Betriebssysteme und Sicherheit – Sicherheit

Florian Kerschbaum
TU Dresden
Wintersemester 2011/12

Begriffe

Kryptographie: „Geheimschrift“

Nachrichten schreiben ohne das sie von einem Anderen gelesen (verändert) werden können

Kryptanalyse:

Analyse geheimer Nachrichten

Kryptologie:

Kryptographie + Kryptanalyse

Steganographie:

Nachrichten schreiben ohne das ein Anderer bemerkt, dass sie geschrieben wurden

Beispiel 1

NUBSWRJUDSKLH

KRYPTOGRAPHIE

Beispiel 2

SEDDERN

DRESDEN

Substitution

Ersetze Zeichen (Buchstaben) durch Andere

- Nach Regel

  $y = x + 3 \pmod{26}$

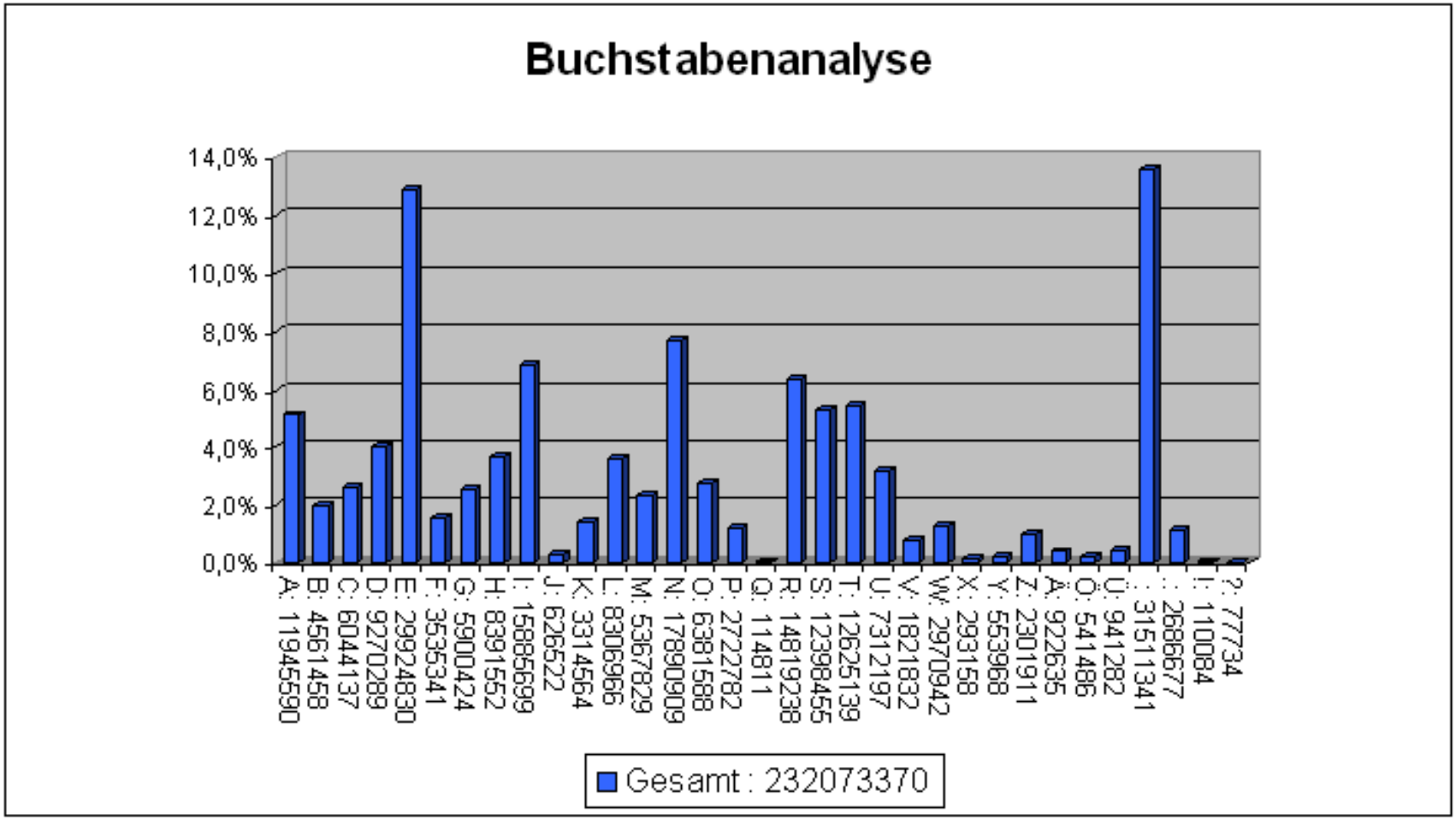
  Caesar‘s Cipher

  – Schlüssel: 3

- Nach Tabelle

  – Schlüssel: Tabelle

Kryptanalyse - Substitution



Quelle: Wikipedia

Transposition

Änderung der Positionen der Zeichen

Beispiel Regel Spaltentransposition:

- Schreibe mit fester Zeilenlänge n
- Lese Spalten
- Schlüssel: n

Beispiel - Spaltentransposition

UNTERNEH

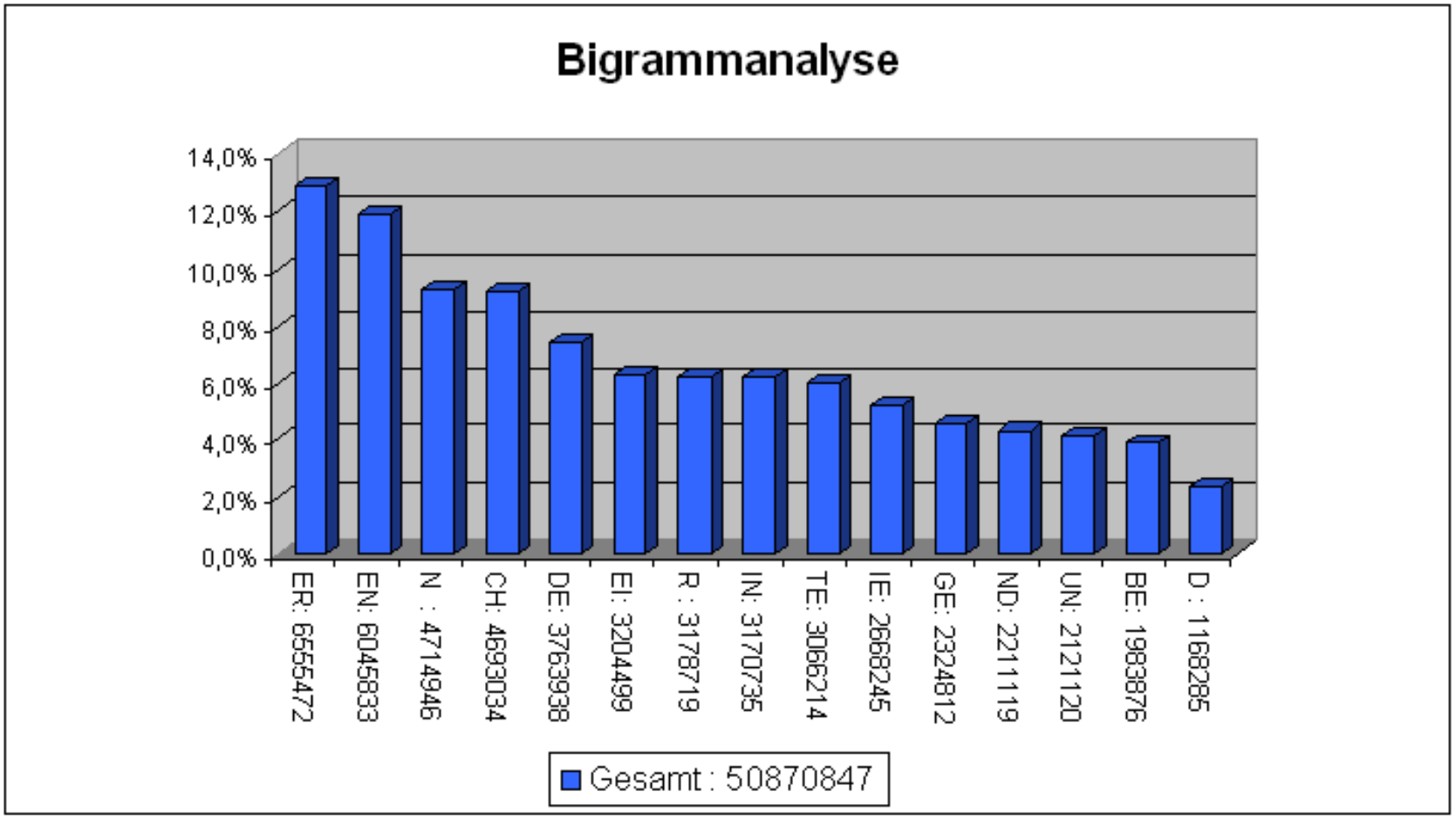
MENSUNDI

TSICHERH

HEITBSC

UMTHNESETNIIESCTRUHBNNESEDRCHIH

Kryptanalyse - Transposition



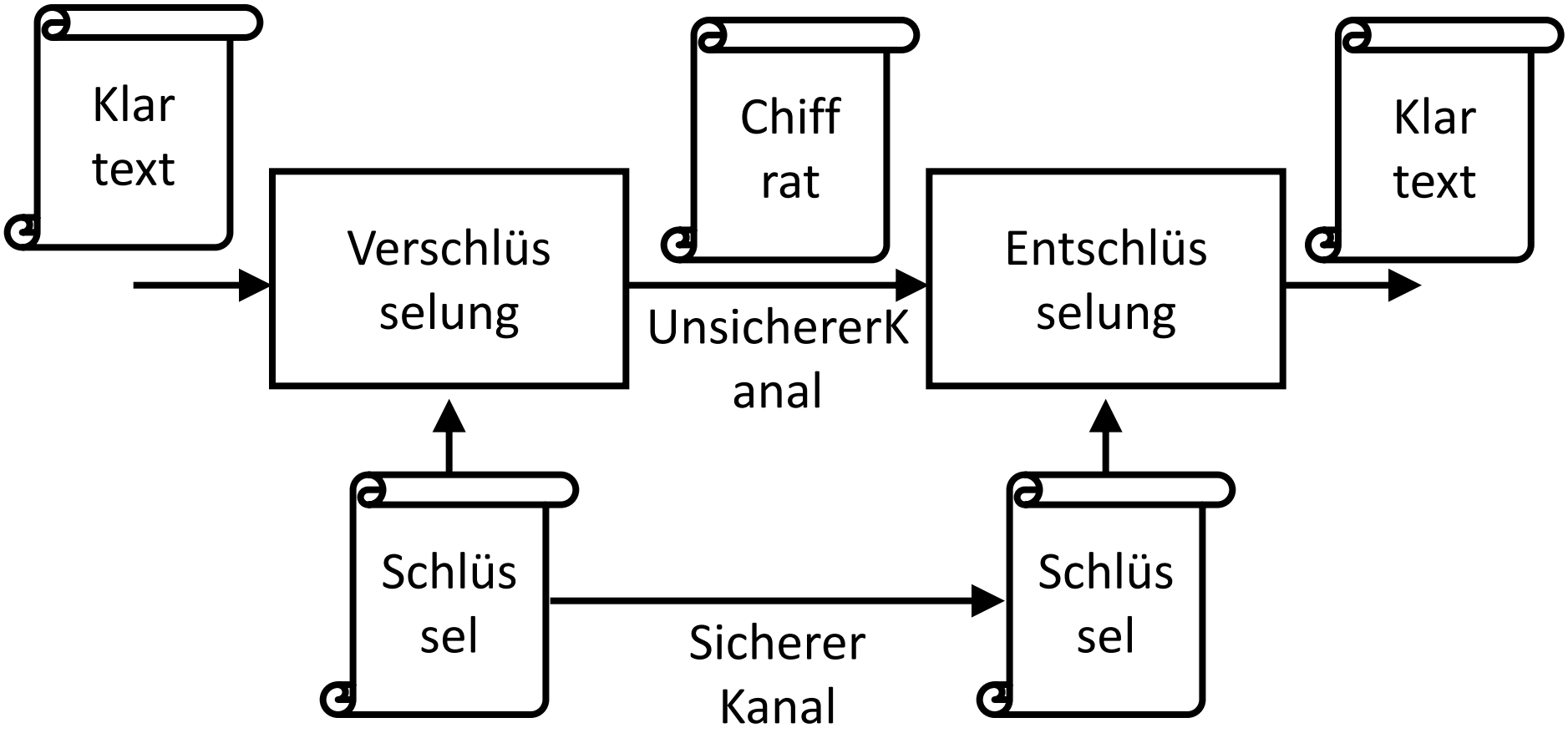
Quelle: Wikipedia

Kryptanalyse - Spaltentransposition

```
UMTHNESETNIIESCTRUHBNNESEDRCHIH
 UMTHN
UMTHNESETNIIESCTRUHBNNESEDRCHIH
  UMTHN
UMTHNESETNIIESCTRUHBNNESEDRCHIH
   UMTHN
UMTHNESETNIIESCTRUHBNNESEDRCHIH
    UMTHN
UMTHNESETNIIESCTRUHBNNESEDRCHIH
```

Moderne Kryptographie

Shannon: Informationstheorie



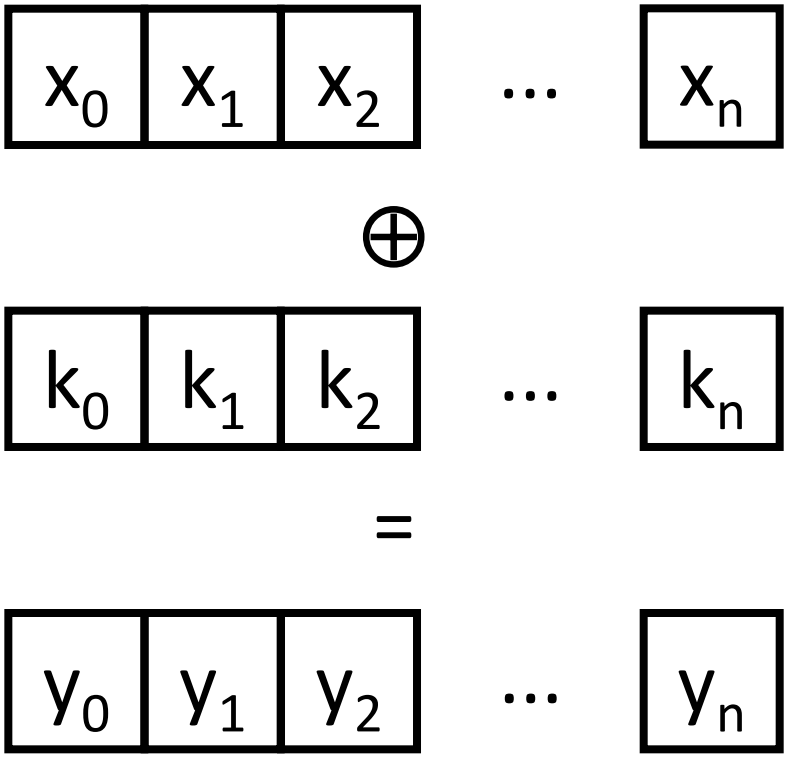
Kerckhoff Prinzip

Die Sicherheit eines Kryptosystems sollte ausschließlich auf der Geheimhaltung des Schlüssels beruhen, niemals aber auf der Geheimhaltung des Systems selber.

One-Time Pad

Regel:

$y_i = x_i + k_i \pmod 2$



Bedingungen

- Schlüssel genauso lang wie Nachricht
- Schüssel gleichverteilt zufällig

Beweisbar Sicher

Chiffrat ist gleichverteilt unabhängig von der Verteilung des Klartexts

$x_i = 0$ mit Wahrscheinlichkeit p ($x_i = 1$: 1-p)

$k_i = 0$ mit Wahrscheinlichkeit 0,5 ($k_i = 1$: 0,5)

$y_i = 0$ mit Wahrscheinlichkeit:

$$p(y_i = 0) = p(x_i = 0)\, p(k_i = 0) + p(x_i = 1)\, p(k_i = 1)$$

$$= 0{,}5p + 0{,}5(1-p)$$

$$= 0{,}5$$

Beweisbar Sicher

Jedes Chiffrat y kann in jeden beliebigen Klartext x entschlüsselt werden

$$k = y \oplus x$$

Chiffrat kann keine Information über Klartext enthalten

Verschlüsselung „abstreitbar“

Problem

Einfache Substitution/Permutation nicht sicher

One-Time Pad ineffizient im sicheren Kanal

Frage: Kann ich mit wenig Kommunikation auf dem sicheren Kanel viel auf dem unsicheren Kanal übertragen?

AES

Vorläufer:

- DES (1977)
- 56 Bit Schlüssel
- Kritik an S-Boxen

Standard 1997 – 2002

128 Bit Schlüssel

Endliche Körper

- Endliche Menge von Elementen x
- Addition / Multiplikation
- Menge „abgeschlossen“
  - Neutrales Element 0 / 1

    $x + 0 = x$

    $x \cdot 1 = x$
  - Inverses

    $x + (-x) = 0$

    $x \cdot x^{-1} = 1$ (ohne 0)

Konstruktion Endlicher Körper

Zu jeder Primzahlpotenz p^a gibt es genau einen Körper

Polynomringe: Polynomdivisionsreste

Polynomringe 2^a

Elemente:

Polynome bis Grad a-1 und

Koeffizienten $\in \{0,1\}$ (mod 2)

Z.B.

$x^6 + x^4 + 1$

Addition:

$(x^6 + x^5 + 1) + (x^6 + x^4 + 1) = x^5 + x^4$

$x^n + x^n = 0$

Polynomringe – Multiplikation

Man nehme ein irreduzibles Polynom vom Grad a

Z.B. $m = x^8+x^4+x^3+x+1$

$p = p_1 \cdot p_2 \bmod m$

$p_1 = x^7 + x^5 + 1$

$p_2 = x^6 + x^4 + x$

$p = x^{13} + x^{11} + x^6 + x^{11} + x^9 + x^4 + x^8 + x^6 + x$

$= x^{13} + x^9 + x^8 + x^4 + x$

Polynomringe – Multiplikation

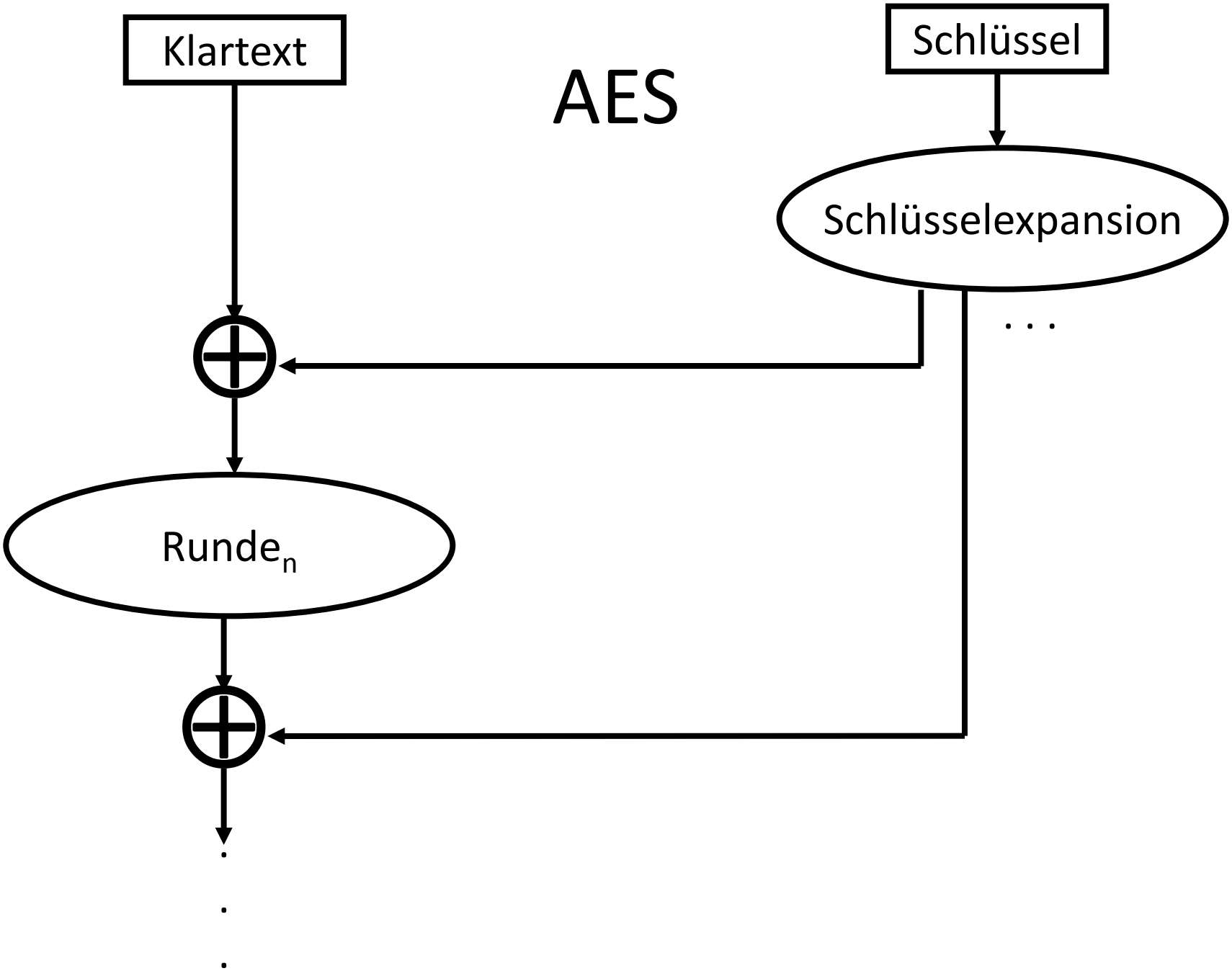
$$p = x^{13}+x^{9}+x^{8}+x^{4}+x \pmod{x^{8}+x^{4}+x^{3}+x+1}$$

$$\underline{- \quad x^{13}+x^{9}+x^{8}+x^{6}+x^{5} \qquad (x^{5} \cdot (x^{8}+x^{4}+x^{3}+x+1))}$$

$$x^{6}+x^{5}+x^{4}+x$$

$$(x^{7}+x^{5}+1)\cdot(x^{6}+x^{4}+x) \bmod (x^{8}+x^{4}+x^{3}+x+1) =$$

$$= x^{6}+x^{5}+x^{4}+x$$


Klartext
Schlüssel
AES
Schlüsselexpansion
. . .
Runde$_n$

AES Runden

10 Runden

Jede Runde besteht aus

- SubBytes
- ShiftRows
- MixColumns

AES State

128 Bit = 16 Byte: 0, 1, 2, 3, ...

Spaltenweise 4x4 Matrix

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

AES – SubBytes

Es sei Byte b ein Element im Polynomring modulo x8+x4+x3+x+1

Berechne b^{-1} (für $b \neq 0$)

Konvertiere b^{-1} zu Bits: $b_7b_6b_5b_4b_3b_2b_1b_0$

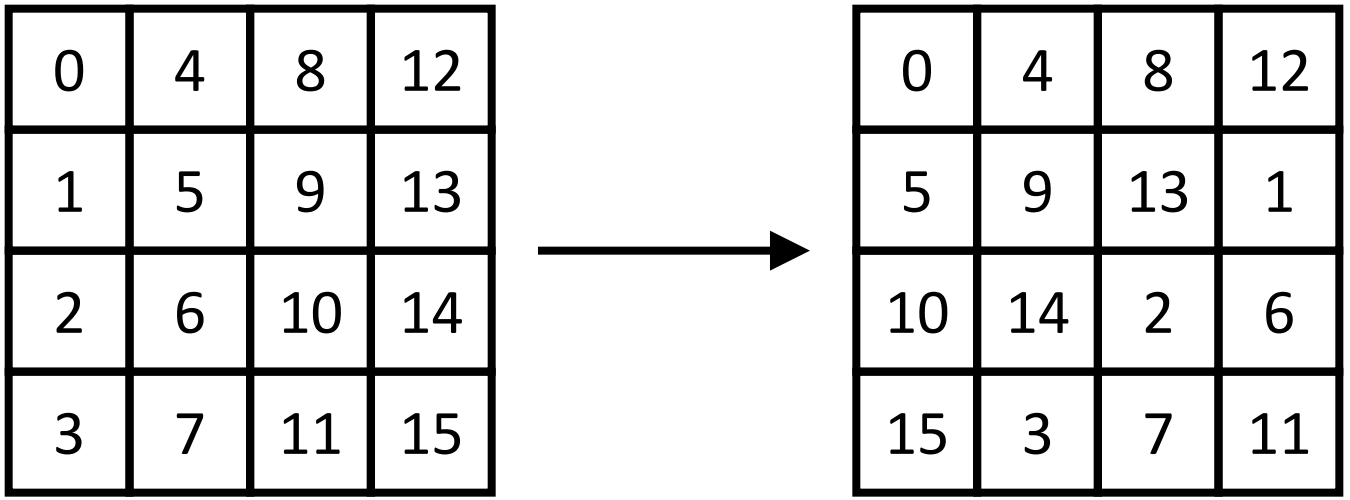
Definiere c ($c_7c_6c_5c_4c_3c_2c_1c_0$) = 01100011

Berechne (i=0 ... 7)

$a_i = b_i + b_{i+4} + b_{i+5} + b_{i+6} + b_{i+7} + c_i \pmod 2$

Konvertiere a_i zu Byte

AES - ShiftRows



AES – MixColumns

Es sei jedes Byte b_i einer Spalte ein Element im Polynomring

Multipliziere

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Wiederhole für jede Spalte

AES – Schlüsselexpansion

Schlüssel ist 128 Bit = 4 Worte (a 4 Bytes)

Es w_i (i=0 ... 3) das i-te Wort

Beginne mit Ausgangsschlüssel

Berechne w_i (i=4 ... 43)

$i \neq 0 \pmod 4$: $w_i = w_{i-4} \oplus w_{i-1}$

$i = 0 \pmod 4$: $w_i = w_{i-4}$

$\oplus$ SubBytes($w_{i-1}[1]$, $w_{i-1}[2]$, $w_{i-1}[3]$, $w_{i-1}[0]$)

$\oplus$ (0100000) << ((i-4)/4)

AES – Entschlüsselung

Jede Operation (XOR, MixColumns, ShiftRows, SubBytes) ist invertierbar

Für Operationen in umgekehrter Reihenfolge aus

Schlüsselexpansion bleibt

Erkenntnis

Substitution ist unsicher

Permutation ist unsicher

Schlüsselwiederverwendung bei XOR ist unsicher

ABER

Wiederhole alles oft genug und es kann als sicher gelten

Implementierung

AES ist effizient

AES kann gut in Hardware implementiert werden

S-Boxen

1. Vorberechnung und Speicherung
2. Berechnung und Speicherung bei erster Ausführung
3. Immer Berechnung

Problem

AES ist eine Blockchiffre

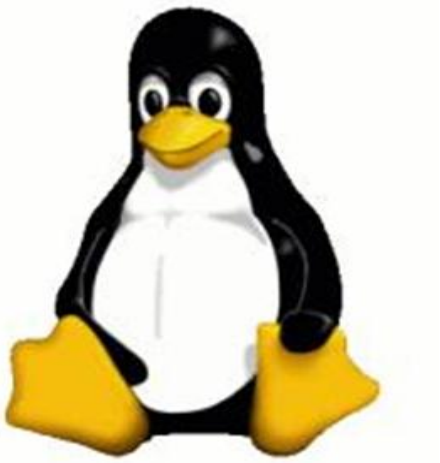
- 128 Bit (Klartext) x 128 Bit (Schlüssel) $\rightarrow$ 128 Bit (Chiffrat)

Wie verschlüsselt man einen Strom von Daten (> 128 Bit) ?

Erste Idee: Teile in Blöcke ein und verschlüseele jeden Block einzeln

(ECB: Electronic Code Book)

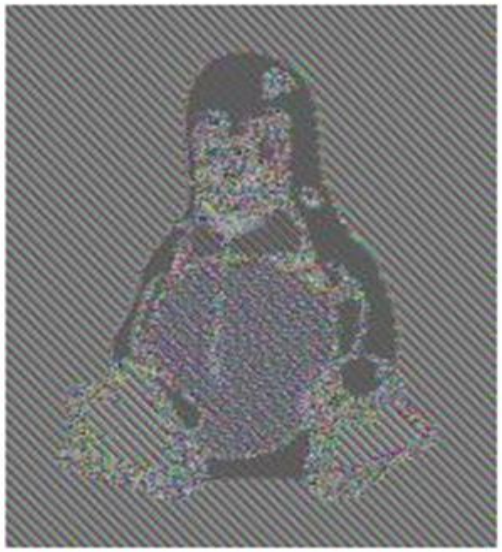
Beispiel



Original

Beispiel

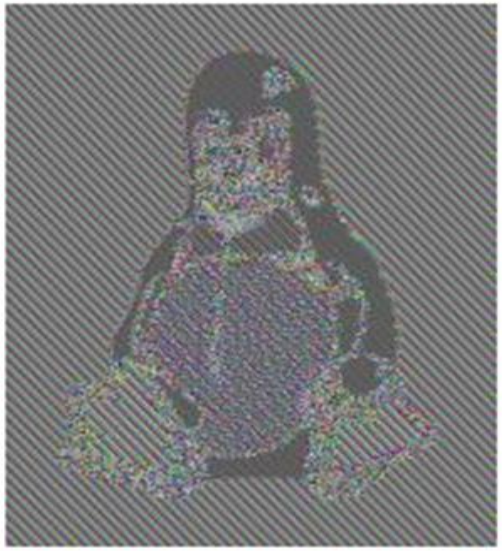
Original



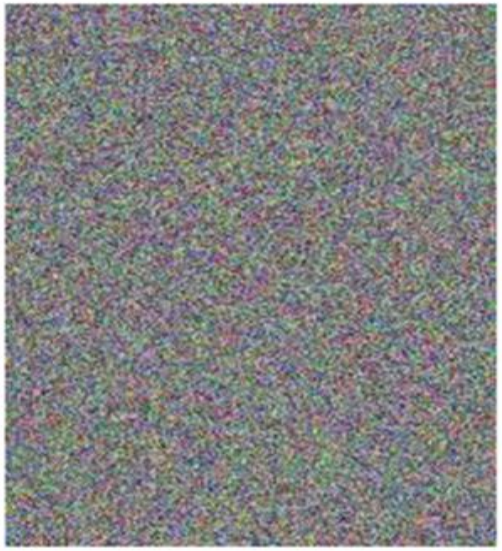
Encrypted using ECB mode

Beispiel

Original



Encrypted using ECB mode



Encrypted using other modes

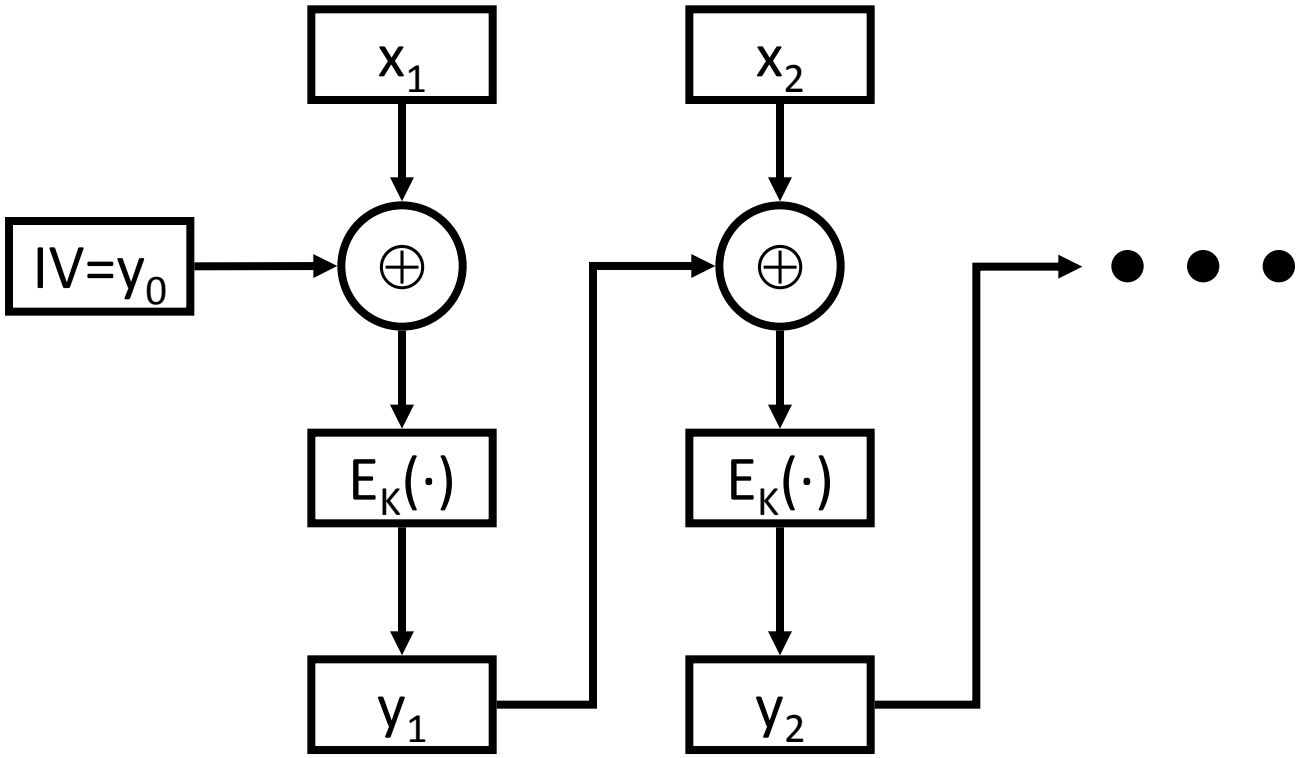
Lösung

Verkette Blöcke miteinander, so dass der letzte Block von allen vorherigen abhängt

(CBC: Cipher Block Chaining)

$$y_i = E(y_{i-1} \oplus x_i)$$

CBC – Verschlüsselung



CBC – Entschlüsselung