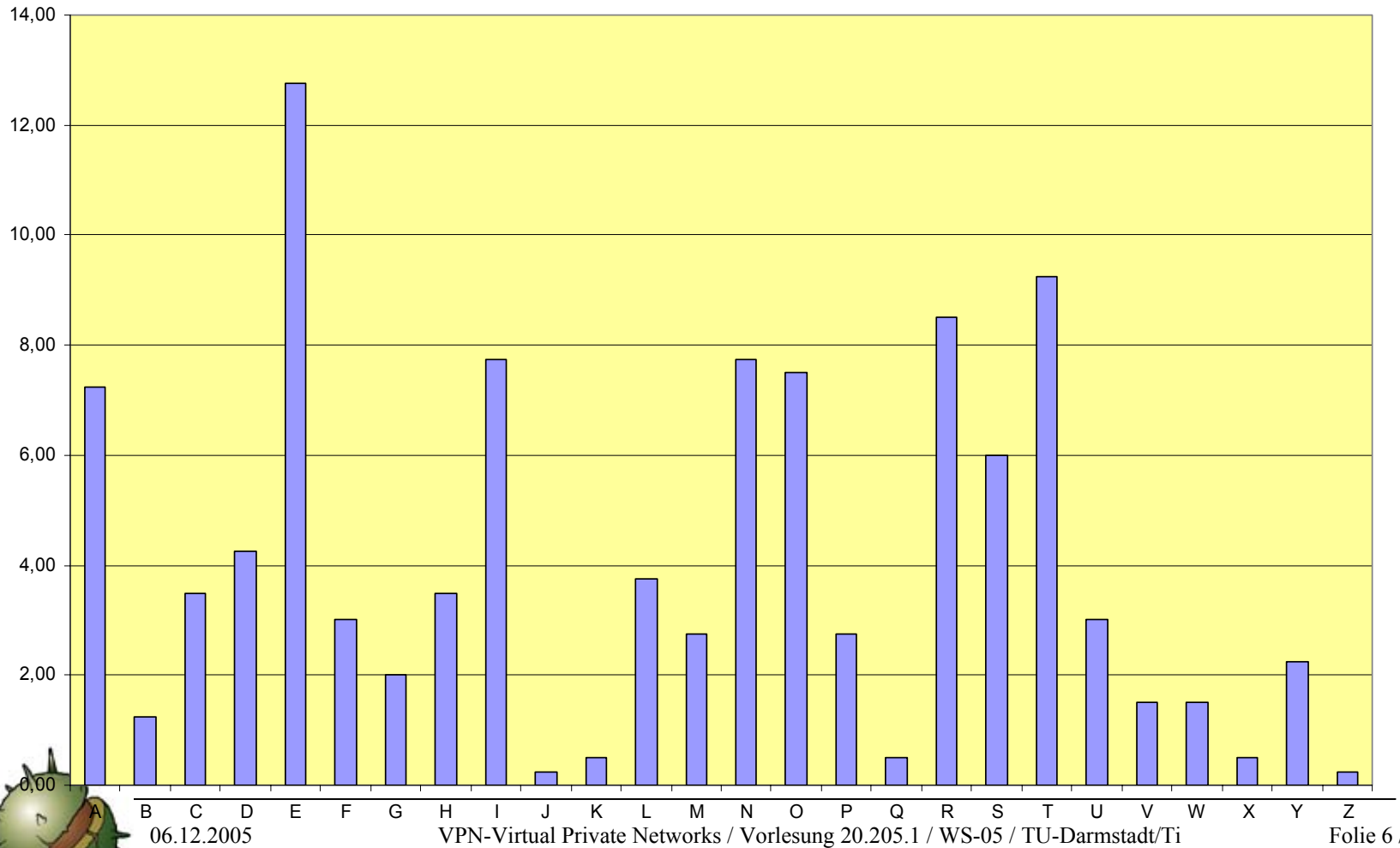
<b>W</b>	<b>IJ</b>	<b>L</b>	<b>D</b>	<b>F</b>
<b>A</b>	<b>N</b>	<b>G</b>	B	C
E	H	K	M	O
P	Q	R	S	T
U	V	X	Y	Z



# Qualität von Verschlüsselungen

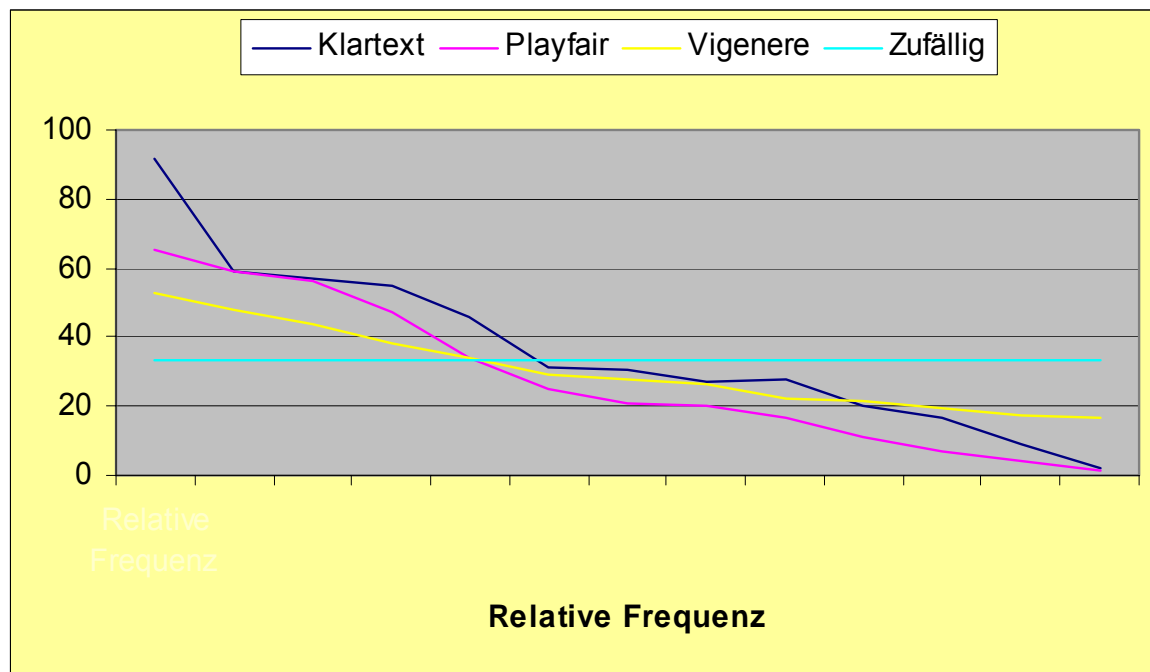
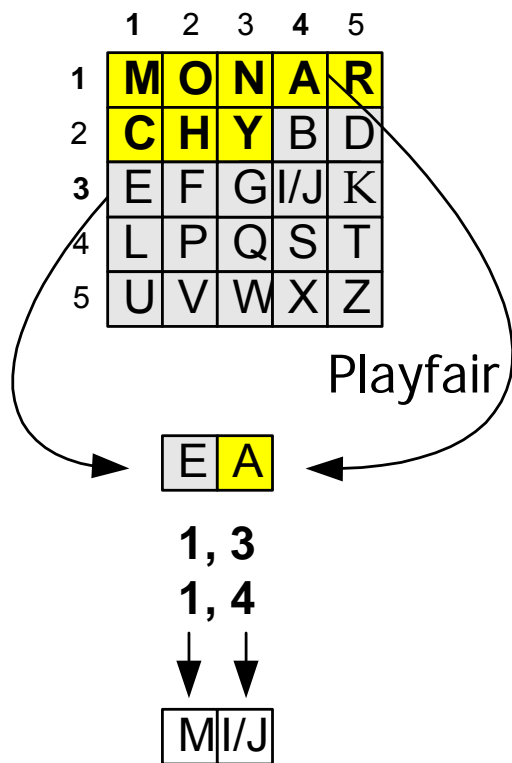


- Anatomie der Sprache: *Frequenzanalyse*





# Qualität von Verschlüsselungen





# Verschlüsselungstechniken

---

- Transpositionstechniken (*Permutation nach Mustern*)
  - Zeichen bleiben erhalten, jedoch die relative Position der Buchstaben ändert sich.  
Z.B. Zickzack-Muster mit der Tiefe von zwei

treffe mich nach der toga party

t r f m c n d r o a a t  
e f e i h a h e t g p r y

TRFMCNCDROAATEFEIHAHETGPRY

- **Dies Beispiel ist für die Kryptoanalyse trivial zu dechiffrieren**





# Verschlüsselungstechniken

---

- Transpositionstechniken (*Permutationsordnung mittels Matrix*)

Schlüssel: 4 3 1 2 5 6 7

Klartext: t r e f f e m

i c h n a c h

d e r t o g a

p a r t y y z

Schlüssel: 4 3 1 2 5 6 7

Klartext: e h r r f n t

t r c e a t i

d p f a o y e

c g y m h a z

Chiffre: E H R R F N T T R C E A T I D P F A O Y E C G Y M H A Z





# Verschlüsselungstechniken

---

- Doppelte Transposition führt zu komplexeren Permutationen
- Doppelte Substitution führt zu komplexeren Tauschverfahren
- Optimale Umsetzung beider Verfahren durch Rotor-Maschinen

- Paradigmawechsel durch Kerckhoff:
  - **Sicherheit eines Kryptosystems darf nicht von der Geheimhaltung des Algorithmus abhängen, sondern ausschließlich auf die Geheimhaltung des Schlüssels beruhen.**

- Claude Shannon leitete die moderne Kryptographie ein (1940)
  - Diffusion und Konfusion
  - Erste praktische Umsetzung Feistel-Netzwerk (DES) 1973





# Symmetrische Kryptosysteme

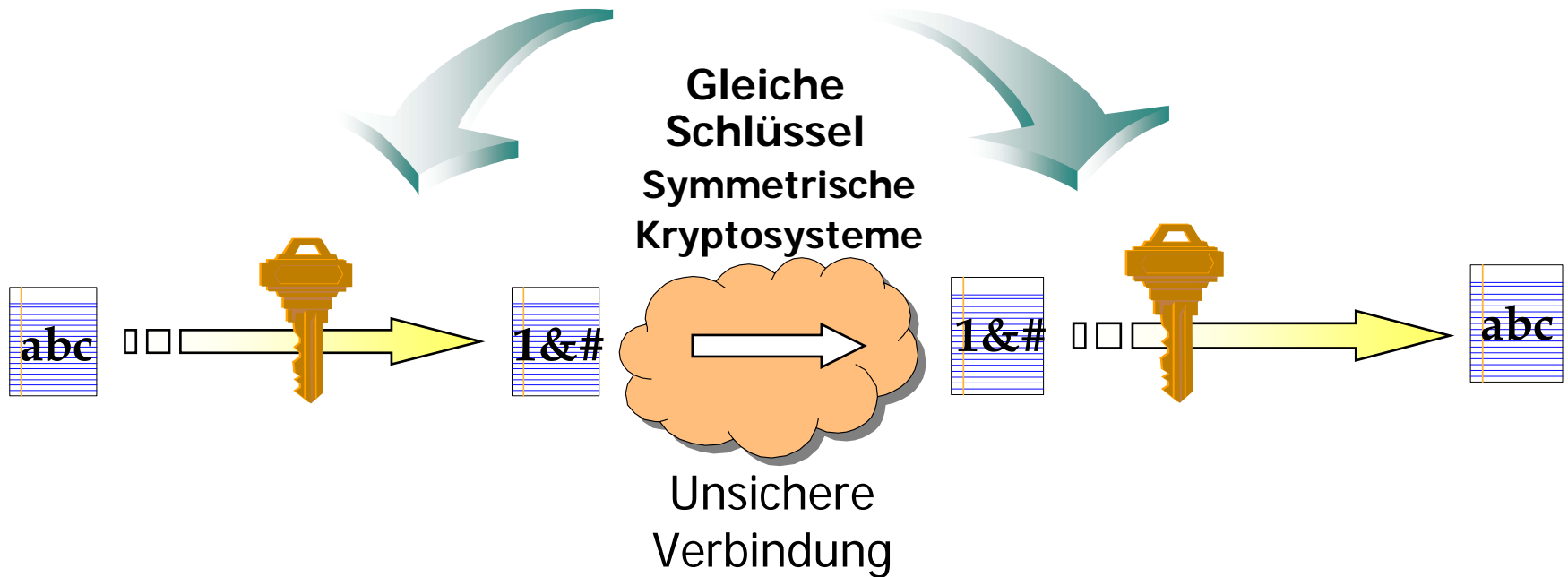
---

- Für Chiffrierung und Dechiffrierung wird derselbe Schlüssel benutzt.
- Relative hoher Durchsatz beim Verschlüsselungsvorgang im Vergleich zu asymmetrischen Verfahren
- Anwendungsgebiete:
  - Datenspeicherung und Sicherung vor unbefugten Dritten
  - Authentifizierungsverfahren
  - VPN
  - WLAN
- Vertreter:
  - DES, Triple DES, AES, IDEA und Blowfish, etc.





# Symmetrische Verschlüsselung (*Private Key-Verfahren*)



Schwierigkeit der Schlüsselübertragung bleibt nach wie vor !

Shannon's Fassung: „*The enemy knows the system beeing used*“





# Symmetrische Kryptosysteme

---

- ***Stromchiffre (stream ciphers)***
  - ein kontinuierlicher Bitstrom (Klartext) wird bit- oder byteweise chiffriert.
  - Beispiel: RC4 als Verfahren im IEEE802.11b (WEP)
  
- ***Blockchiffere (block ciphers)***
  - Klartext wird generell in Bit-Blöcken fester Längen (64-Bit) verarbeitet.
  - Modes of Operation z.B. für DES
  - Einzelne Bit-Blöcke können alleine oder in Verbindung mit Vorgängern oder Nachfolgern verknüpft werden.
  - Es existieren vier bedeutende Verfahren (ECB, CBC, CFB, OFB)
  - Von selbst entwickelten Verfahren ist Abstand zu nehmen (PCBC und SAP)

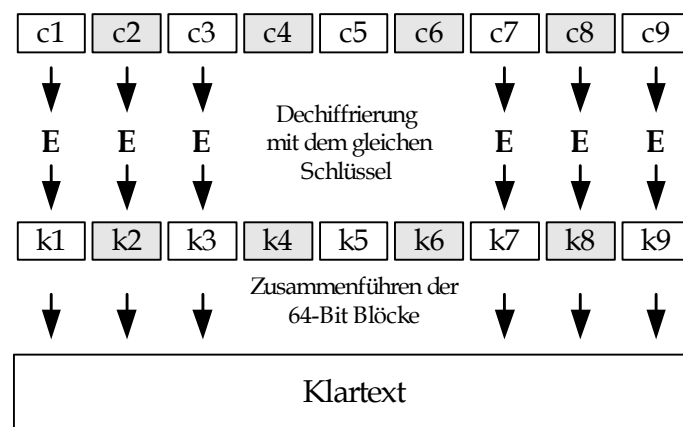
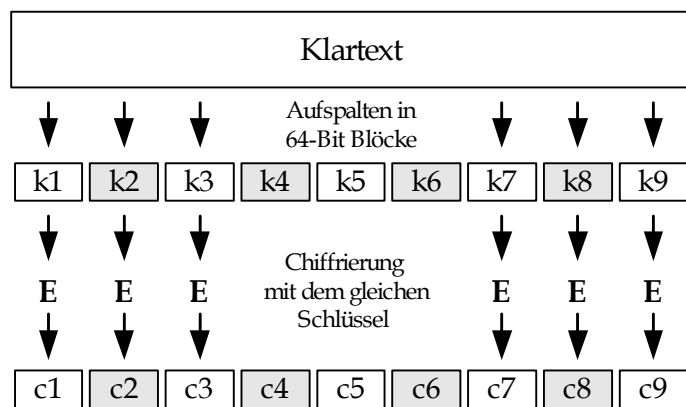




# Blockchiffere (*ECB*)

## *Elektronisches Codebuch*

- Klartext wird in n-Bit-Blöcke zerlegt
- Der letzte Block wird bei Bedarf aufgefüllt (Padding)
- Jeder Block wird separat mit dem gleichen Schlüssel verschlüsselt
- Identische Klartexte erzeugen identische Chifferte.

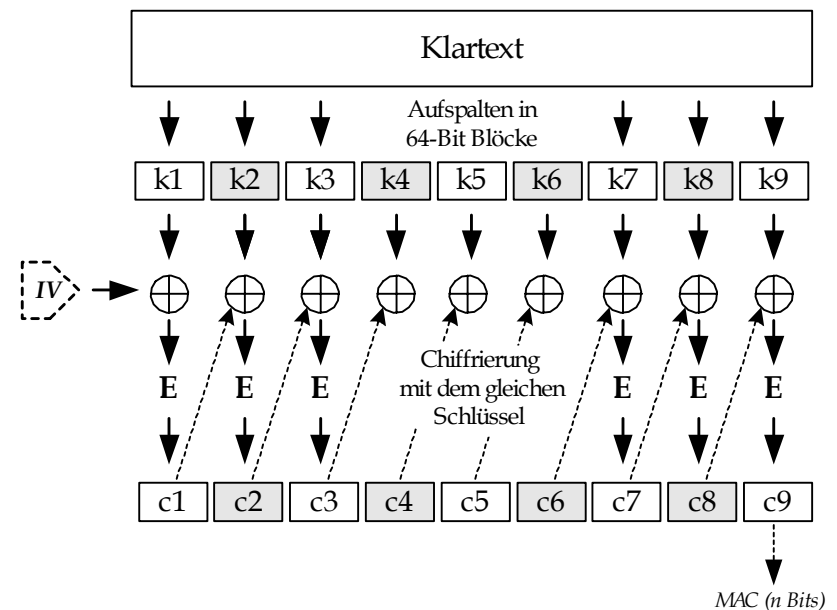


# Blockchiffere (CBC)

## Blockverkettung



- Klartext wird in n-Bit-Blöcke zerlegt
- Blöcke werden jedoch untereinander verkettet (*Integration*)
- Ein Initialisierungsvektor (IV) ist notwendig

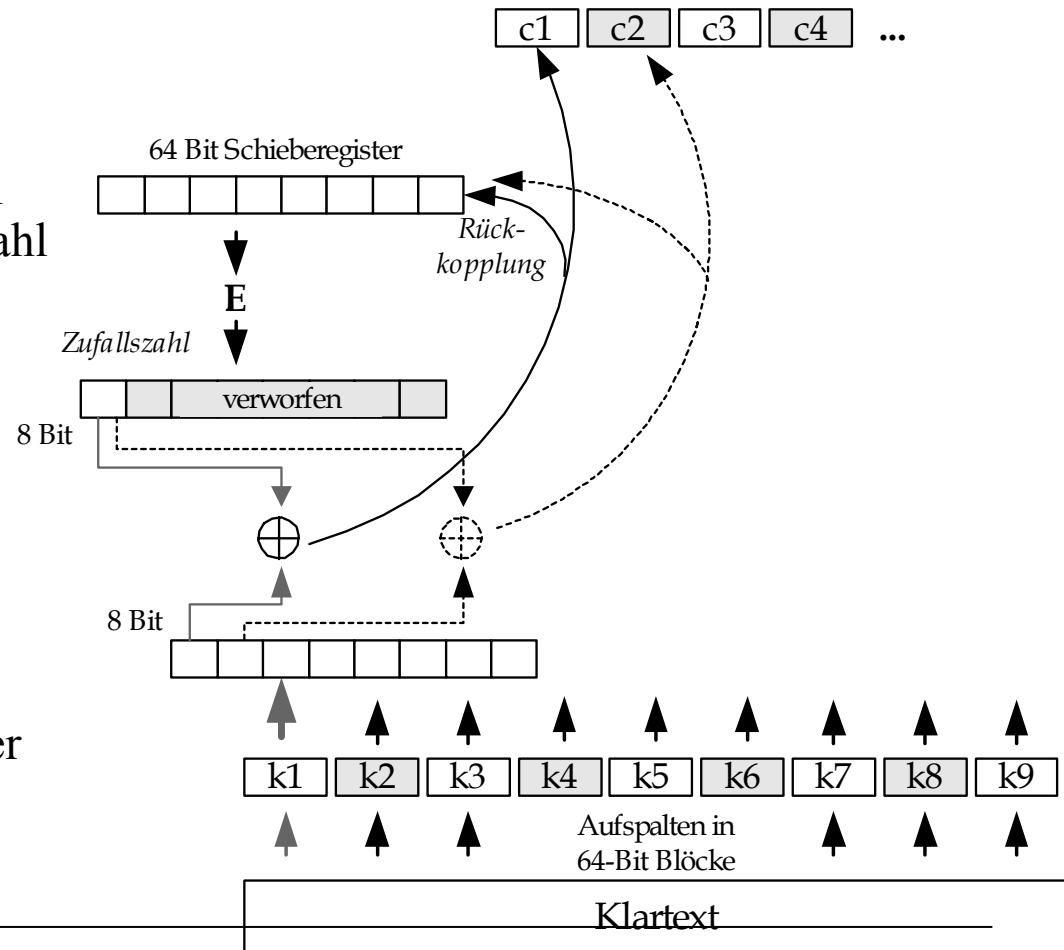


# Blockchiffere (CFB)

## Schlüsseltextrückführung



- CFB bietet Möglichkeit Nachrichten zu verarbeiten die kleiner als die Bit-Anzahl der Blocklänge ist
- Es werden Pseudo-zufallszahlen durch ein Schieberegister erzeugt
- CFB kann für Echtzeitbedingungen eingesetzt werden (Telnet)
- n-Bit CFB ist selbst-synchronisierend nach einer Anzahl n Schritte

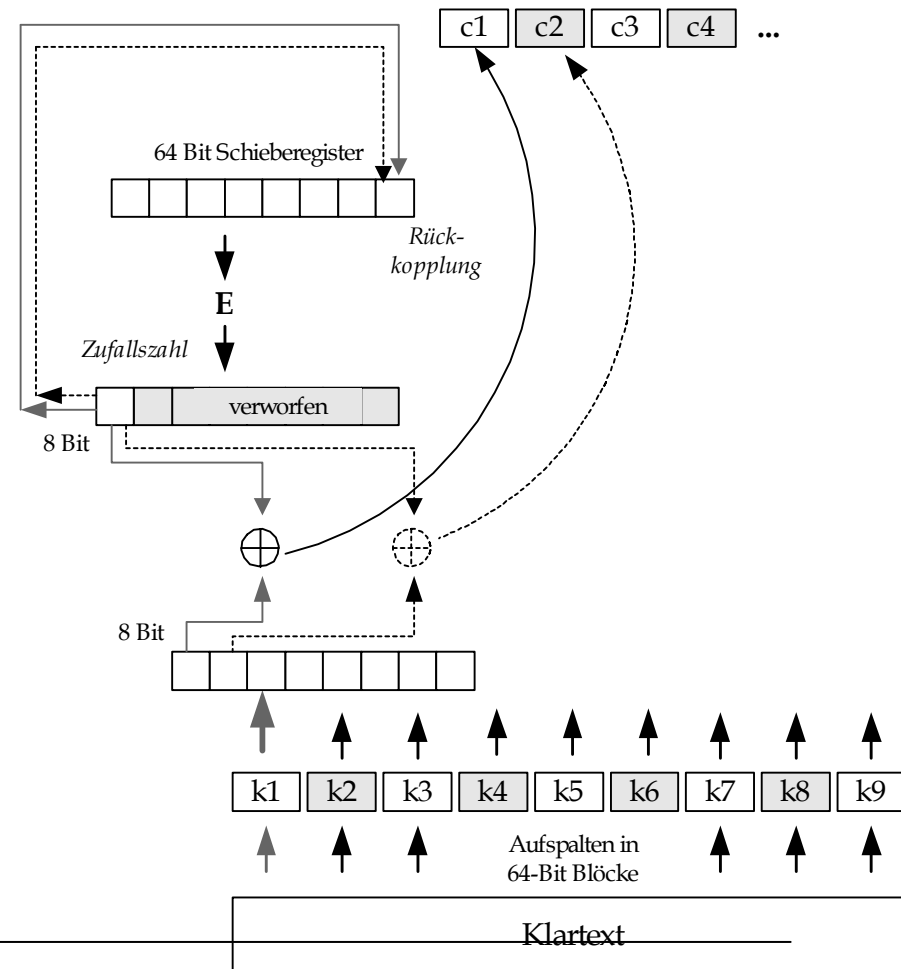


# Blockchiffere (*OFB*)

## *Ergebnisrückführung*



- Starke Ähnlichkeit mit CFB  
Unterschied liegt in der Rückkopplung
- Fehlerfortpflanzung wird vermieden, da keine Wiederverwendung des Chiffrats erfolgt
- Fehler im Schieberegister führt dazu, dass nachfolgender Text unbrauchbar wird
- Synchronisationsfehler sind nicht behebbar

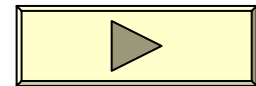




# Standard-Verfahren DES

---

- Geht auf Horst Feistel (1973) zurück
- Es wendet konsequent die Idee von Shannon an (Diffusion/Konfusion)
- Es folgt dem Design:
  - Es verwendet Produktchiffieren, die zwei oder mehr grundlegende Substitutionen und/oder Permutationen in Folge durchführt
  - Es werden Substitutionsrunden mit einer Substitutionsfunktionen durchgeführt
  - Es verwendet einen schlüsselbasierten Algorithmus, um Schlüsselbits in Subkeys zu verwandeln







CRYPTOGRAPHY RESEARCH, INC.

# DES Key Search Project

## Cracking DES

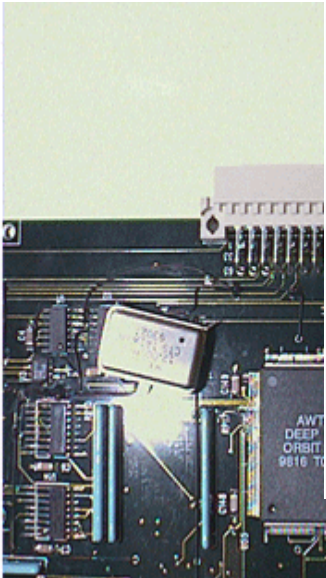
**Secrets of Encryption Research,  
Wiretap Politics & Chip Design**  
How federal agencies subvert privacy



### Cracking DES

Secrets of  
Encryption Research,  
Wiretap Politics  
& Chip Design

EFF ELECTRONIC FRONTIER FOUNDATION



Quelle: <http://www.cryptography.com/des>



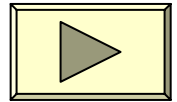
# Abkehr vom Feistel-Netzwerk

## Die Krypto-Olympiade 1997-2000

---



- Der öffentliche Evaluierungsprozess
- Die Anforderungen
  - Unterschiedliche Schlüssellängen unterstützen (128,192,256 Bit)
  - Sicher gegen Brut-Force Attacken für 20-30 Jahre
  - Performant auf unterschiedlichen Hardware Plattformen
  - Keine Patent- oder Exportbeschränkungen verletzen
  - wesentlich sicherer und mindestens so schnell wie DES sein





# DES-Nachfolger (*AES*)

---

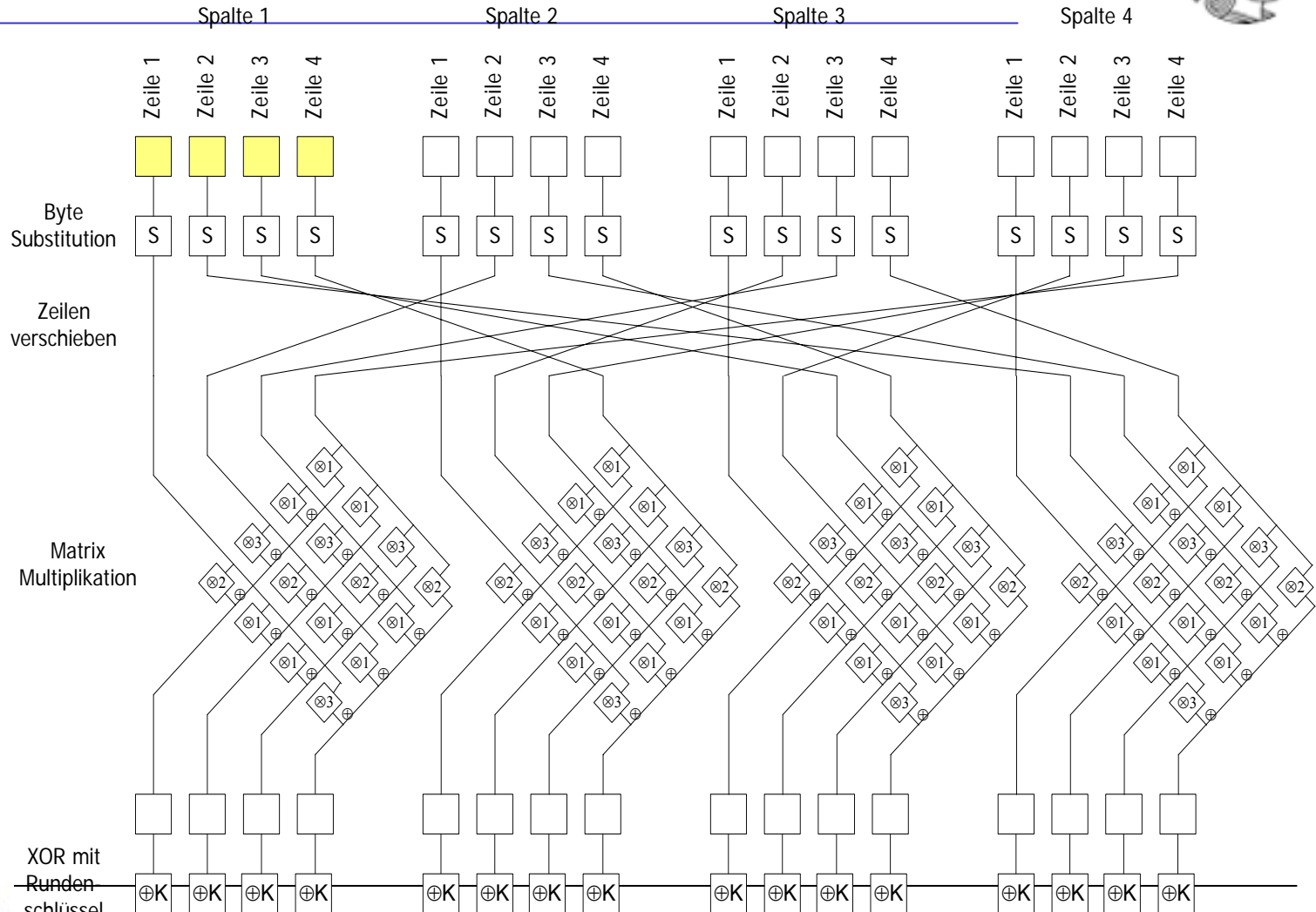
- Die Finalisten der zweiten und letzten Evaluierungsrunde
  - MARS (IBM),
  - RC6 (RSA Laboratories),
  - Serpent (Eli Biham, et. al.),
  - Twofisch (Bruce Schneier et. al.),
  - Rijndael (Daemen & Rijmen)

	Encryption und Decryption	Schlüsselverhalten
MARS	<b>II</b>	<b>II</b>
RC6	<b>I</b>	<b>II</b>
<b>Rijndael</b>	<b>I</b>	<b>I</b>
Serpent	<b>III</b>	<b>II</b>
Twofish	<b>II</b>	<b>III</b>



# Der Rijndael Algorithmus (1)

## Eine Runde des Rijndael Algorithmus mit 128 Bit Schlüssel





## Der Rijndael Algorithmus (2)

Schritt 0: Ordnen des Ausgangstexts in 4x4 Byte Matrix Blöcke

16 Bytes des Ausgangstextes

T	H	I	S	I	S	A	N	E	X	A	M	P	L	E	.	...
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-----

Ordnen als 4x4 Byte Matrix, die Spalten von oben nach unten auffüllend

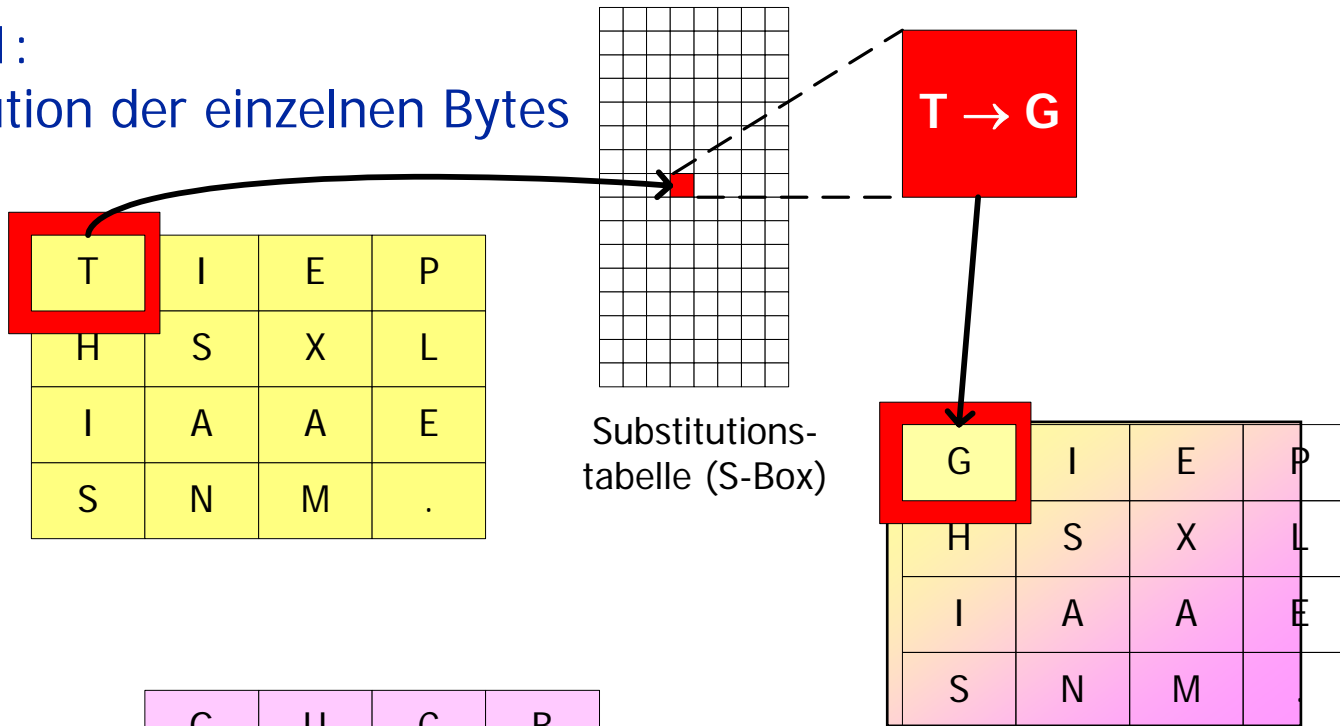
T	I	E	P
H	S	X	L
I	A	A	E
S	N	M	.





# Der Rijndael Algorithmus (3)

Schritt 1:  
Substitution der einzelnen Bytes



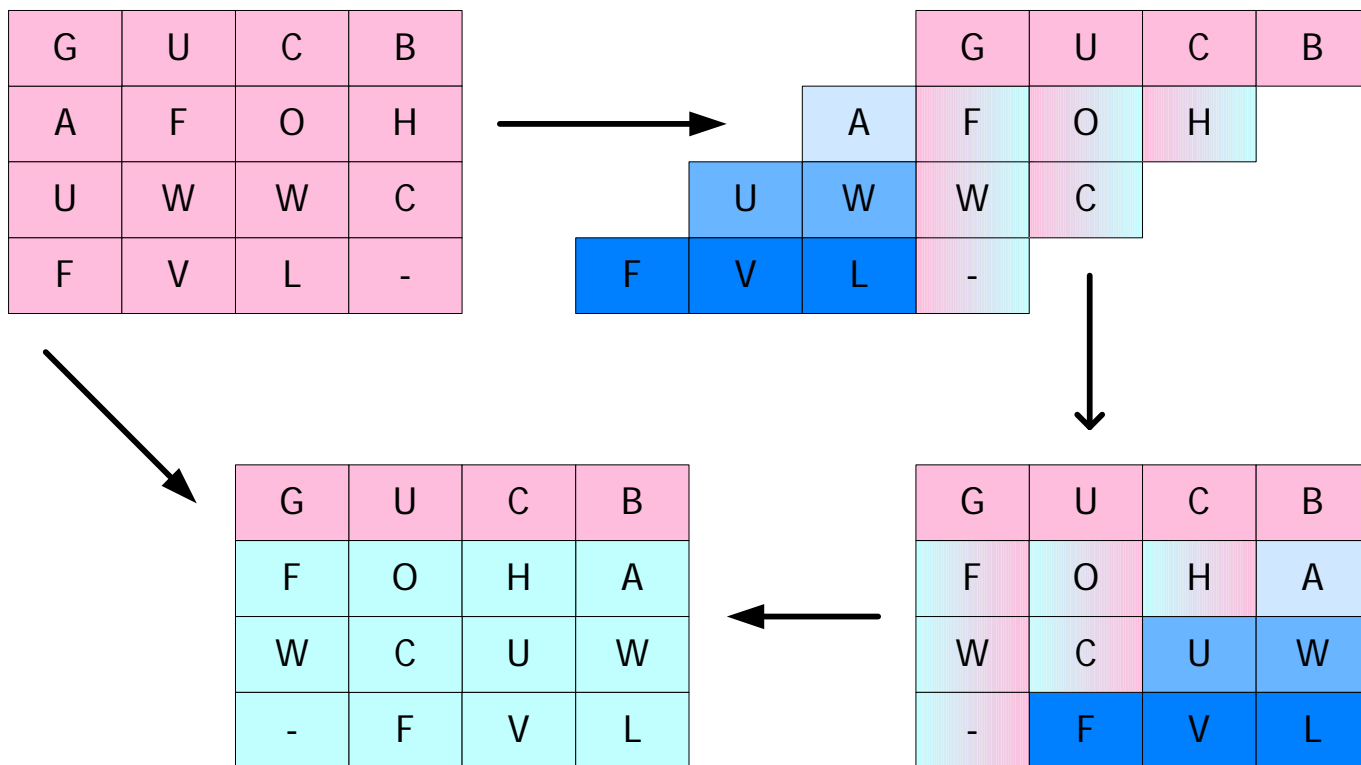
Und so weiter für alle Elemente  
der 4x4 Matrix





# Der Rijndael Algorithmus (4)

## Schritt 2: Shiften der Zeilen

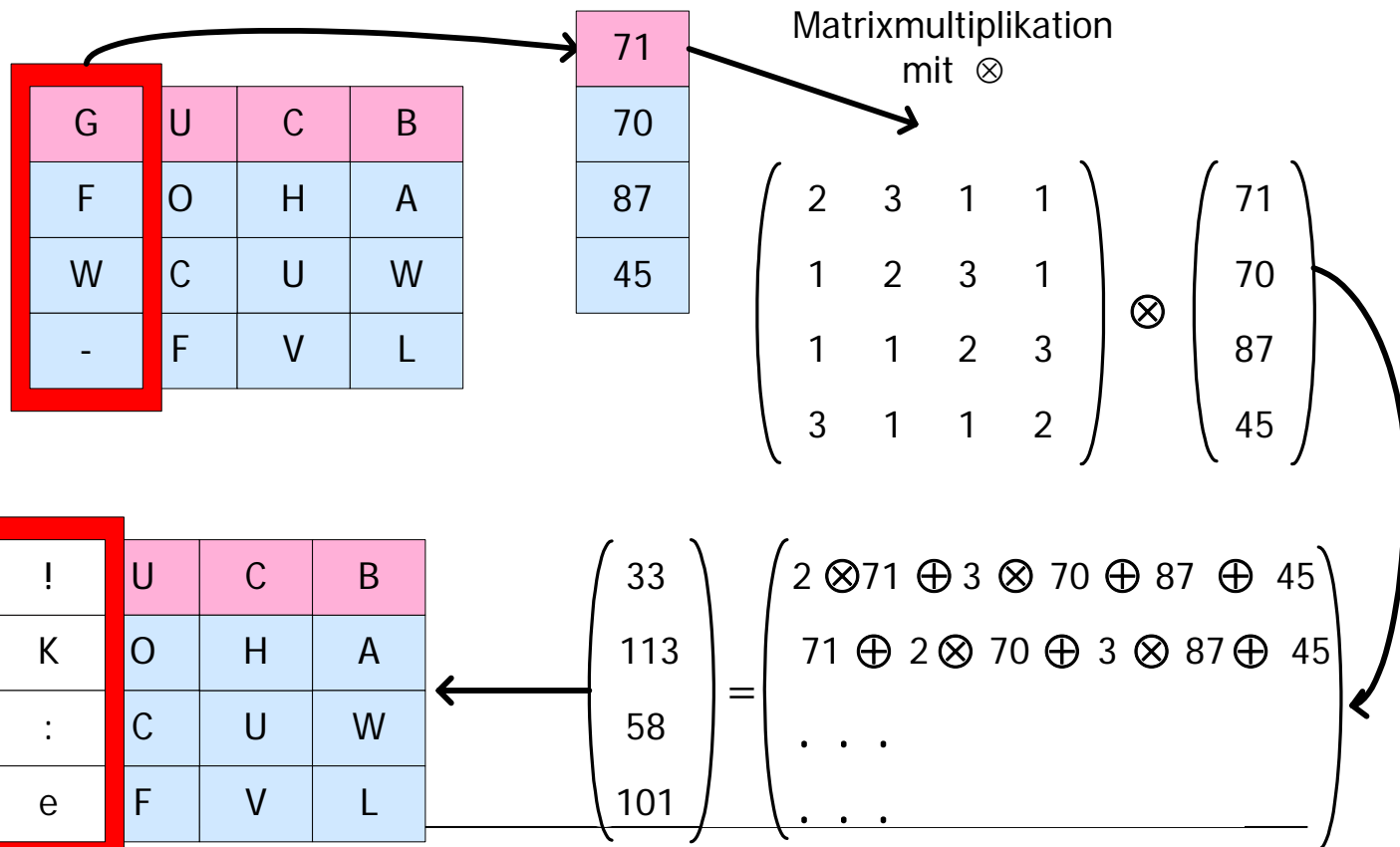




# Der Rijndael Algorithmus (5)

## Schritt 3: Mixen der Spalten

Ab hier muß spätestens mit dem ASCII Code weitergearbeitet werden



und analog mit den weiteren Spalten



# Der Rijndael Algorithmus (6)

## Schritt 4: Addition des Rundenschlüssels (128 Bit)

!	j	?	9
K	T	N	.
:	?	j	:
e	k	z	u

in eine Reihe schreiben (128 Bit)

!	j	?	9	K	T	N	.	...
---	---	---	---	---	---	---	---	-----

in binärer Form notieren

010111001000111011010010100101000111101101 . . .

⊕ mit Rundenschlüssel per XOR (⊕) verknüpfen

110101100101001011010100111111101101001100 . . .

---

100010101101110000000110011010101010100001 . . .

falls gewünscht, wieder in Textform (Chifftrat) schreiben



j	K	!	:	?	v	U	L	lie 28 / 39	...
---	---	---	---	---	---	---	---	-------------	-----



# Modulo-Arithmetik, das Rechnen in endlichen Zahlenbereichen

---

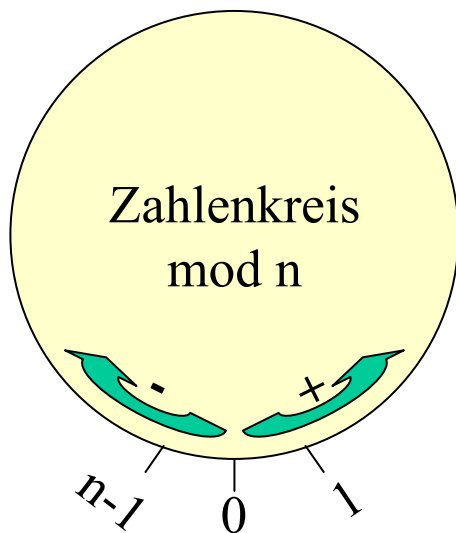
- Bekannt ist die Arithmetik in einem Zahlenbereich durchzuführen, der unendlich viele Zahlen enthält.
- Bei einer fortlaufenden Addition wird das Ergebnis stets größer, diese unbegrenzte Funktion lässt sich nur ungenügend auf Computer darstellen, da deren Zahlenraum begrenzt ist.
- Bei Computern wird, wenn eine obere Grenze der fortlaufenden Addition erreicht wird (oder gedanklich überschritten), beginnt die Summendarstellung wieder mit der kleinsten Zahl, meist mit der Zahl 0. Dieser Prozess kann sich wiederholen.
- Der Zahlenbereich in dem sowohl die Operation als auch die Ergebnisse enthalten sind, wird Zahlenmodul genannt, symbolisiert durch:
  - **$x$  modulo  $n$** , häufig abgekürzt zu  **$x \bmod n$**
- Das bedeutet das jede Zahl  $x$  aus dem Modul  $n$  nur einen der  $n$  möglichen Werte annehmen kann.





# distributive Modul-Arithmetik, das Rechnen in endlichen Zahlenbereichen

- **$x \bmod n$**
- Lineare Zahlenmenge  $M = \{0, 1, 2, \dots, n-1\}$ , so gilt  $x \in M$  oder  $0 \leq x < n$



Für eine modulo  $n$  – Addition gilt dann z.B:  
mit  $0 \leq a, b < n$

$$(a + b) \bmod n : \begin{cases} \text{falls } a + b < n; \text{ dann } a + b \\ \text{falls } a + b = n; \text{ dann } 0 \\ \text{Falls } a + b > n; \text{ dann } (a + b) - n \end{cases}$$

Die Modul-Arithmetik gelten für die Addition, die Subtraktion und die Multiplikation sowie die bekannten Rechenregeln, sie sind also jeweils assoziativ, kommutativ und zusammen distributiv



# Schlüsselaustauschverfahren (Diffie & Hellmann, 1976)



- **Es ist nicht nötig dem Kommunikationspartner auf einem sicheren Kanal bzw. Übertragungsweg den geheimen Schlüssel zu übermitteln**

- Zwei völlig unbekannte Personen oder auch Rechner (VPN-Gateways) können ohne vorherigen Austausch eines geheimen Schlüssels sofort vertraulich miteinander kommunizieren.

Eingangsfunktion

$$e = g^k \text{ mod } p$$

Umkehrfunktion

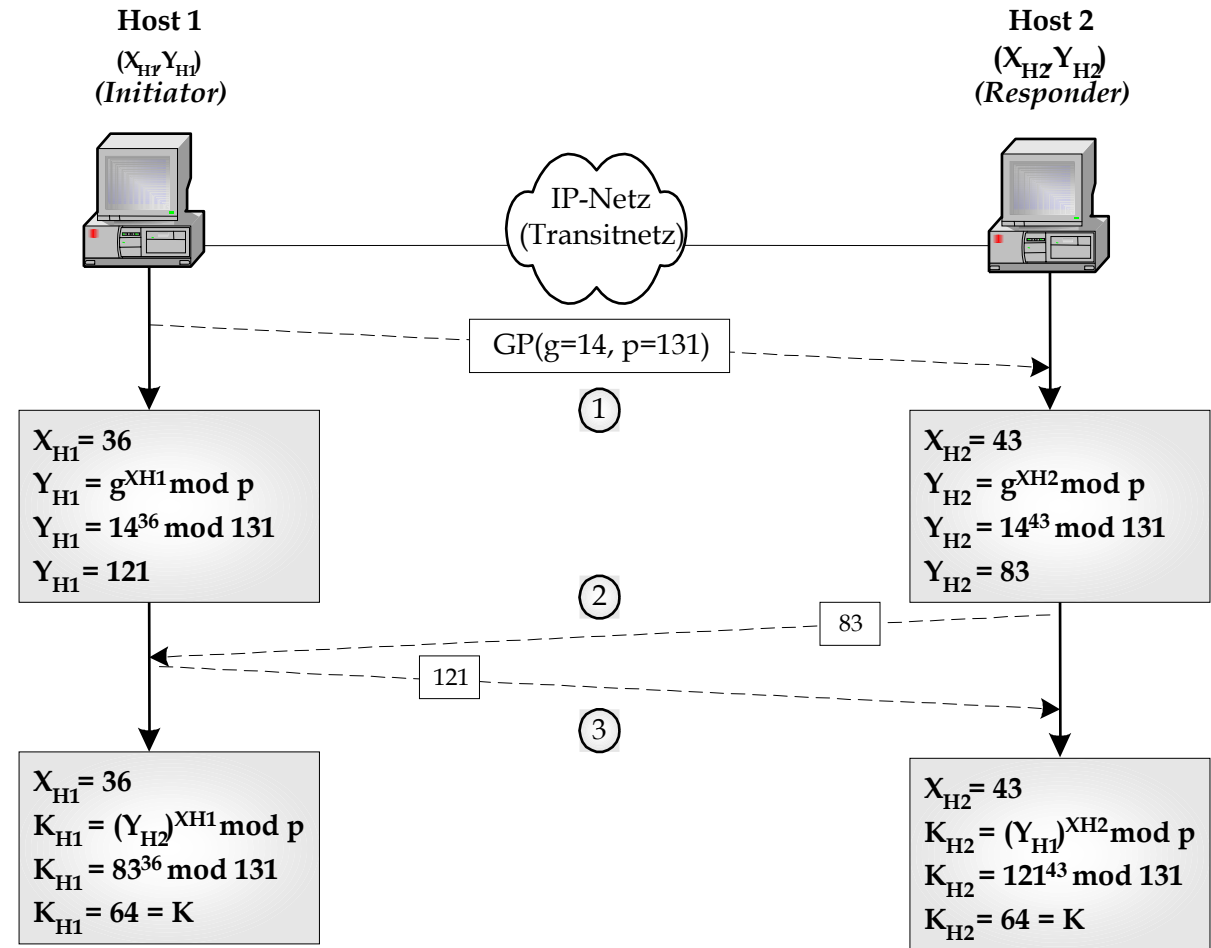
$$d_l_g (g^k) = e$$



# Diffie-Hellman Schlüsselaustausch



X = Geheimer DH-Wert  
Y = übertragener DH-Wert



# Diffie-Hellman

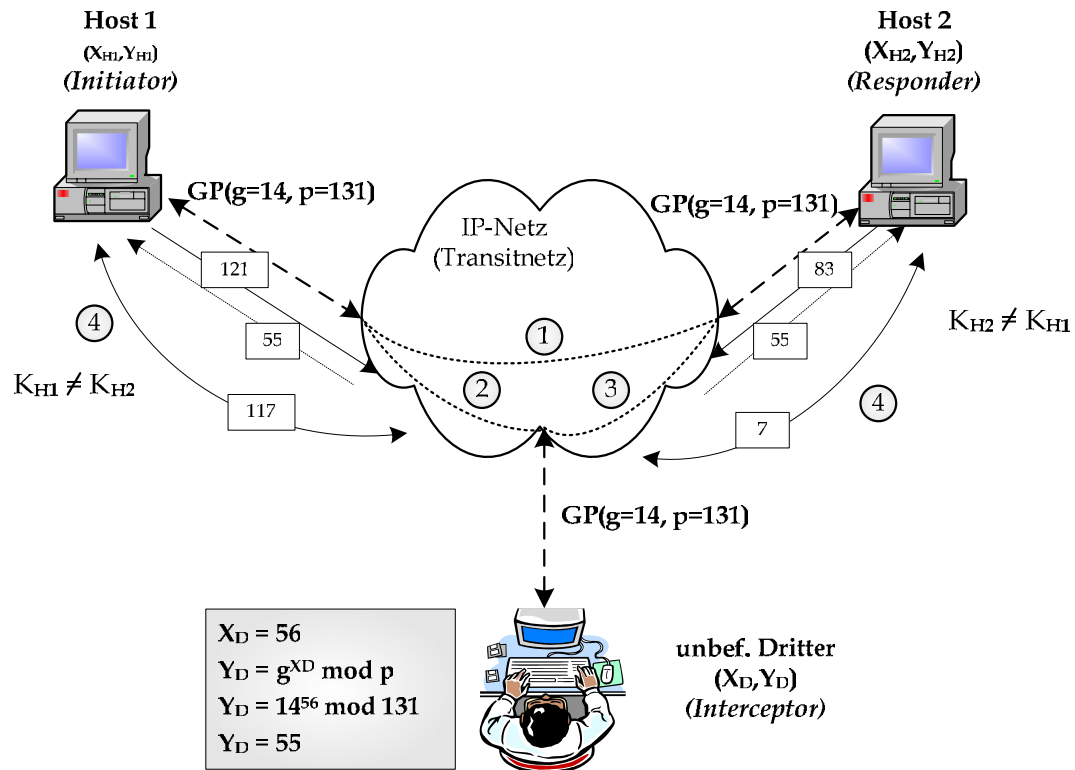
## *man-in-the-middle-attack*



- Ein unbefugter Dritter (D) nutzt das DH-Verfahren, um wechselseitig mit H1 und H2 eigene Credentials auszutauschen.
- GP = public
- D berechnet:  
 $K_{H2} = (Y_{H2}^{X_D}) \bmod 131$   
 $K_{H1} = (Y_{H1}^{X_D}) \bmod 131$
- Wie sieht  $K_{H2}$ ,  $K_{H1}$  bei H1 und H2 aus?

$X_{H1} = 36$   
 $Y_{H1} = g^{X_{H1}} \bmod p$   
 $Y_{H1} = 14^{36} \bmod 131$   
 $Y_{H1} = 121$

$X_{H2} = 43$   
 $Y_{H2} = g^{X_{H2}} \bmod p$   
 $Y_{H2} = 14^{43} \bmod 131$   
 $Y_{H2} = 83$





# Kryptographische Hashwertfunktion (i)

---

- Ursprünglich für schnelle Sortier- und Suchalgorithmen eingeführt.
- Reine Prüfsummen sind ungeeignet für die Kryptographie
  - leicht möglich verschiedene Nachrichten mit gleicher Prüfsumme zu erzeugen.
  - Prüfsummenverfahren wäre eine XOR-Verknüpfung aller Bytes eine Nachricht. Die entstehende Prüfsumme ist wieder ein Byte lang.
  - Beispiel: „Hallo Bob“ entspricht im ASCII-Zeichensatz in hexadezimaler Byte-Folge:
    - 48 61 6C 6C 6F 20 42 6F 62
    - Die XOR-Prüfsumme lautet:  $48 \oplus 61 \oplus 6C \oplus 6C \oplus 6F \oplus 20 \oplus 42 \oplus 6F \oplus 62 = 29$
  - Durch Modifikation der Nachricht in „Hallo Tot“ oder „Hurraton“ wird die XOR-Prüfsumme nicht verändert.
    - Paare von Nachrichten, die bei einer vorgegebenen Hashfunktion den gleichen Hashwert liefern, werden auch als Kollisionen bezeichnet.
  - Kryptographische Hashfunktionen stellen wesentliche höhere Anforderungen an eine Hashfunktion als z.B. der einfache Cyclic Redundancy Check (CRC) die in der reinen Datenübertragung häufig eingesetzt wird.





## Kryptographische Hashwertfunktion (ii)

---

- Kryptographie benutzt **Einweg** Hashfunktionen
  - Sie muss für beliebig lange Nachrichten einen (relativ) kleinen Wert mit vorgegebener Länge liefern
  - Sie muss von jedermann leicht zu berechnen sein
  - Es muss praktisch ausgeschlossen sein, zu einem vorgegebenen Hashwert eine Nachricht zu konstruieren, die genau diesen Hashwert liefert. (*Kollisionsfrei*)
    - *Weiterhin darf der Hashwert nicht kleiner sein als 128 Bit, um einen Geburtstagsangriff zu widerstehen. Dies würde einen Angreifer dazu zwingen,  $2^{64}$  Dokumente zu untersuchen, um zwei Nachrichten zu finden, die den gleichen Hashwert haben.*
    - *Es wird derzeit dazu übergegangen einen 160 Bit Hashwert bei dem  $2^{80}$  Dokumente untersucht werden müssen einzusetzen.*
- In der Praxis werden Einweg-Hashfunktion auf Grundlage einer Kompressionsfunktion entworfen. Dabei liefert die Funktion einen Hashwert z.B. der Länge n zu einem größeren Eingabeblock der Länge (m). Die Kompressionsfunktion enthält als Eingabe eine Nachricht (M) und die Ausgabe der vorherigen Textblöcke.

$$h_i = f(M, h_{i-1})$$

- MD5-Message-Digest-Verfahren wurde von Ron Rivest am MIT entwickelt.
  - Der Algorithmus ist im RFC-1321 definiert.
  - Die komplette Spezifikation kann z.B. vom FTP-Server der UNI-Köln (<ftp://ftp.uni-koeln.de/rfc1300-1499>) eingesehen werden.

MD5 ist nicht mehr tauglich um für elektronische Signaturen

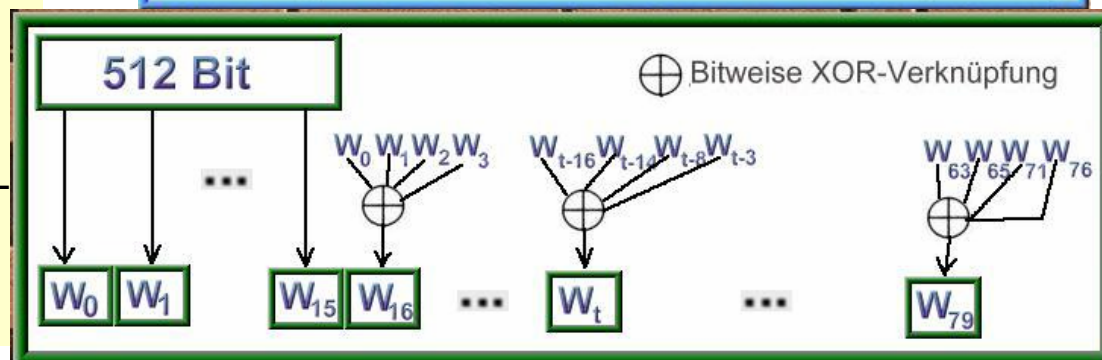
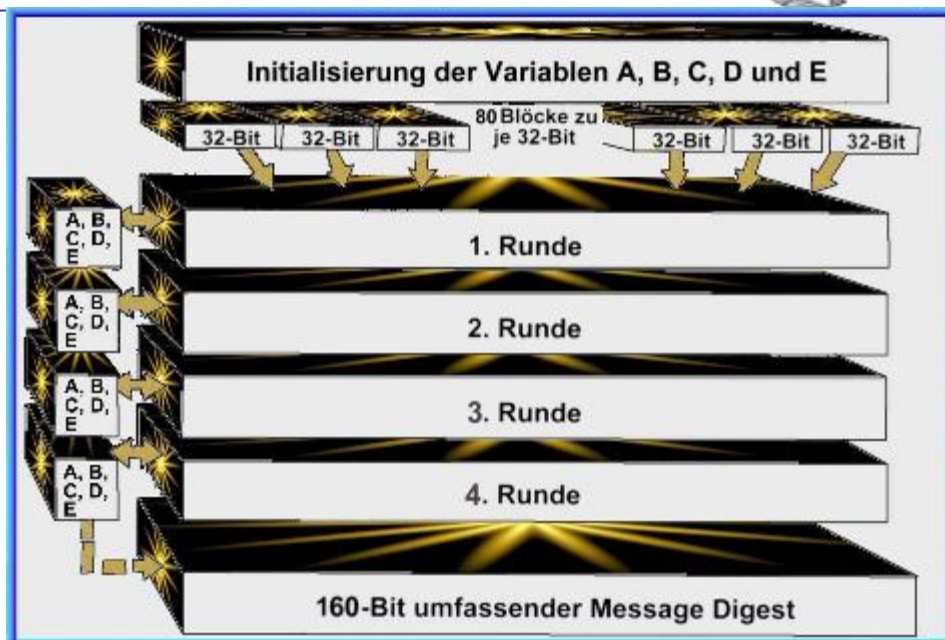
---





## Beispiel: SHA (NIST, 1993)

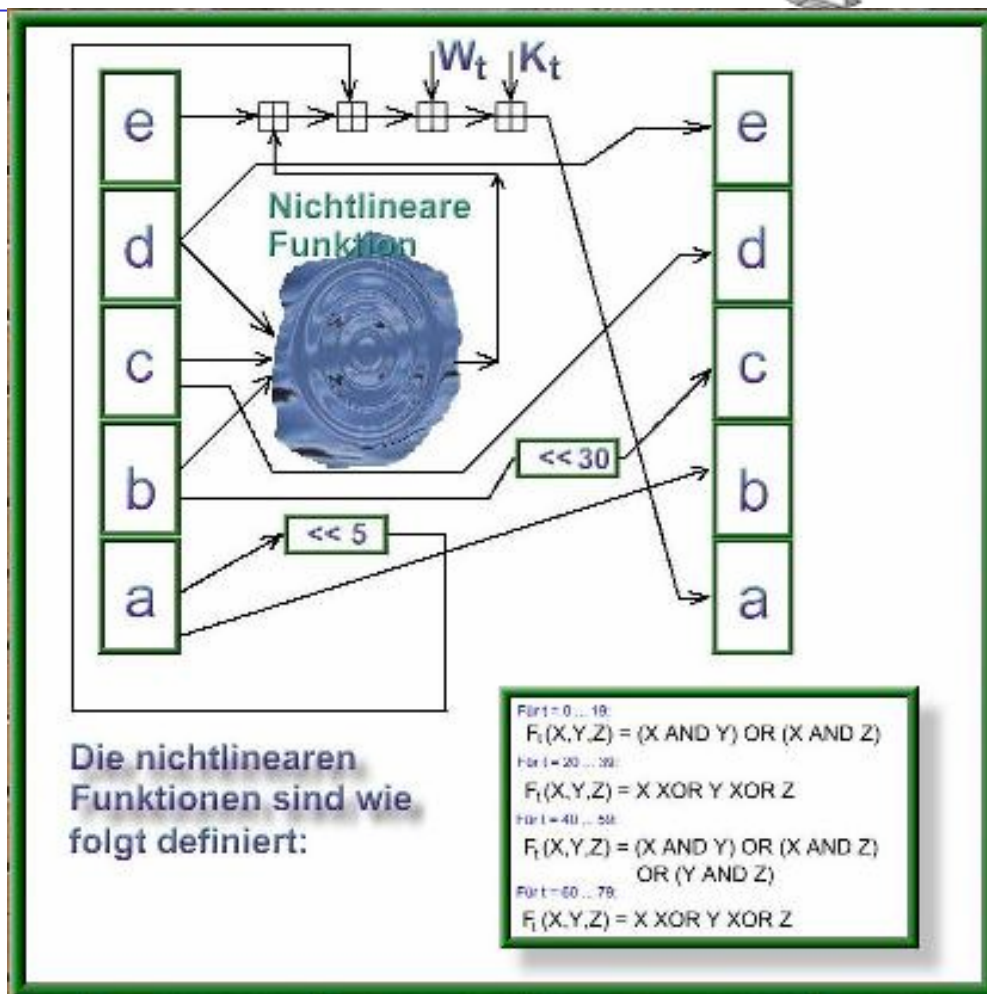
- SHA verarbeitet nur Nachrichten (M) von  $2^{64}$  Bits max. Länge
  - Der Message Digest hat eine Länge von 160 Bit Der Algorithmus unterteilt die Nachricht in 512 Bit-Nachrichtenblöcke
1. Falls (M) kleiner als 512 Bit ist wird (M) mit einem Eins-Bit und lauter Nullen aufgefüllt (Füllbits) plus als 64-Bit-Zahl die Länge der Nachricht.
  2. Es werden fünf Variablen (A=0x67452301, B=0xefcdab89, C=0x98badcfe, D=0x10325476, E=0xc3d2e1f0) initialisiert.
  3. Der Eingabeblock wird in achtzig 32-Bit-Teilblöcke unterteilt die durch vier Runden gemischt werden





## Beispiel: SHA: Ablauf einer Runde

- In einer Runde werden jeweils zwanzig Arbeitsschritte (R1:  $t=0\dots 19$ , R2:  $t=20\dots 39$ , R3:  $t=40\dots 59$ , R4:  $t=60\dots 79$ ,
  - Weiterhin ändert sich bei jeder Runde die nichtlineare Funktion, sowie die Konstante  $K_t$ , die rundenabhängig ist.
  - Z.B. für R1. ist  $K_t = 0x5a827999$
4. Schritt 3 wird solange durchlaufen, bis alle vier Runden bzw. 80 Arbeitsschritte ( $t=0\dots 79$ ) durchlaufen sind. Das Ergebnis ist dann ein 160-Bit-Message-Digest des 512-Bit-Eingabeblocks
  5. Jetzt muss Schritt 4. So lange durchlaufen werden, bis alle 512-Bit-Eingabeblocks der Nachricht verarbeitet werden. Der letzte 160-Bit Message-Digest ist dann der der gesamten Nachricht.





# Literatur

---

- <http://www.tu-darmstadt.de/ss/comments/20.183.1>
- **ITU-T Recommendation Series X: Data Networks and Open System Communications, Directory X.509 (03/2000)**
- **ITU-T Recommendation Series X: Data Networks and Open System Communications, Directory X.500 (02/2001)**
- **Trust Center: DUD Fachbeiträge, Vieweg Verlag, ISBN 3-528-05523-5**
- **Buchmann J.: Einführung in die Kryptographie, 2., erweiterte Auflage, Springer Verlag, ISBN 3-540-41283-2.**
- **Selke, G.: Kryptographie, O'Reilly-Verlag, ISBN 3-89721-155-6**
- **Burnett S., Paine S.: Kryptographie, mitp-Verlag, deutsche Übersetzung.**
- **Singh, S.: Die Kunst der Verschlüsselung, Hanser-Verlag München 2002, ISBN 3-446-20169-6**
- **Ertel W.: Angewandte Kryptographie, Hanser-Verlag, München 2003, ISBN 3-446-22304-5**
- **Bauer, F.L.: Entzifferte Geheimnisse, Methoden und Maximen der Kryptologie, zweite erweiterte Auflage, Springer Verlag ISBN 3-540-626332-8**
- **Wohlmacher P.: Digitale Signaturen und Sicherheitsinfrastrukturen, it-Verlag, ISBN 3-936052-01-8.**





# Übungen

---

- **Was ergibt:**  $27 \bmod 12 =$
- **Was ergibt:**  $-27 \bmod 12 =$
- **Was ergibt:**  $-27 \bmod -12 =$
- **Was ergibt:**  $27 \bmod -12 =$





# Übungen

---

- **Was ergibt:**  $27 \bmod 12 =$   
**3** **Rechnung:** denn  $(2 * 12) + 3 = 27$  (also mit  $q = 2$ ,  $\bmod n = 12$  und Rest  $r = 3$ )
- **Was ergibt:**  $-27 \bmod 12 =$   
**9** **Rechnung:** denn  $(-3) * 12 + 9 = -27$  (also mit  $q = -3$ ,  $\bmod n = 12$  und Rest  $r = 9$ )
- **Was ergibt:**  $-27 \bmod -12 =$   
**-3** **Rechnung:** denn  $(2 * (-12)) - 3 = -27$  (also mit  $q = 2$ ,  $\bmod n = -12$  und Rest  $r = -3$ )
- **Was ergibt:**  $27 \bmod -12 =$   
**-9** **Rechnung:** denn  $(-3) * (-12) - 9 = 27$  (also mit  $q = 3$ ,  $\bmod n = -12$  und Rest  $r = -9$ )

