Symmetrische Kryptographie
AES

yanosz

Chaos Computer Club Cologne e.V.
https://koeln.ccc.de

12. Oktober 2015

Hinweis

Dieser Foliensatz basiert auf den Folien zur Vorlesung Cryptography-I, Wintersemester 2012 / 2013, B-IT, Bonn
Einzelne Folien wurden mit freundlicher Genehmigung übernommen.

Inhalt

Symmetrische Kryptographie

- Gleicher Schlüssel k zum Verschlüsseln und Entschlüsseln
- Für einen Cleartext $x \in \mathbb{B}^*$ und Ciphertext $y \in \mathbb{B}^*$:

$$x \in \mathbb{B}^* \Rightarrow$$
$$x = \underbrace{0101011\ldots010101}_{\text{Beliebig oft - Vektor}}$$

$$\text{Enc}_k : \mathbb{B}^* \to \mathbb{B}^*, \text{Enc}_k(x) = y$$
$$\text{Dec}_k : \mathbb{B}^* \to \mathbb{B}^*, \text{Dec}_k(y) = x$$
$$\text{Dec}_k(\text{Enc}_k(x)) = x$$

Symmetrische Kryptographie — Beispiel: XOR / **O**ne-**T**ime-**P**ad

- $\mathrm{Enc}_k(x) := x \oplus k$
- $\mathrm{Dec}_k(y) := y \oplus k$
- Key $k = 0101$, Nachricht: $x = 1100$

$$\mathrm{Enc}_k(x) = \mathrm{Enc}_{0101}(1100) = 1100 \oplus 0101 = 1001 = y$$
$$\mathrm{Dec}_k(y) = \mathrm{Dec}_{0101}(1001) = 1001 \oplus 0101 = 1100 = x$$

Erinnerung:
$0 \oplus 0 = 0, 0 \oplus 1 = 1$
$1 \oplus 0 = 1, 1 \oplus 1 = 0$

Advanced **E**ncryption **S**tandard

- Verschiedene Schlüssellängen: $k \in \{\mathbb{B}^{128}, \mathbb{B}^{192}, \mathbb{B}^{256}\}$
- 128 Bit Blockgröße: $\mathrm{Enc}_k, \mathrm{Dec}_k : \mathbb{B}^{128} \to \mathbb{B}^{128}$
- Geschichte:
  - 1997 Wettbewerb: Advanced Encryption Standard — Nachfolger Data Encryption Standard (**DES**)
  - 2001 Gewinner: Rijndael (J. Daemen, V. Rijmen)
  - 2011 Key-Recovery (A. Bogdanov, D. Khovratovich, C. Rechberger):

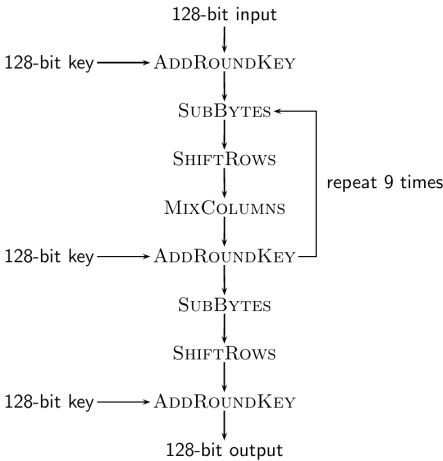
k (Bit)	Operationen
128	$2^{126.2}$
192	$2^{190.2}$
256	$2^{254.6}$

Wie Rijndael funktioniert (128-Bit)

- Darstellung: 4×4 Byte ($a_i \in \mathbb{B}^8$)

a_0	a_1	a_2	a_3
a_4	a_5	a_6	a_7
a_8	a_9	a_{10}	a_{11}
a_{12}	a_{13}	a_{14}	a_{15}

- 10 Runden
  - AddRoundKey — $\oplus$ Rundenschlüssel
  - SubBytes — Byteweise ersetzen
  - ShiftRows — Zeilen verschieben
  - MixColumns —Spaltenweise mischen

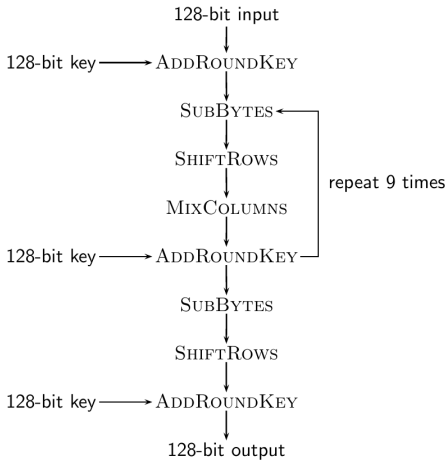


AddRoundKey - Wie Rijndael funktioniert (128-Bit)

Addiere den 128-Bit (Round)-Key k - byteweise:

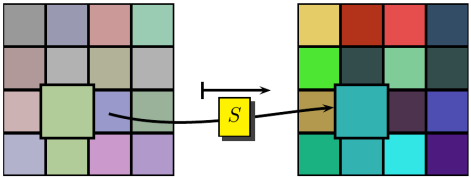
$$\begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{bmatrix} \oplus \begin{bmatrix} k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{bmatrix} =$$

$$\begin{bmatrix} x_0 \oplus k_0 & x_1 \oplus k_1 & x_2 \oplus x_2 & x_3 \oplus k_3 \\ x_4 \oplus k_4 & x_5 \oplus k_5 & x_6 \oplus x_6 & x_7 \oplus k_7 \\ x_8 \oplus k_8 & x_9 \oplus k_9 & x_{10} \oplus x_{10} & x_{11} \oplus k_{11} \\ x_{12} \oplus k_{12} & x_{13} \oplus k_{13} & x_{14} \oplus x_{14} & x_{15} \oplus k_{15} \end{bmatrix}$$

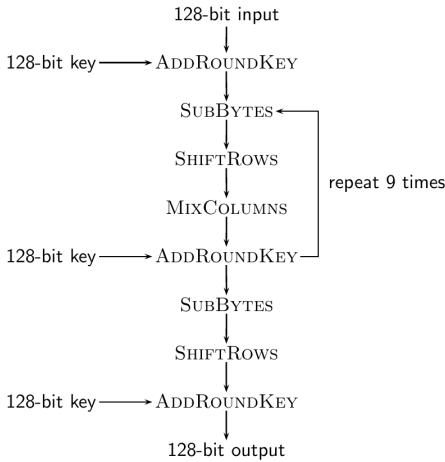


SubBytes (S-Box)- Wie Rijndael funktioniert (128-Bit)

Ersetze Byte für Byte nach Lookup-Tabelle:

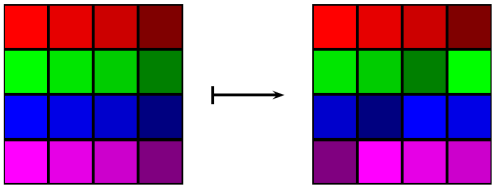


x	$f(x)$
0x00	0x63
0x01	0x7C
0x02	0x77
...	...

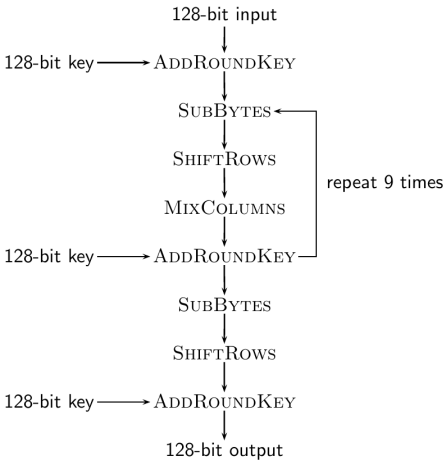


ShiftRows - Wie Rijndael funktioniert (128-Bit)

Verschiebe Zeilenweise:

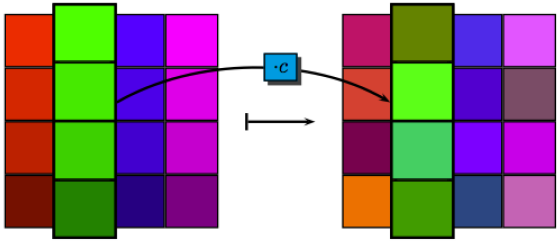


$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \mapsto \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{11} & a_{12} & a_{13} & a_{10} \\ a_{22} & a_{23} & a_{20} & a_{21} \\ a_{33} & a_{30} & a_{31} & a_{32} \end{bmatrix}$$



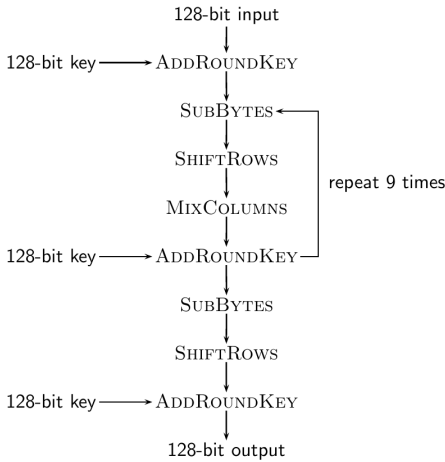
MixColumns - Wie Rijndael funktioniert (128-Bit)

Ersetze Spaltenweise:



Ersetzung:

$f : \mathbb{B}^4 \times \mathbb{B}^4 \times \mathbb{B}^4 \times \mathbb{B}^4 \to \mathbb{B}^4 \times \mathbb{B}^4 \times \mathbb{B}^4 \times \mathbb{B}^4$



Warum ist AES sicher / vertrauenswürdig?

- Widerstand gegen:
  - Linear Cryptoanalysis
  - Differential Cryptoanalysis
- SubBytes (S-Boxes), MixColumns:
  - Operationen in GF(2^8) $= \mathbb{F}_{256}$ bzw. $\mathbb{F}_{256}[z]/(z^4 + 1)$
  - Nicht-Linear (S-Box)
  - Offener Wettbewerb, keine NSA-Vorgabe
- Für Quantencomputer kein effizienter Algorithmus bekannt.

$\mathbb{F}_{256}$ und $\mathbb{F}_{256}[z]/(z^4+1)$

GF(2^8) bzw. $\mathbb{F}_{256}$ — Körper mit 256 Elementen

- $\underbrace{(1,0,0,0,0,0,0,1)}_{\texttt{F1 bzw. } x^7+1} + \underbrace{(1,0,0,0,0,0,0,0)}_{\texttt{F0 bzw. } x^7} = \underbrace{(0,0,0,0,0,0,0,1)}_{\texttt{01 bzw. } x^0=1}$
- $(x^7+1)\cdot x^7 = x^{14}+x^7 \equiv x^4+x^3+x \pmod{x^8+x^4+x^3+x+1}$ in $\mathbb{Z}_2[x]$ bzw. $\texttt{F1}\cdot\texttt{F0} \equiv \texttt{1A}$

$\mathbb{F}_{256}[z]/(z^4+1)$ — Polynom-Ring über $\mathbb{F}_{256}$

- Ring mit $256^4 = 2^{32}$ Elementen
- z^4+1 ist nicht irreduzibel $\Rightarrow$ Kein Körper

SubBytes in $\mathbb{F}_{256}$

Definition in $Z_2[z]$

$S : \mathbb{F}_{256} \to \mathbb{F}_{256}$

$z \mapsto 05z^{254} + 09z^{253} + F9z^{251} + 25z^{247} + F4z^{239} + 01z^{223} + B5z^{191} + 8Fz^{127} + 63$

Matrix-Schreibweise

$$z \mapsto z^{-1} \widehat{=} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \mapsto \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

MixColumns

Definition in $\mathbb{F}_{256}[z]/(z^4+1)$

Seien $R = \mathbb{F}_{256}[z]/(z^4+1)$, $d = \mathtt{0B}z^3 + \mathtt{0D}z^2 + \mathtt{09}z + \mathtt{0E} \in R$.

$$\begin{aligned} \text{MixColumns} : R &\to R \\ a &\mapsto a \cdot d \end{aligned}$$

Matrixschreibweise in $\mathbb{F}_{256}$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} \mathtt{02} & \mathtt{03} & \mathtt{01} & \mathtt{01} \\ \mathtt{01} & \mathtt{02} & \mathtt{03} & \mathtt{01} \\ \mathtt{01} & \mathtt{01} & \mathtt{02} & \mathtt{03} \\ \mathtt{03} & \mathtt{01} & \mathtt{01} & \mathtt{02} \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$