technische universität
dortmund

Vorlesung (WS 2014/15)

Sicherheit: Fragen und Lösungsansätze

Dr. Thomas P. Ruhroth

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

Kryptographie I

[mit freundlicher Genehmigung basierend auf einem Foliensatz von

Prof. Dr. Claudia Eckert (TU München)]

Literatur:

Claudia Eckert: IT-Sicherheit: Konzept - Verfahren - Protokolle, 7., überarb. und erw. Aufl., Oldenbourg, 2012.

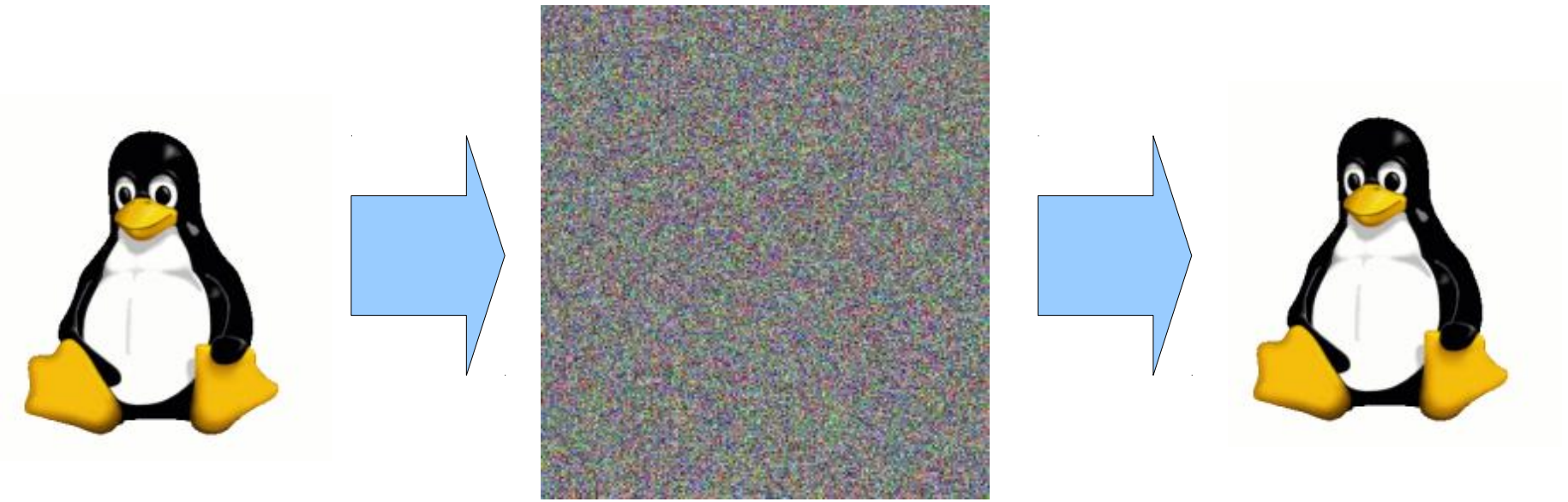
E-Book:
http://www.ub.tu-dortmund.de/katalog/titel/1362263

Agenda

- Grundlagen
- Symmetrische Verfahren
  - One-Time-Pad
  - DES
  - AES

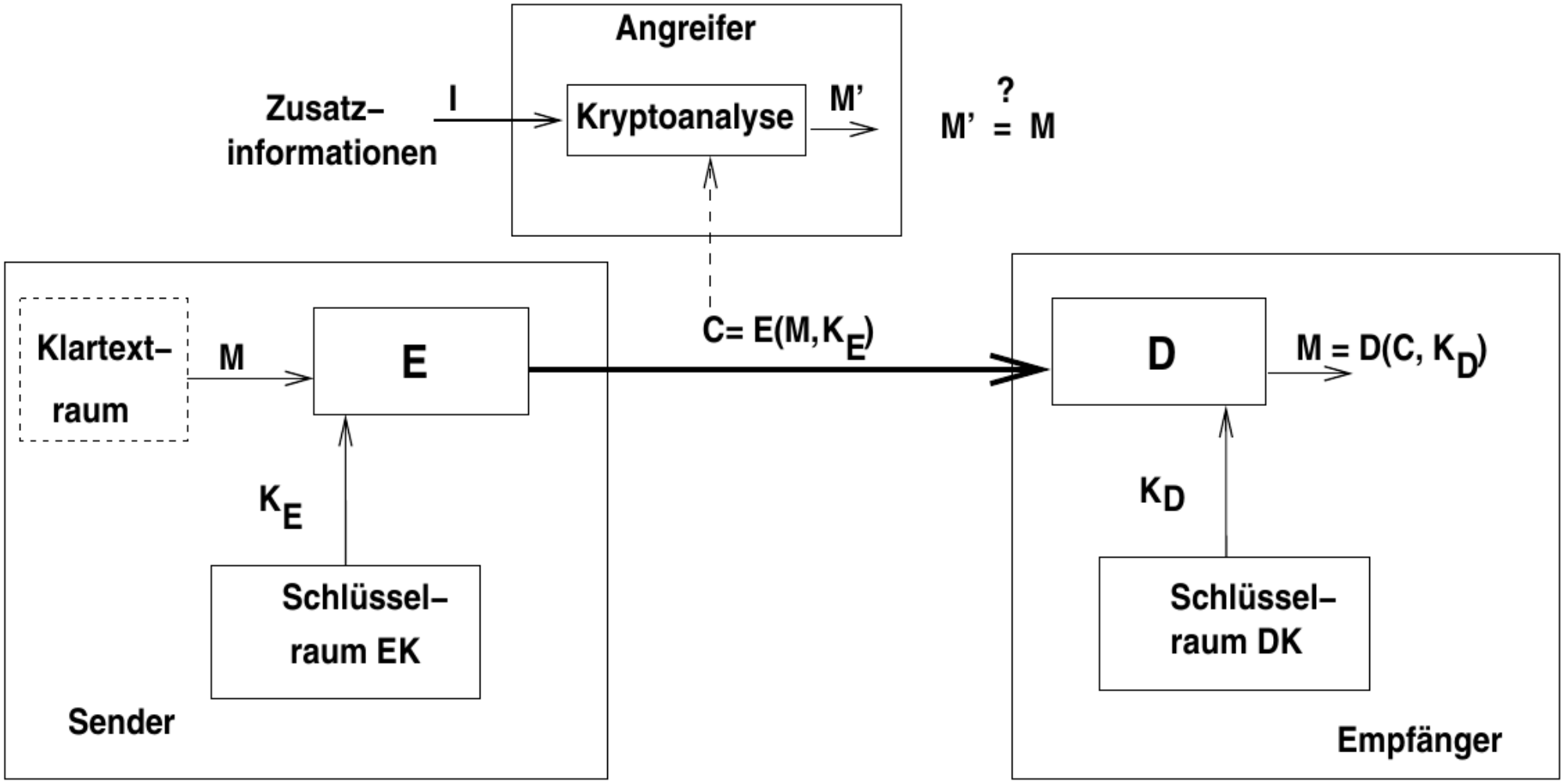
- Kennen der Problemstellung der Kryptographie
- Grundlegende Bestandteile von symmetrischen Verfahren
- Drei Algorithmen kennen

Verschlüsselung



Tux the Penguin, the Linux mascot. Created in 1996 by Larry Ewing and The GIMP

Kryptographische Verfahren



Formal: Kryptographische Verfahren

Definition: Ein kryptographisches Verfahren ist gegeben durch ein Tupel: (M,C,EK,DK,E,D)

1. M ist die Menge von Klartextnachrichten (Plaintext) *M*, über dem Alphabet A_1
2. C ist die Menge von Kryptonachrichten (Chiffretext) *C*, über dem Alphabet A_2
3. der Menge von Verschlüsselungs-Schlüsseln EK,
4. der Menge von Entschlüsselungs-Schlüsseln DK, und der Abbildung:

   $f : EK \rightarrow DK$, mit: $d = f(e)$, $e \in EK$, $d \in DK$
5. Der Familie E der Verschlüsselungsverfahren $E = \{E_k \mid E_k : M \rightarrow C\}$
6. Der Familie D der Entschlüsselungsverfahren $D = \{D_k \mid D_k : C \rightarrow M\}$, mit

   $\forall M \in M$ gilt: $D_d(E_e(M)) = M$ mit $e \in EK$, $d \in DK$, $f(e) = d$

Kryptographische Verfahren

Ziel: Grundlagen zu Krypto-Verfahren

- Kryptographie:

  Lehre von den Methoden zur Ver- und Entschlüsselung

- Kryptoanalyse:

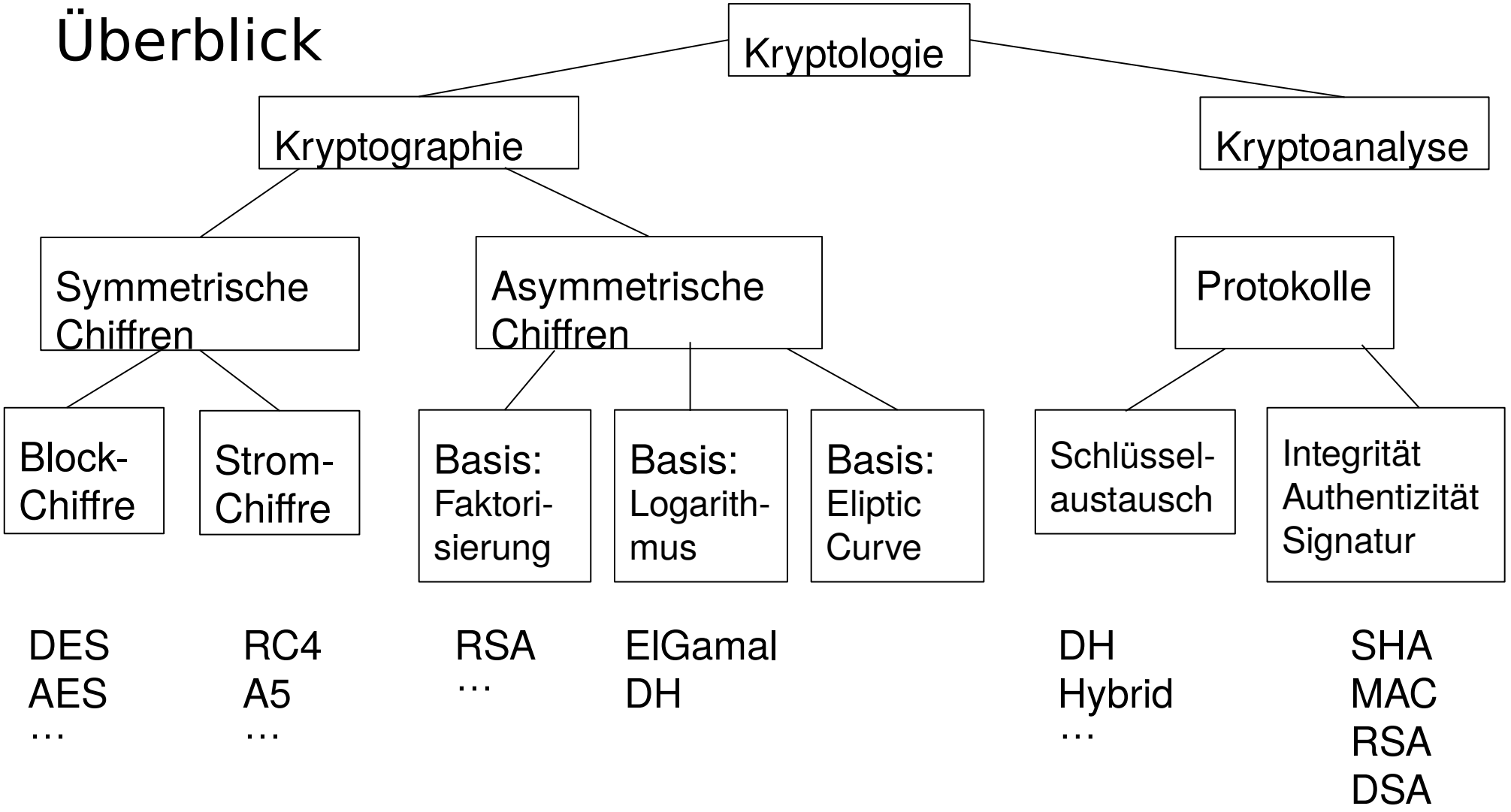
  Wissenschaft von Methoden zur Entschlüsselung

- Kryptologie:

  Kryptographie und -analyse, eng verzahnt

Frage: was ist Steganographie? Abgrenzung zur Kryptographie?

Überblick


Kryptologie
Kryptographie
Kryptoanalyse
Symmetrische Chiffren
Asymmetrische Chiffren
Protokolle
Block-Chiffre
Strom-Chiffre
Basis: Faktori-sierung
Basis: Logarith-mus
Basis: Eliptic Curve
Schlüssel-austausch
Integrität Authentizität Signatur
DES
AES
…
RC4
A5
…
RSA
…
ElGamal
DH
DH
Hybrid
…
SHA
MAC
RSA
DSA

Anforderungen

Anforderungen an kryptographische Verfahren:

- Sicherheit darf nicht von Geheimhaltung der Ver- und Entschlüsselungsfunktionen abhängen!

  Häufiger Verstoß dagegen: **Security by Obscurity**

Kerckhoffs-Prinzip:

- Stärke des Verfahren sollte nur von der Güte des geheimen Schlüssels abhängen!
- **Bem.:** Berechnungsaufwand zum Schlüsselknacken: abhängig u.a.
  - von Rechnertechnologie: Multi-Cores: gut/schlecht?
  - von Parallelisierungsmöglichkeit: P2P, Cloud, Quanten-Comp?

Bewerten der Sicherheit

- Idee:
  - Ein kryptographisches System ist sicher, wenn es keinen besseren Angriff als Raten gibt.
- Vorsicht: Schlüsselraum-Größe muss beachtet werden!
  - Zu kleiner Schlüssel Raum
    → Alle Schlüssel schnell ausprobiert

Schlüsselraum

- **Konsequenz:**
  - Schlüsselraum *EK* muss **sehr groß** sein
  - Ausprobieren aller Schlüssel (brute-force) soll nicht mit praktikablem Aufwand möglich sein (**exhaustive search nicht möglich**)
- **Beispiel:** 56-Bit Schlüssel (u.a. DES): Schlüsselraum = 2^{56}
  - 1998 Deep-Crack-Supercomputer: Kosten ca. 250.000 $
    - Knacken eines DES-Schlüssels in 56 Stunden!
  - 2006: COPACOBANA (http://www.copacobana.org/):
    - < 10.000 $, durchschnittlich 7 Tage zum Knacken von DES
- **Anforderung:** (u.a. von Bundesnetzagentur Januar 2012)
  - symmetrische Verfahren: Schlüssel ≥ 128 Bit
  - asymmetrische Verfahren: Schlüssel ≥ 2048 Bit,
    Schlüssel ≥ 224 Bit bei ECC-Varianten

One-Time-Pad

- Idee:
  - Schlüssel ist genauso lang wie die Nachricht
  - Schlüssel XORbit Nachricht = Chipher
- Schlüssel muss
  - genauso lang sein wie die Nachricht,
  - gleichverteilt zufällig gewählt werden,
  - geheim bleiben und
  - darf nicht wiederverwendet werden, auch nicht teilweise.

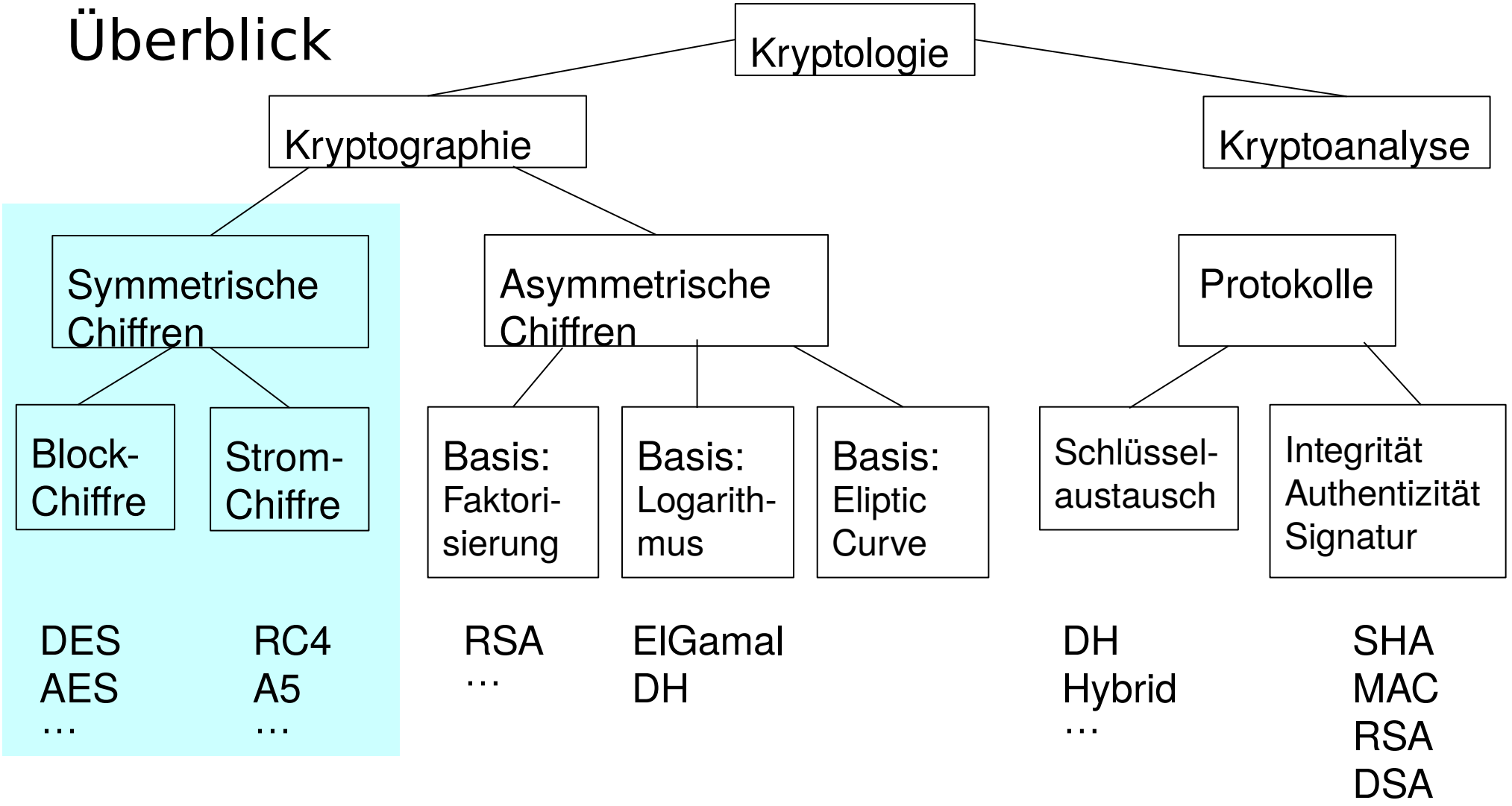
One-Time-Pad

- Idee:
  - Schlüssel ist genauso lang wie die Nachricht
  - Schlüssel XORbit Nachricht = Chipher
- Kerckhoffs-Prinzip ist vollständig erfüllt:
  - Anzahl Schlüssel = Anzahl Nachrichten = Anzahl Chiphertexte
- Probleme:
  - Lange Schlüssel
  - Schlüssel nur einmal verwendbar

XOR

- Viele Verfahren kann man als One-Time-Pad mit einer Einmalschlüsselberechnung aus einem Startschlüssel und der Nachricht ansehen
- XOR ist effektive zu berechnen (Hardware)
- XOR ist selbstinvers

Überblick


Kryptologie
Kryptographie
Kryptoanalyse
Symmetrische Chiffren
Asymmetrische Chiffren
Protokolle
Block-Chiffre
Strom-Chiffre
Basis: Faktori-sierung
Basis: Logarith-mus
Basis: Eliptic Curve
Schlüssel-austausch
Integrität Authentizität Signatur
DES
AES
…
RC4
A5
…
RSA
…
ElGamal
DH
DH
Hybrid
…
SHA
MAC
RSA
DSA

Symmetrische Verfahren

- Ver- und Entschlüsselungs-Schlüssel sind gleich, oder leicht auseinander ableitbar, $d = f(e)$
- Nutzung eines gemeinsamen, geheimen Schlüssels (Secret-Key)

Bekannte Repräsentanten:

- DES (Data-Encryption-Standard): noch immer weit verbreitet
- AES (Advanced Encryption-Standard)
- RC4, A5/3

Achtung:

- Starke Chiffre: notwendig, aber nicht hinreichend zur Gewährleistung des Schutzziels Vertraulichkeit, **warum?**

Key-Management

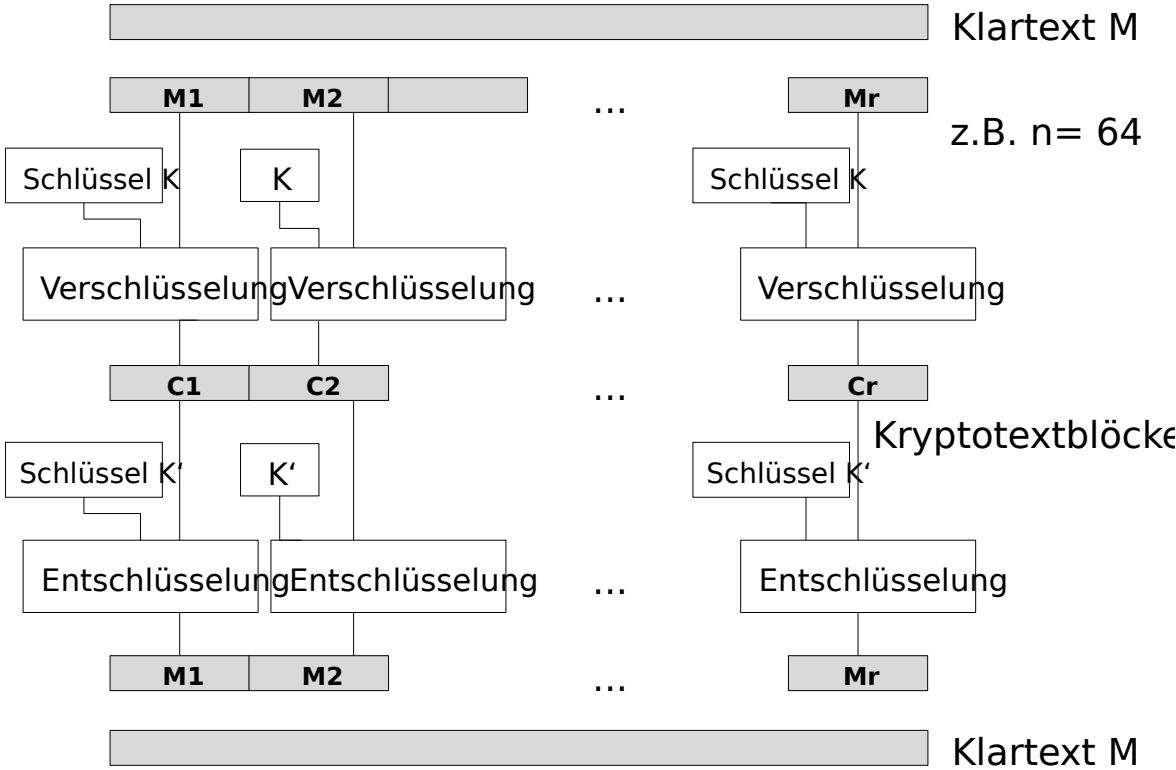
Starke Verfahren und gutes Key-Management sind erforderlich

1. Schlüsselgenerierung, hohe Entropie, starke Schlüssel
2. Sicherer Austausch des gemeinsamen Schlüssels
3. Sichere Schlüsselspeicherung

Blockchiffre und Stromchiffre

Blockchiffre

- zu verschlüsselnder Klartext wird in Blöcke fester Länge aufgeteilt (z.B. 64 Bit)
- blockweises Verschlüsseln,
- für jeden Block gleicher Schlüssel K

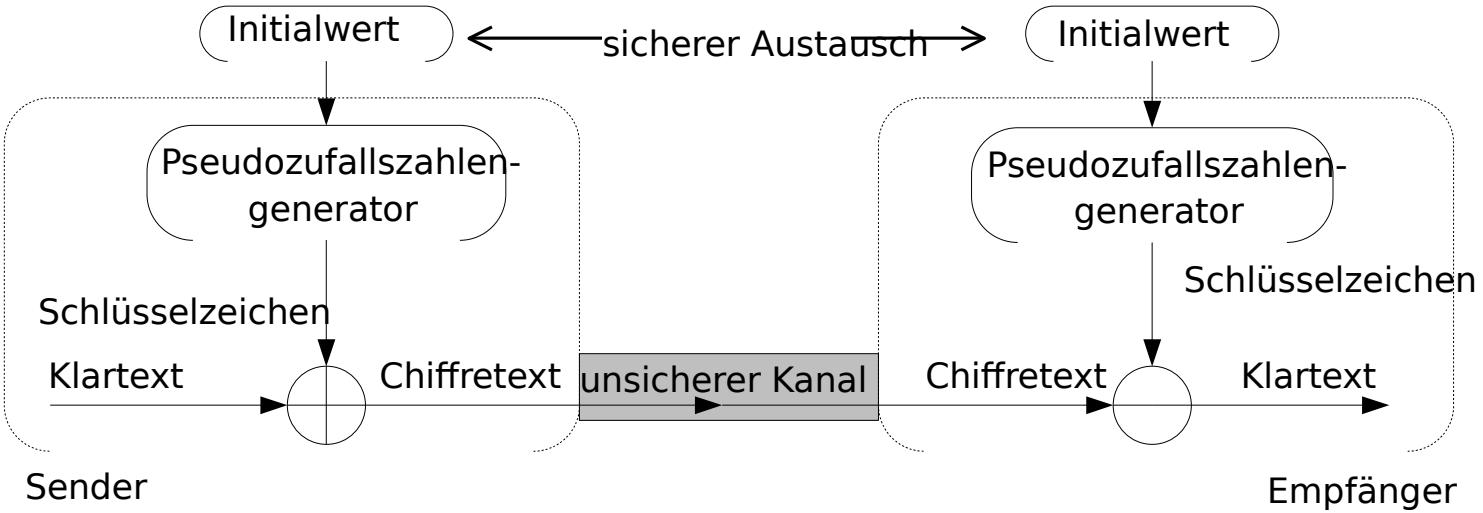


Verschiedene Modi: ECB, CBC, CFM, OFM, CM (Counter Mode)

Stromchiffre

Bit-weise Verschlüsselung mittels (pseudo)zufälligem Schlüsselstrom *K*

- Schlüssel K hat die gleiche Länge wie Nachricht *M*,
- Verschlüsselung: *M* XOR *K* (bitweise XOR bzw. add mod 2)
- **Beispiele für Einsatz** von Stromchiffren?



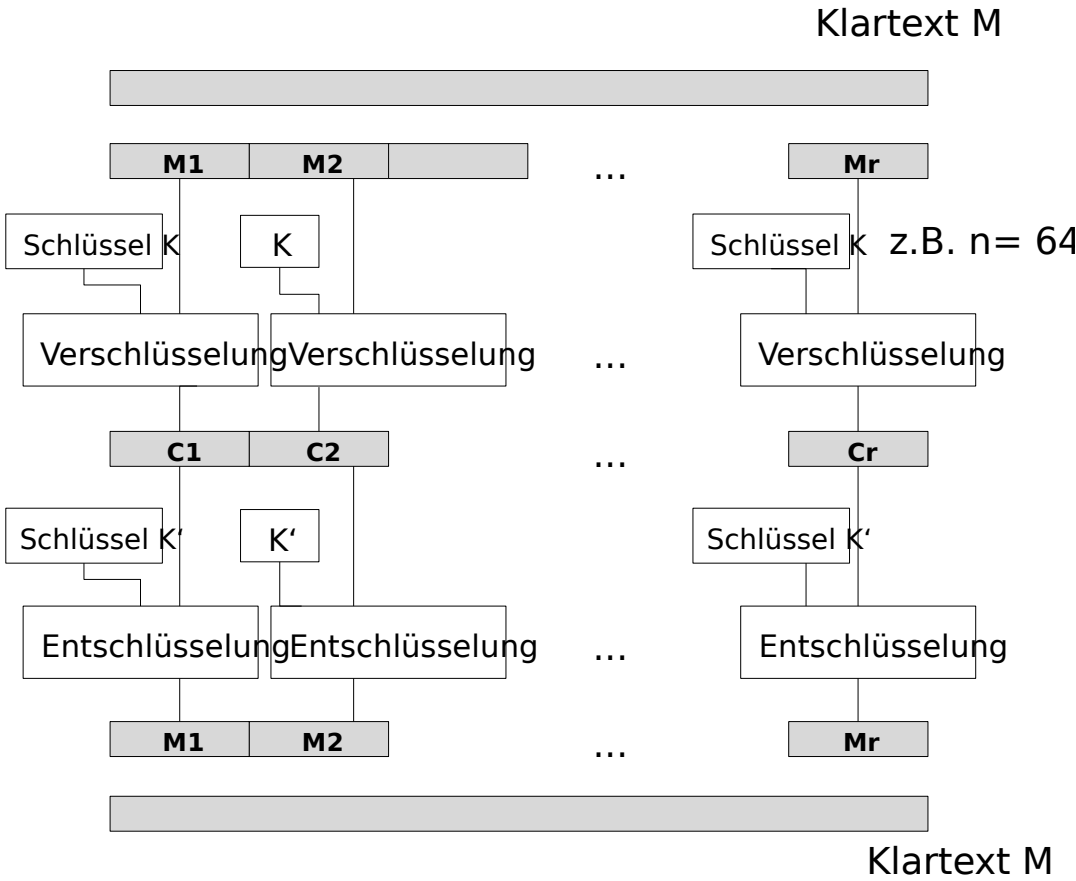
Geforderte Eigenschaft

- Für die Verschlüsselung unterschiedlicher Nachrichten M_1 und M_2 müssen stets unterschiedliche Schlüsselströme K_1, K_2 verwendet werden
- Angriffsszenario bei Verwendung gleicher Schlüssel K_i

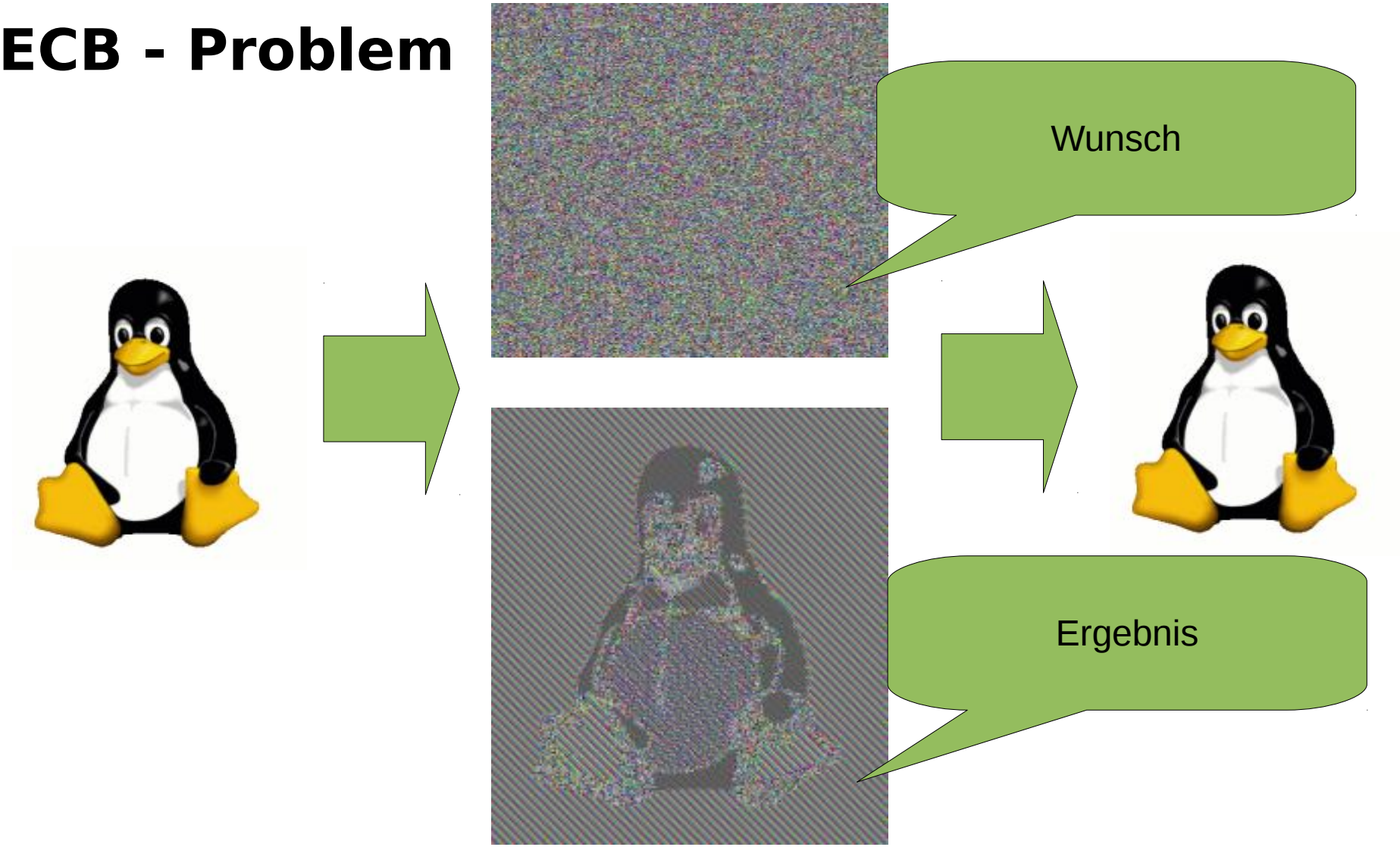
Frage: Wie kann man effizient die Schlüsselströme K_i generieren?

Verschlüsselungs-Modi: ECB – Electronic Code Book

- Verschlüsselung kann parallelisiert werden: sehr effizient
- Verschlüsselungsschlüssel kann wiederholt verwendet werden, ggf. problematisch?
- Gleiche Klartextblöcke ergeben gleiche Chiffretextblöcke, ggf. problematisch?



ECB - Problem



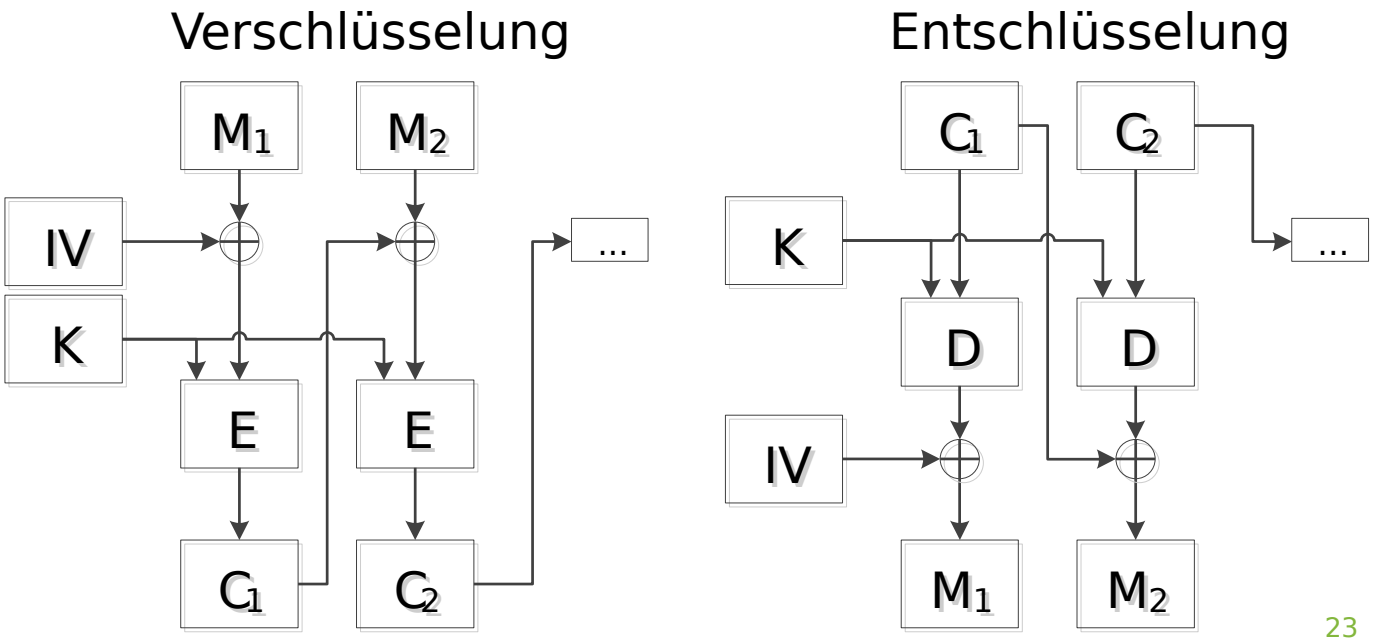
Tux the Penguin, the Linux mascot. Created in 1996 by Larry Ewing and The GIMP

CBC - Cipher Block Chaining

Klartextblock wird mit vorherigem Chiffretextblock verknüpft (XOR)

- Startwert = Initialisierungsvektor IV wird benötigt
- Gleiche Klartextblöcke ergeben ungleiche Chiffretextblöcke
- Einspielungen sind erkennbar: Fehler bei Entschlüsselung

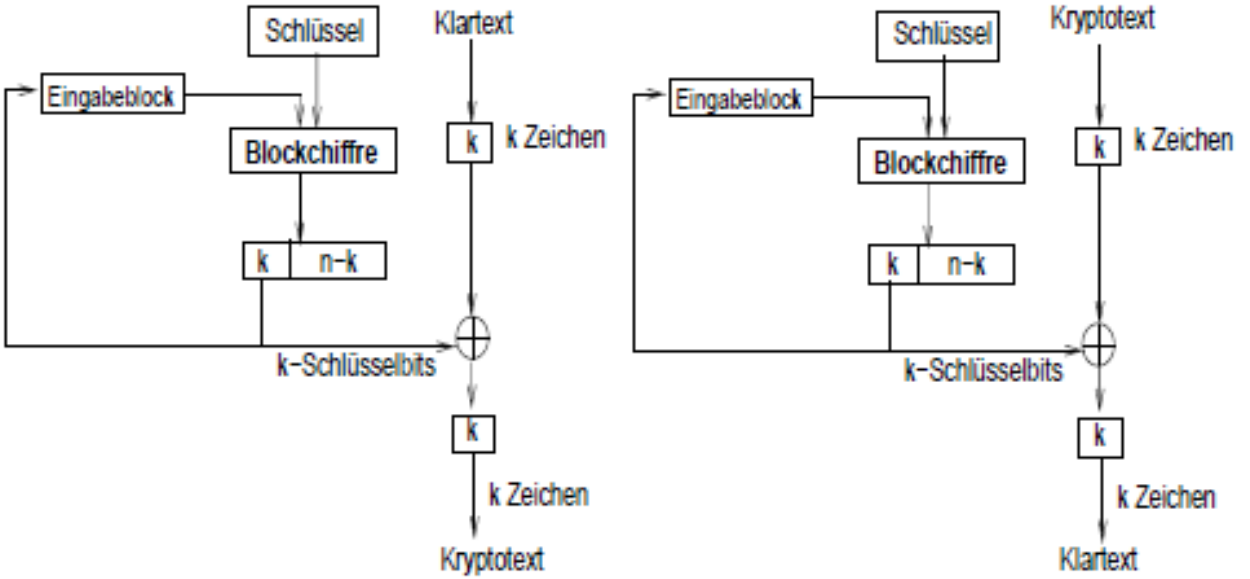
Probleme?



OFB oder CFB: Output Feedback oder Cipher Feedback

- Blockchiffre wird als Stromchiffre verwendet
- Chiffren fungieren als Pseudozufallszahlengeneratoren (die eigentliche Stromchiffre ist dann wieder eine XOR Verschlüsselung)
- Generierung von Schlüsselbits: Eingabeblock wird verschlüsselt
- Empfänger generiert die gleichen Schlüsselbits

Probleme?



Verschlüsselung: OFB-Modus Entschlüsselung

DES (Data Encryption Standard)

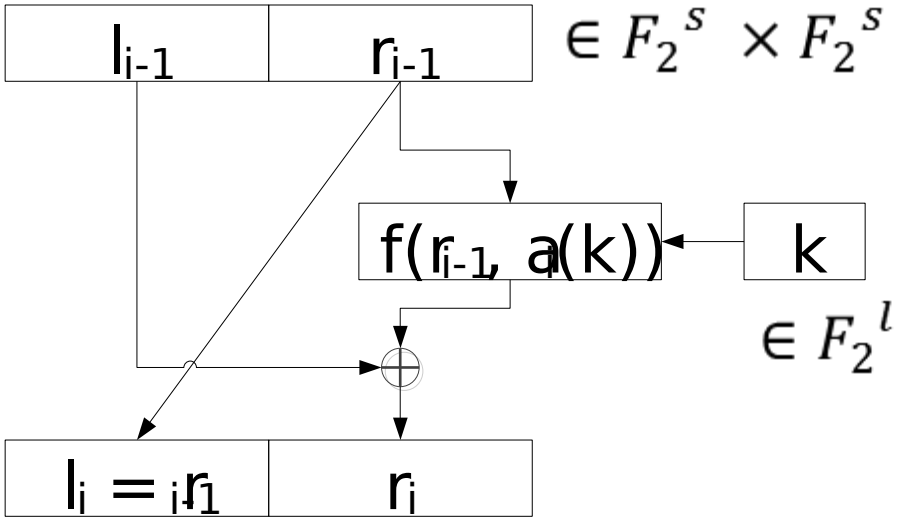
Große Akzeptanz und Verbreitung: u.a. Bank-Umfeld

- Bis 1998 zertifiziert als Verschlüsselungsstandard, seit 2001 AES

Eigenschaften:

- DES ist eine Blockchiffre, 64 Bit Blocklänge,
- Schlüssellänge: 64 Bit, real nur 56 Bit, 8 Bit sind Paritybits
- Symmetrisch: Schlüssel für Ver- und Entschlüsselung verwendet
- DES: kommt in allen Modi zum Einsatz, am Häufigsten DES-CBC
- DES ist in Hardware sehr effizient implementierbar
- DES basiert auf Techniken zur Diffusion und Konfusion (Shannon)
  - Konfusion: Verschleiern des Zusammenhangs zwischen Schlüssel und Cyphertext: Substitutionen als häufige Technik

- Diffusion: statistische Eigenschaften verschleiern durch Verbreitung des Einflusses von Plaintext-Bits auf viele Bits des Ciphertextes
  - Bitweise Permutation ist hierfür einfache Technik oder
  - Mixcolum (wird beim AES genutzt)
- DES kombiniert Substitutionen und Permutationen: Produktchiffre
- DES ist eine Feistel-Chiffre:
  - Plaintext-Blöcke werden in Runden verarbeitet,
  - Block wird in linke und rechte Hälfte aufgeteilt,
  - f wird auf rechte Hälfte und den Rundenschlüssel k angewandt,
  - Ergebnis wird mit linker Hälfte XOR verknüpft

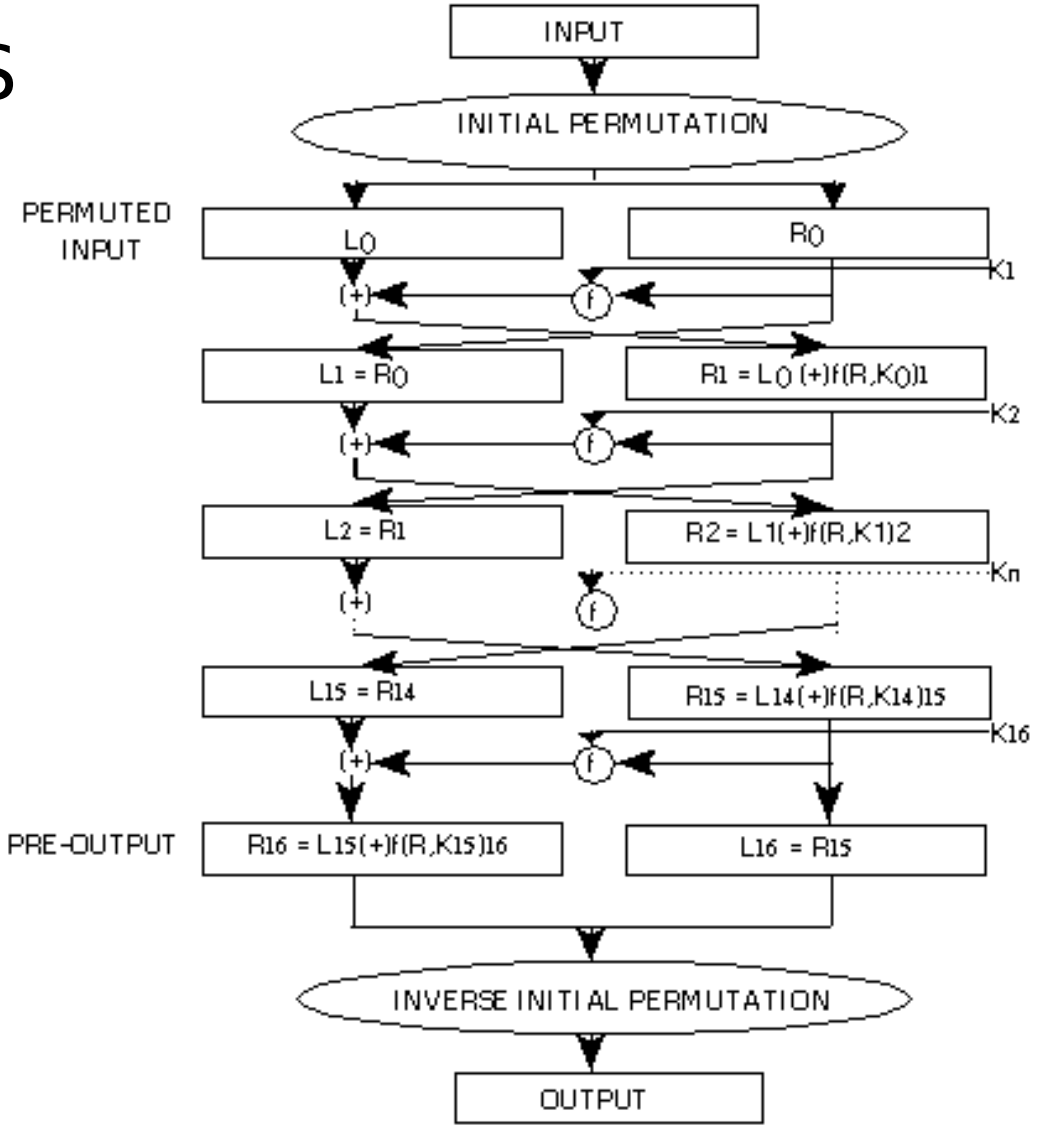


Funktionsweise des DES

- 64-Bit Eingabeblock
- Initialpermutation IP

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7



- 64 Bit Schlüssel K
- 16 Runden mit je einem Teilschlüssel

  K_i abgeleitet aus K

Die Funktion f

E: Expansion:

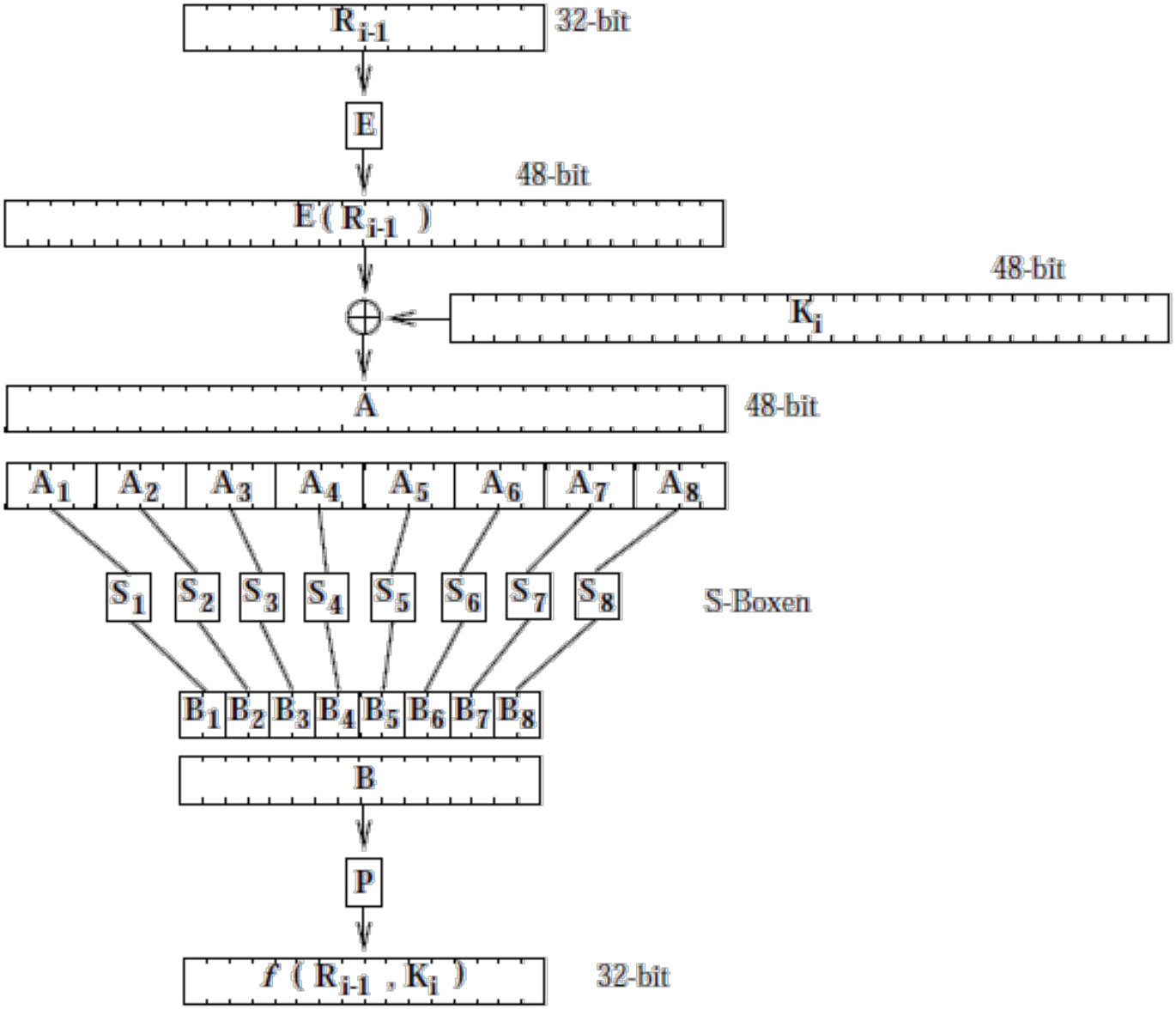
32-bit Eingabe durch Verdopplung von Bits auf 48 erweitern

8 Blöcke a 6 Bit,

Eingabe in S-Boxen

S1 – S8: Substitution;

6 Bit durch 4-Bit ersetzt



S-Box S1 Substitution:

ein 6-Bit Eingabeblock wird substituiert durch einen 4-Bit Ausgabeblock

S1	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7
1	0	F	7	4	E	**2**	D	1	A	6	C	B	9	5	3	8
2	4	1	E	8	D	6	2	B	F	C	9	7	3	A	5	0
3	F	C	8	2	4	9	1	7	5	B	3	E	A	0	6	D

Beispiel: S1(0 0101 1) = (2)$_{16}$ = (0010)$_2$

Spalte

Zeile

DES-Entschlüsselung

Entschlüsselungsalgorithmus ist nahezu gleich dem Verschlüsselungsverfahren des DES:

Ursache: Eigenschaften der zugrundeliegender Feistel-Chiffre

- Umkehrung der Verschlüsselung erfolgt Runde-für-Runde
- Entschlüsselungsrunde 1 kehrt Verschlüsselungsrunde 16 um
- **Basis dafür:**

  die finale Permutation (IP^{-1}) der Verschlüsselung und die initiale Permutation IP der Entschlüsselung heben sich auf:

- **Effekt:** Eingabe (L_0^d, R_0^d) für Entschlüsselung:

  (L_0^d, R_0^d) = (R_{16}, L_{16}) (also letzte Runde der Verschlüsselung)

- Entschlüsseln: Rundenschlüssel in umgekehrter Reihenfolge anwenden:

  Runde 1: Schlüssel 16 , etc.

DES Sicherheit

- Durch S-Boxen ist der kryptographische Kern nicht-linear, durch Substitution wird Konfusion erzeugt
- Spezifikationen und der Aufbau der S-Boxen, Permutationstabellen sind seit 1977 offengelegt (gut!),
- Auch wenn Obscurity-Bestandteile dabei waren:

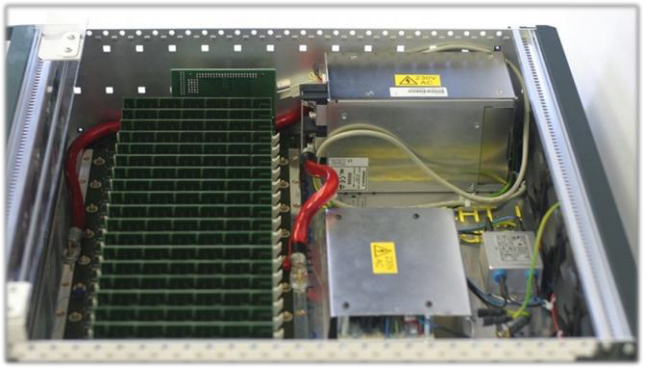
  Design-Kriterien für S-Boxen waren erst viel später offen, das führte zu jahrelangen Gerüchte bezüglich möglicher Back-Doors!
- **Bemerkenswert:** der Kryptoanalyse 15 Jahre voraus:
  - Design der S-Box war so, dass der DES robust gegen Differentielle Kryptoanalyse ist
  - diese Analyse Technik wurde aber erst 1990 von Biham und Shamir öffentlich als sehr starke Analyse-Technik ‚entdeckt'

DES Sicherheit (Fortsetzung)

- Durch die Permutation P wird Diffusion erreicht:
  - die 4 Outputbits einer S-Box werden so permutiert, dass sie viele verschiedene S-Boxen der nächsten Runde beeinflussen
- Avalanche Effect wird erzielt (gut!)
  - Diffusion durch Expansion, P-Permutation, S-Boxen bewirkt, dass bereits nach der 5-ten Runde jedes Bit von jedem Bit des Eingabe-Blocks und von jedem Bit des Schlüssels abhängt
- **Problem:** Schlüsselraum des DES ist zu klein:
  - Brute-Force-Angriffe u.a. COPACOBANA

    Cost-optimized-Parallel Code Braker:

    Schlüssel-Knacken in weniger als 7 Tagen



Fazit

- DES ist eine starke Chiffre, stark gegen Kryptoanalysen,
- aber die Schlüssellänge von 64 Bit (real nur 56 Bit) ist für heutige Rechnerarchitekturen zu schwach

Problem: DES noch immer Bestandteil vieler Anwendungen

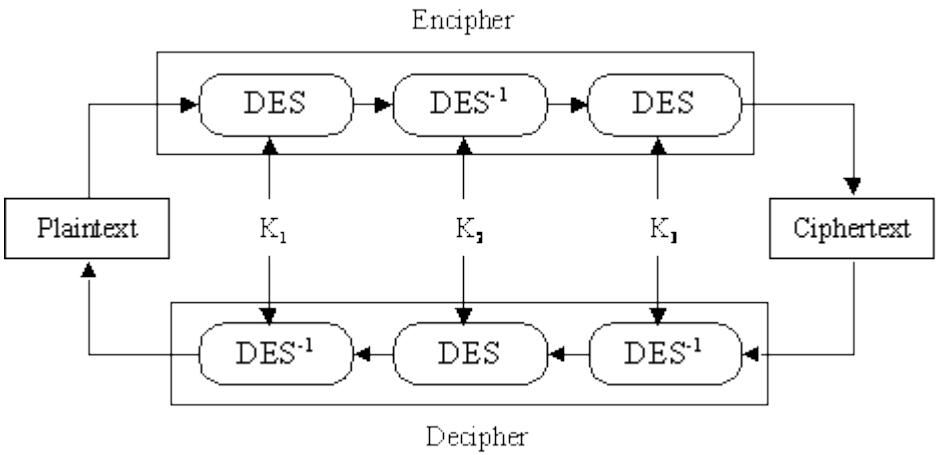
- Austausch von DES gegen stärkeres Verfahren mit längeren Schlüsseln, z.B. AES, ist sehr aufwändig.

Lösung:

- Beibehalten des DES, aber

  Vergrößern des Schlüsselraums!
- 3DES (Tripel-DES), z.B.

  $y = DES_{K3}(DES^{-1}_{K2}(DES_{K1}(x)))$



AES (Advanced Encryption-Standard)

- NIST-Kryptostandard für symmetrische Verfahren, seit 2001
- AES wird verwendet u.a. in: TLS, IPSec, 802.11i, SSH, Skype
- symmetrische Blockchiffre mit Blocklänge 128 Bit und drei Schlüssellängen: 128, 192 oder 256 Bit
- AES arbeitet auch in Runden: 10, 12, oder 14 abhängig von der Schlüssellänge, z.B. 10 Runden bei 128 Bit Schlüssellänge
- AES ist keine Feistel-Chiffre, da AES pro Iteration den gesamten Block verschlüsselt
- AES Algorithmus ist in Schichten aufgeteilt: Key Addition, Substitutions- und Diffusions-Schicht
- jede Schicht manipuliert jedes Bit der Eingabe

Überblick über AES

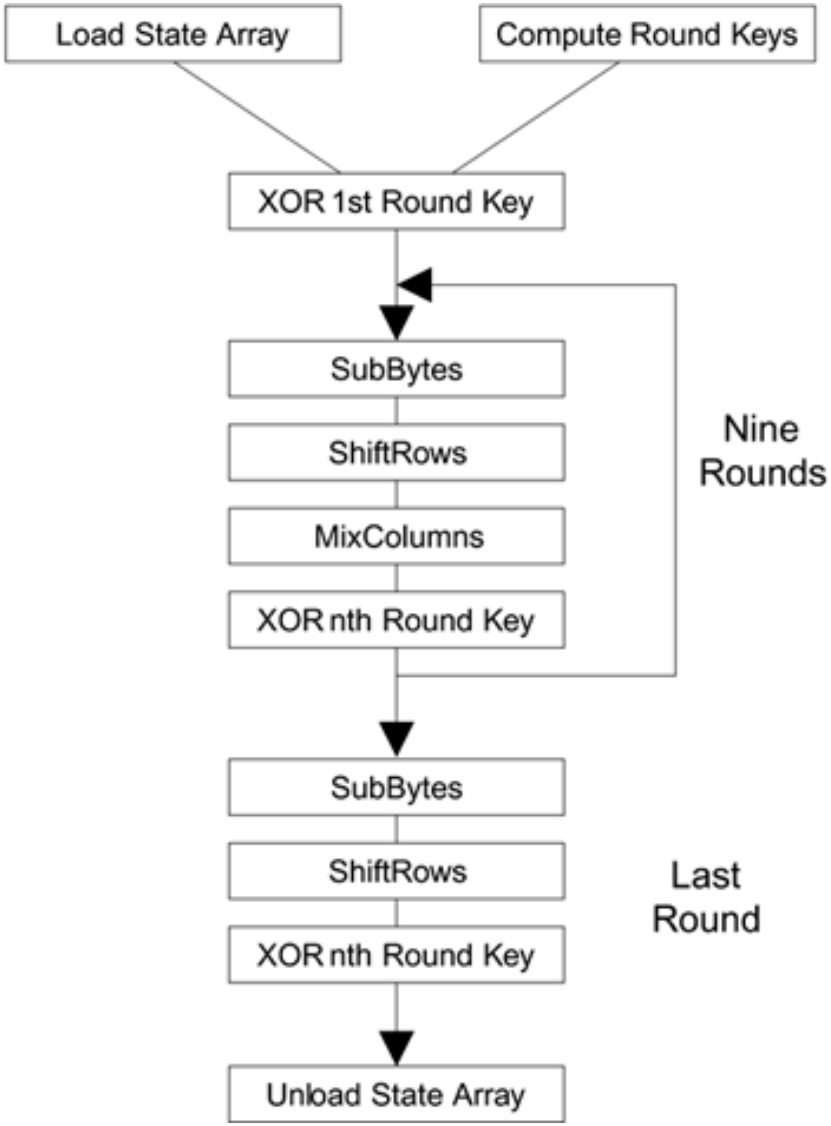
Eingabeblock B als

4x4 Byte-Matrix repräsentiert

B = b_0, …, b_{15} mit

b_0 b_4 b_8 b_{12}
b_1 b_5 b_9 b_{13}
b_2 b_6 b_{10} b_{14}
b_3 b_7 b_{11} b_{15}

analog werden auch Schlüsselbits in Matrixform repräsentiert



AES arbeitet auf Zeilen und Spalten

der Matrix.

Matrix wird auch als Zustand bezeichnet.

In jeder Runde werden 4 Transformationsschritte durchlaufen:

Round(*State*, *RoundKey*) {

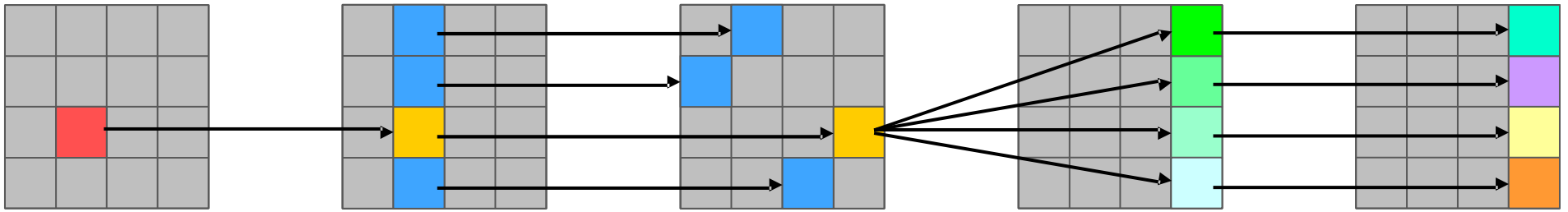
SubBytes(*State*); Substitutionschiffre, byte-weise

ShiftRows(*State*); zyklischer Links-Shift

MixColumns(*State*); Spaltenweise Multiplikation mit Polynom, jedes Byte der Spalte wird mit jedem anderen der Spalte verknüpft

AddRoundKey(*State*, *RoundKey*);

}



Mathematische Basis: Galois-Feld, Galois-Körper

- Ein Galois Feld ist eine endliche Menge von Elementen.
- Auf der Menge kann man addieren, subtrahieren, multiplizieren und das Inverse berechnen
- **AES:** Galois-Felder werden in der S-Box und bei den Operationen MixColumn und AddRoundKey verwendet
- AES arbeitet im Galois-Feld $GF(2^8)$ von 256 Elementen, jedes Element kann damit durch 1 Byte repräsentiert werden
- Die Elemente von $GF(2^8)$ werden als Polynome vom Grad 7 repräsentiert mit Koeffizienten aus GF(2).
- Jedes Element A aus $GF(2^8)$ wird somit repräsentiert durch:

  $A(x) = a_7x^7 + \ldots + a_1x^1 + a_0x^0$, a_i aus $GF(2) = \{0,1\}$

- Speicherung von A(x) 8-Bit Vektor $A = (a_7, a_6, \ldots, a_0)$

- Additionen und Subtraktion auf GF(2^8): AddRoundKey-Operation

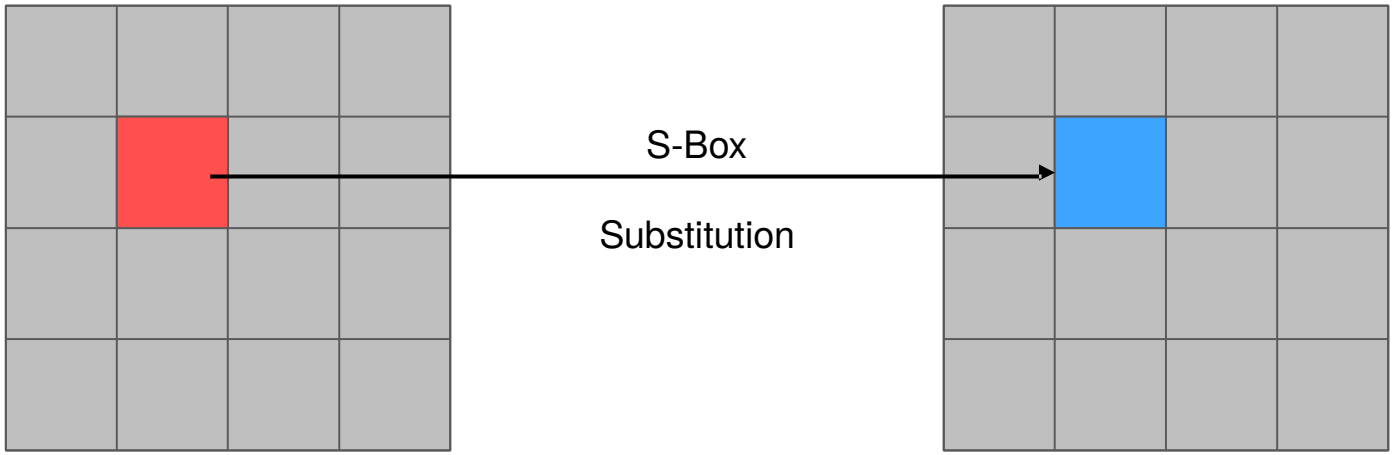
  $C(x) = A(x) + B(x) = c_7x^7 + \ldots + c_1x^1 + c_0x^0$ mit

  $c_i = a_i + b_i \bmod 2 = a_i - b_i \bmod 2$ (xor Operation)

- Multiplikation auf GF(2^8): MixColumn Operation
  - Standard Multiplikation auf Polynomen,
  - das Ergebnis wird idR. einen höheren Grad als 7 haben, es muss reduziert werden. Erforderlich ist ein
  - irreduzibles Polynom P (vergl. Primzahl) vom Grad 8
  - Reduktion: Produkt der Multiplikation wird durch P dividiert
- Bei AES in Spezifikation festgelegt: $P(x) = x^8 + x^4 + x^3 + x+1$
- Berechnen des Inversen in GF(2^8): Byte-Substitution (S-Box) : Gegeben: Element A(x) und P(x): $A^{-1}(x) * A(x) = 1 \bmod P(x)$
- In AES: Tabelle mit berechneten multiplikativen Inversen

SubBytes: Byte Substitions-Schicht

- Jedes Byte des Eingabeblocks wird durch ein anderes Byte ersetzt: Festgelegt durch die S-Box, nicht linear
- 16 identische S-Boxen in 4x4 Matrix angeordnet.
- Eine S-Box ist eine invertierbare Abbildung: $\{0,1\}^8 \rightarrow \{0,1\}^8$
- Häufig: für alle 256 Eingaben der S-Box die Ausgaben in einer vorab berechneten Tabelle abgelegt: effizienter Lookup



S-Box Konstruktion: stark gegen differentielle u. lineare Analyse

S-Box für AES

argument byte a: seen as composed of two hexadecimal symbols li and co
value byte v: table entry for line li and column co

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

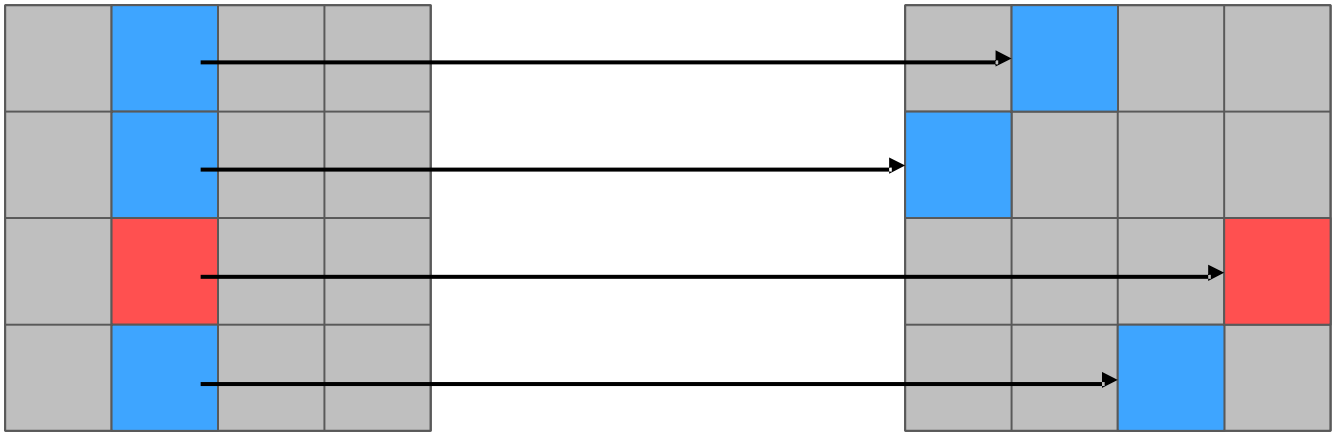
Shift Row

(Diffusions-Schicht: ShiftRows und MixColumn)

- Zyklischer Links-Shift von Bytes
  - erste Zeile, kein Shift,
  - 3-te Zeile 2 Positionen,
  - 2-te Zeile: 1 Position nach links,
  - 4-te Zeile 3 Positionen

von:
$$\begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix}$$

nach:
$$\begin{pmatrix} 1 & 5 & 9 & 13 \\ 6 & 10 & 14 & 2 \\ 11 & 15 & 3 & 7 \\ 16 & 4 & 8 & 12 \end{pmatrix}$$

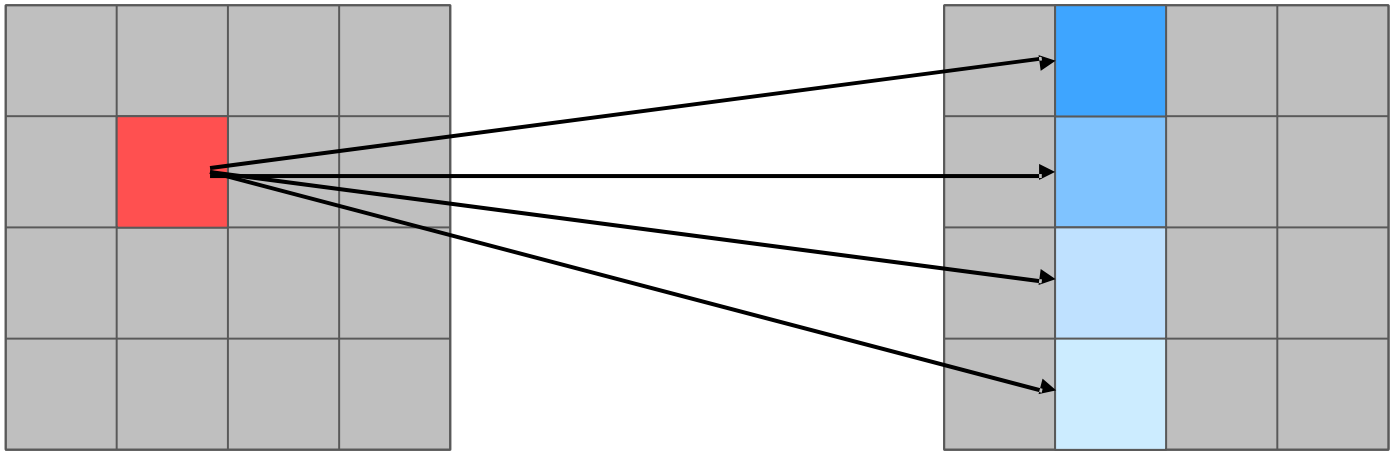


MixColumns

(Diffusions-Schicht: ShiftRows und MixColumn)

- jede Spalte der 4x4 Matrix wird mit Matrix *C* multipliziert im GF(2^8)
- Jede 4-Byte Spalte wird als Vektor interpretiert, Multiplikation mit
- Matrix $C = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}$

MixColumns-Abbildung:
- linear, invertierbar
- Ausbreitung von Differenzen: Änderung in Eingabe, Ausbreitung auf alle 4 Ausgaben



AddRoundKey

- Eingabe: 4x4 Byte Matrix des aktuellen Zustandes
  128 Bit (16 Bytes) Rundenschlüssel
- Operation: bitweise XOR der Eingaben
- Bemerkung: XOR entspricht der Addition im GF(2)
- AddRoundKey ist selbstinvers

KeyExpansion

Rundenschlüssel werden aus dem CipherKey (128, 192, 256 Bit) abgeleitet (siehe Specs)

- Anzahl Rundenschlüssel = 1 + Anzahl der Runden (Nr)
- jeder Rundenschlüssel: 128 Bit

KeyExpansion - Durchführung

the first column of a new round key and the remaining columns,
but each column *i* is defined in terms of the

- corresponding column *i* - 4 of the preceding round key
- the immediately preceding column *i* - 1

the column *i* is computed by directly applying the bitwise *XOR operation*

the preceding column is first transformed by a non-linear function
that is a suitable composition of

- the bytewise application of the substitution function S_{RD}
- a permutation that shifts the positions in a column
- the addition of a round constant

AES Entschlüsselung

AES ist keine Feistel-Chiffre: alle Schichten müssen invertiert und die Rundenschlüssel in umgekehrter Folge angewandt werden

AES Sicherheit

- AES besitzt starke Diffusions- und Konfusions-Eigenschaften
- Bis 2011 waren keine analytischen Angriffe auf AES bekannt, die effizienter als Brute Force sind
- 2011: A. Bogdanov, D. Khovratovich, C. Rechberger:
  - Angriff, der 4 mal schneller als Brute Force ist,
  - ca 2126.1 Operationen erforderlich für AES-128 Bit
  - ca 2189.7 für AES-192 und ca 2254.4 für AES-256
- AES gilt weiterhin als sehr sicher, Verbesserung könnte durch Erhöhung der Rundenzahl erreicht werden

Nächste Woche

Asymmetrische Verfahren