AES

Jens Kubieziel
jens@kubieziel.de

Friedrich-Schiller-Universität Jena
Fakultät für Mathem atik und Informatik

07. Dezember 2009

Outline

Wichtige Algorithmen im 20. Jahrhundert

- ADFGVX

Wichtige Algorithmen im 20. Jahrhundert

- ADFGVX
- Enigma

Wichtige Algorithmen im 20. Jahrhundert

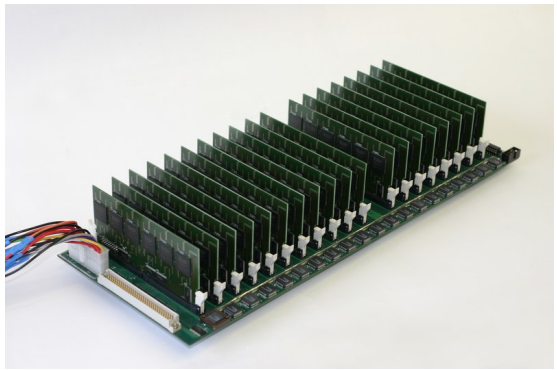
- ADFGVX
- Enigma
- DES

Wichtige Algorithmen im 20. Jahrhundert

- ADFGVX
- Enigma
- DES
- und viele weitere

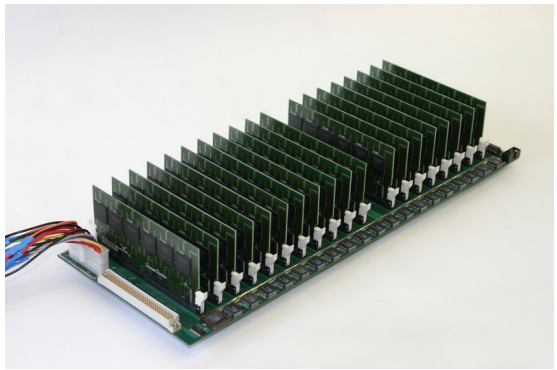
DES

- Data Encryption Standard, symmetrischer Algorithmus mit 56 Bit Schlüssellänge



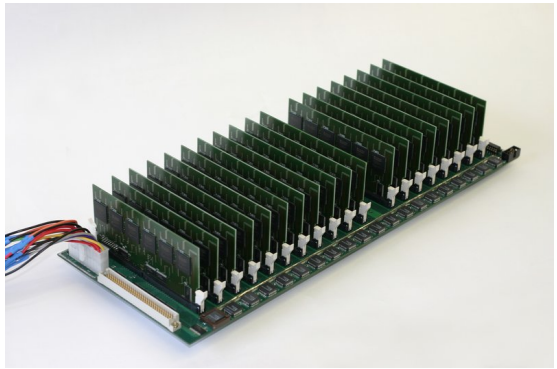
DES

- Data Encryption Standard, symmetrischer Algorithmus mit 56 Bit Schlüssellänge
- Standard seit 1976 (seit 2004 nicht mehr empfohlen)



DES

- Data Encryption Standard, symmetrischer Algorithmus mit 56 Bit Schlüssellänge
- Standard seit 1976 (seit 2004 nicht mehr empfohlen)
- Durchprobieren aller Schlüssel dauert sechs Tage



Standardisierung von AES

Anforderungen

- symmetrische Blockchiffre
- Blocklänge 128 Bit, Schlüssellänge 128, 192 und 256 Bit
- leichte Umsetzung in Hard- und Software
- überdurchschnittliche Performance
- sicher gegen bekannte Angriffe
- frei von patentrechtlichen Ansprüchen

Standardisierung von AES

Rijndael

- Blockchiffre mit variabler Schlüssel- und Blocklänge

Standardisierung von AES

Rijndael

- Blockchiffre mit variabler Schlüssel- und Blocklänge
- von JOAN DAEMEN und VINCENT RIJMEN

Standardisierung von AES

Rijndael

- Blockchiffre mit variabler Schlüssel- und Blocklänge
- von JOAN DAEMEN und VINCENT RIJMEN
- FIPS 197 vom Oktober 2000

Outline

Der AES-Algorithmus

Grundlagen

- Block in 16 Teile zu 8 Bit=1 Byte aufgeteilt

Der AES-Algorithmus

Grundlagen

- Block in 16 Teile zu 8 Bit=1 Byte aufgeteilt
- Operation in $\mathbb{F}_{2^8} = \mathbb{F}_2[x]/g\mathbb{F}_2[x]$ mit dem irreduziblen Polynom

$$g = x^8 + x^4 + x^3 + x + 1$$

Der AES-Algorithmus

Grundlagen

- Block in 16 Teile zu 8 Bit=1 Byte aufgeteilt
- Operation in $\mathbb{F}_{2^8} = \mathbb{F}_2[x]/g\mathbb{F}_2[x]$ mit dem irreduziblen Polynom

$$g = x^8 + x^4 + x^3 + x + 1$$

- Byte $b_7b_6b_5b_4b_3b_2b_1b_0$ repräsentiert durch ein Polynom aus $\mathbb{F}_{2^8}$

Der AES-Algorithmus

Grundlagen

- Block in 16 Teile zu 8 Bit=1 Byte aufgeteilt
- Operation in $\mathbb{F}_{2^8} = \mathbb{F}_2[x]/g\mathbb{F}_2[x]$ mit dem irreduziblen Polynom

$$g = x^8 + x^4 + x^3 + x + 1$$

- Byte $b_7b_6b_5b_4b_3b_2b_1b_0$ repräsentiert durch ein Polynom aus $\mathbb{F}_{2^8}$
- Schreibweise als 4×4-Matrix:

$$M = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \qquad a_{ij} \in \mathbb{F}_{2^8}$$

Der AES-Algorithmus

Rundentransformation

Insgesamt gibt es neun Runden mit den Operationen:

1. `AddRoundKey`

Der AES-Algorithmus

Rundentransformation

Insgesamt gibt es neun Runden mit den Operationen:

1. `AddRoundKey`
2. `SubBytes`

Der AES-Algorithmus

Rundentransformation

Insgesamt gibt es neun Runden mit den Operationen:

1. `AddRoundKey`
2. `SubBytes`
3. `ShiftRows`

Der AES-Algorithmus

Rundentransformation

Insgesamt gibt es neun Runden mit den Operationen:

1. `AddRoundKey`
2. `SubBytes`
3. `ShiftRows`
4. `MixColumns`

Der AES-Algorithmus

Rundentransformation

Insgesamt gibt es neun Runden mit den Operationen:

1. `AddRoundKey`
2. `SubBytes`
3. `ShiftRows`
4. `MixColumns`

Der AES-Algorithmus

Rundentransformation

Insgesamt gibt es neun Runden mit den Operationen:

1. `AddRoundKey`
2. `SubBytes`
3. `ShiftRows`
4. `MixColumns`

und eine Abschlussrunde *ohne* die obige Operation `MixColumns`.

Der AES-Algorithmus

Rundentransformation

Insgesamt gibt es neun Runden mit den Operationen:

1. `AddRoundKey`
2. `SubBytes`
3. `ShiftRows`
4. `MixColumns`

und eine Abschlussrunde *ohne* die obige Operation `MixColumns`. Weiterhin werden für jede Runde separate Schlüssel errechnet.

Der AES-Algorithmus

SubBytes

Sei $a_{ij} \in \mathbb{F}_{2^8}$ aus der Matrix M und $S = g \circ f$ mit

$$f(a) = \begin{cases} a^{-1} & x \neq 0 \\ 0 & x = 0 \end{cases} \text{ und}$$

Der AES-Algorithmus

SubBytes

Sei $a_{ij} \in \mathbb{F}_{2^8}$ aus der Matrix M und $S = g \circ f$ mit

$$f(a) = \begin{cases} a^{-1} & x \neq 0 \\ 0 & x = 0 \end{cases} \text{ und}$$

$$g \begin{pmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a^7 \\ a^6 \\ a^5 \\ a^4 \\ a^3 \\ a^2 \\ a^1 \\ a^0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Der AES-Algorithmus

ShiftRows

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

Der AES-Algorithmus

MixColumns

Jede Eingabespalte $a_{i.}$ wird mit einem Polynom
$c(x) = 03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02$ modulo $x^4 + 1$ multipliziert.

Der AES-Algorithmus

MixColumns

Jede Eingabespalte $a_{i.}$ wird mit einem Polynom $c(x) = 03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02$ modulo $x^4 + 1$ multipliziert.
Schreibweise als Matrixmultiplikation:

$$\begin{pmatrix} b_{i0} \\ b_{i1} \\ b_{i2} \\ b_{i3} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \end{pmatrix}$$

Der AES-Algorithmus

AddRoundKey

Addition der Matrix M mit dem Rundenschlüssel modulo 2

Outline

Der effektivste Weg

Mögliche Angriffe

- Lineare und differenzielle Kryptoanalyse

Mögliche Angriffe

- Lineare und differenzielle Kryptoanalyse
- XSL-Angriff

Mögliche Angriffe

- Lineare und differenzielle Kryptoanalyse
- XSL-Angriff
- Darstellung als Kettenbruch

Mögliche Angriffe

Darstellung als Kettenbruch

$$a_{i,j}^{(6)} = K + \sum_{\substack{e_5 \in \mathcal{E} \\ d_5 \in \mathcal{D}}} \cfrac{C}{K^* + \sum_{\substack{e_4 \in \mathcal{E} \\ d_4 \in \mathcal{D}}} \cfrac{C}{K^* + \sum_{\substack{e_3 \in \mathcal{E} \\ d_3 \in \mathcal{D}}} \cfrac{C}{K^* + \sum_{\substack{e_2 \in \mathcal{E} \\ d_2 \in \mathcal{D}}} \cfrac{C}{K^* + \sum_{\substack{e_1 \in \mathcal{E} \\ d_1 \in \mathcal{D}}} \cfrac{C}{K^* + p_*^*}}}}}$$