Advanced Encryption Standard (AES) Rijndael

Hans Löhr

Hans.Loehr@informatik.stud.uni-erlangen.de

Küchenkolloquium der Henke–WG

18. Juli 2003

Friedrich–Alexander–Universität Erlangen–Nürnberg

Überblick

- Kryptografie: symmetrische Verfahren
- „Geschichte” von AES/Rijndael
- Der Algorithmus Rijndael
- Algebraische Aspekte
- Chosen–Plaintext Attacken

Kryptografie

- Ziele:
  - Vertraulichkeit
  - Integrität
  - Authentifikation
- Verfahren:
  - Symmetrische Verschlüsselung (secret keys)
  - Asymmetrische Verschlüsselung (public/private keys)
  - Digitale Signaturen, ...

Symmetrische Verfahren

- [Stromchiffren (Bitstroeme)]
- Blockchiffren (z.B. 64 bit, 128 bit, 256 bit)
- Geheimer Schlüssel: zum Ver– und Entschlüsseln
- Schlüsselaustausch: meist durch public–key Verfahren
- Verschiedene Modi:
  - **E**lectronic **C**odebook **M**ode (**ECB**)
  - **C**ipher **B**lock **C**haining Mode (**CBC**)
  - **C**ipher **F**eed**B**ack Mode (**CFB**)
  - **O**utput **F**eed**B**ack Mode (**OFB**)
  - ...

Angriffe auf Blockalgorithmen

- **Ciphertext Only Attack**
- **Known Plaintext Attack**
- **Chosen Plaintext Attack**
- Adaptive Chosen Plaintext Attack
- Chosen Ciphertext Attack
- „Chosen Key” Attack
- Rubberhose Attack / Purchase Key Attack

AES - Geschichte: DES

- **DES** (**D**ata **E**ncryption **S**tandard)
  - 70er Jahre, offizieller US–Standard seit 1976
  - 64 Bit Blöcke
  - 64 Bit Keys; aber: 8 Bit Checksumme ⇒ 56 Bit effektiv
  - Biham, Shamir 1990: Differentielle Kryptanalyse
  - Matsui 1993: Lineare Kryptanalyse
  - **Schlüssellänge zu gering!**
- DES zu schwach ⇒ Triple DES (3DES)
  - Langsam!
- Neuer Standard *musste* her.

AES - Geschichte: DES–Nachfolger

- NIST: National Institute of Standards and Technology
- Auswahlprozess: Öffentlicher Wettbewerb
- 5 Finalisten:
  - MARS
  - SERPENT
  - TWOFISH
  - RC6
  - RIJNDAEL
- Entscheidung (NIST): Rijndael

Rijndael [1]

- Autoren: Vincent Rijmen und Joan Daemen
- Blocklänge 128, 192 oder 256 Bit
- Schlüssellänge 128, 192 oder 256 Bit
- Schlüssel–/Blocklängen frei kombinierbar
  ($\Rightarrow$ mehr Spass bei der Implementierung)
- Rundenbasiert (10, 12, 14 Runden; je nach Block–/Schlüssellänge)
- Rundenschlüssel werden erzeugt („Key Schedule”)

Rijndael [2]

- Darstellung von Blöcken/Schlüsseln:
  Rechteckige Schemata: 1 Kästchen $\leftrightarrow$ 1 Byte

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

Rijndael [3]

- Rundenfunktion:
  - **ByteSub**: Bytes werden ersetzt (Tabelle mit 256 Einträgen)
  - **ShiftRow**: Zyklischer byte–weiser shift der „Zeilen”
  - **MixColumn**: „Durchmischt” die „Spalten”
  - **AddRoundKey**: XOR mit Rundenschluessel (byte–weise)
- Vor der ersten Runde: AddRoundKey
- Letzte Runde: MixColumn entfällt
- **Non–linear Layer**: ByteSub
- **Linear Mixing Layer**: ShiftRow und MixColumn
- **Key Addition Layer**: AddRoundKey

Rijndael: ByteSub [1]

- Bytes: Elemente von $\mathbb{F}_{2^8}$

$$\mathbb{F}_{2^8} \simeq \mathbb{F}_2[x]/(m(x))$$

$$\text{wobei } m(x) := x^8 + x^4 + x^3 + x + 1$$

- Jedes Bit entspricht einem Koeffizienten
- Beispiel: 0xD4 $\leftrightarrow$ $1101\ 0100$ $\leftrightarrow$ $x^7 + x^6 + x^4 + x^2$
- Addition: XOR, bitweise
- Multiplikation: modulo $m(x)$
- Invertieren ($\mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$):

$$x \mapsto \begin{cases} 0 & \text{für } x = 0 \\ x^{-1} & \text{sonst} \end{cases}$$

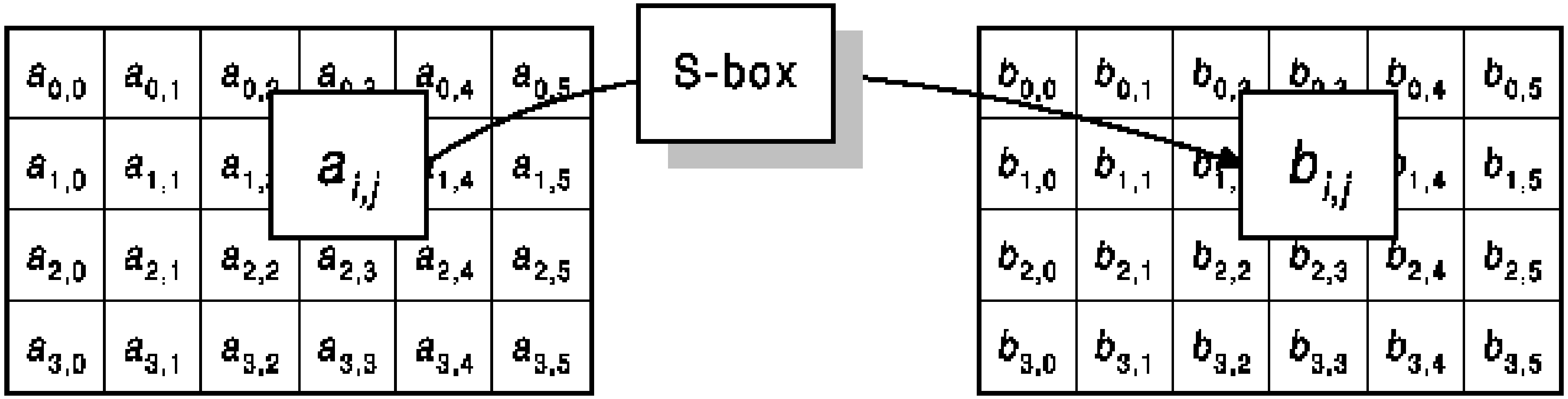
Rijndael: ByteSub [2]

- Affine Transformation:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

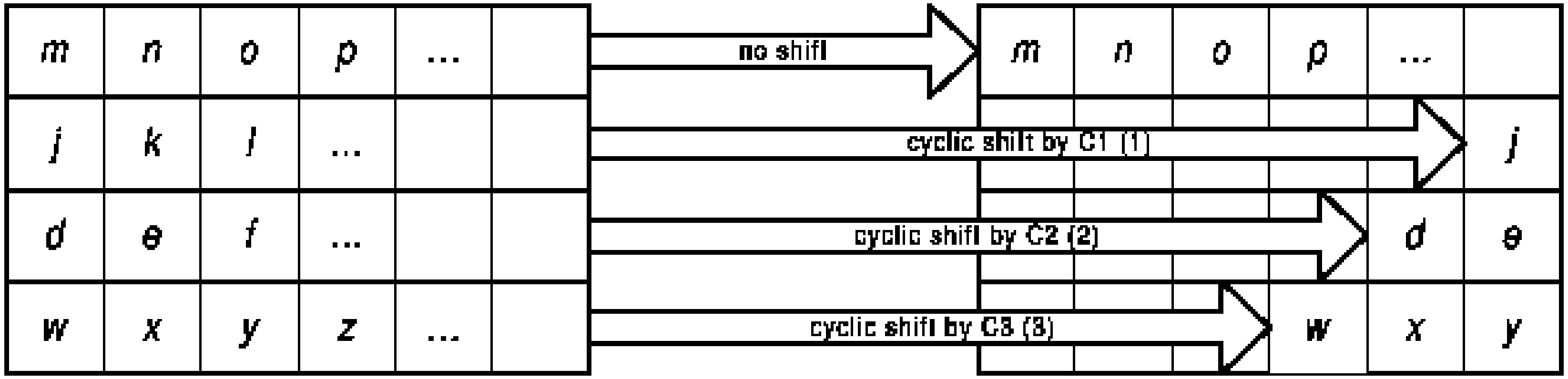
Rijndael: ByteSub [3]

- Bytes x_i werden ersetzt durch $y_i = A * x_i^{-1} + b$ (in ${\mathbb{F}_2}^8$)



Rijndael: ShiftRow

- Zyklischer Links–Shift der Bytes in jeder Zeile (Anzahl abhängig von der Blockgröße)



Rijndael: MixColumn [1]

- Spalten: Polynome über $\mathbb{F}_{2^8}$ vom Grad 3
  $\Rightarrow$ rechnen in $\mathbb{F}_{2^8}[x]$
- Multiplikation modulo $M(x) := x^4 + 1$ mit

$$c(x) :=' 03' \cdot x^3 +' 01' \cdot x^2 +' 01' \cdot x +' 02'$$

- $M(x)$ ist nicht irreduzibel in Charakteristik 2
  ($x^4 + 1 = (x+1)^4$)
- $c(x)$ ist teilerfremd zu $M(x)$,
  also invertierbar modulo $M(x)$:

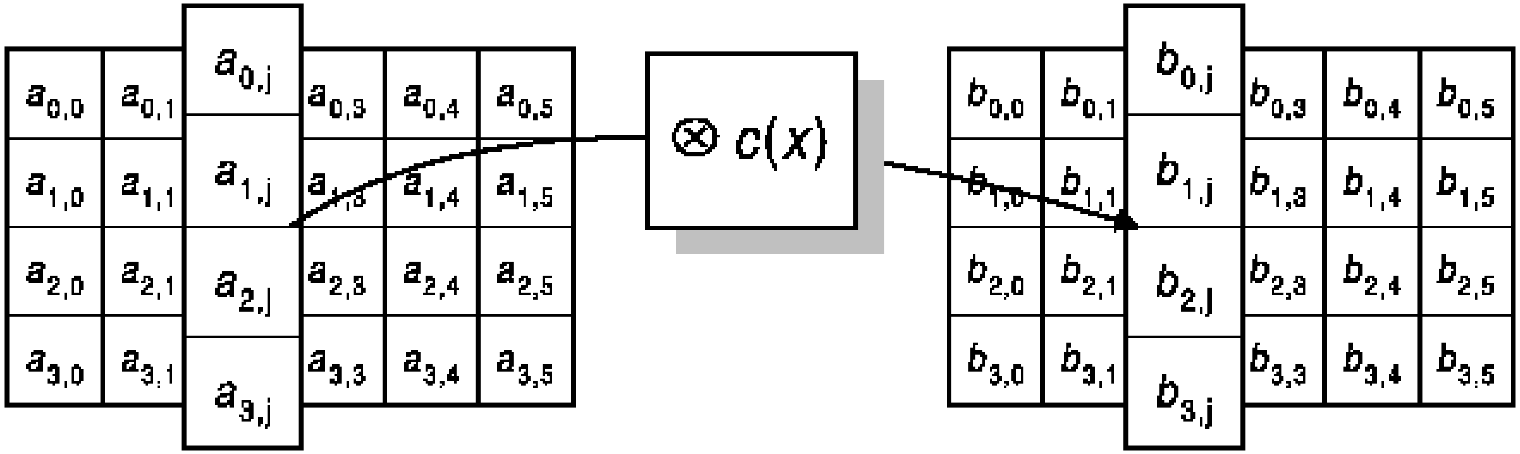
$$d(x) := c(x)^{-1} \text{ mod } M(x) =' 0B' \cdot x^3 +' 0D' \cdot x^2 +' 09' \cdot x +' 0E'$$

Rijndael: MixColumn [2]

- Multiplikation in Matrixschreibweise:

$$B = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot A \text{ über } \mathbb{F}_{2^8}$$

- Operation erfolg Spaltenweise



Rijndael: AddRoundKey

- XOR mit dem Rundenschlüssel

$$\begin{array}{|c|c|c|c|c|c|}\hline a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & a_{0,4} & a_{0,5} \\ \hline a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ \hline a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ \hline a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|c|c|}\hline k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} & k_{0,4} & k_{0,5} \\ \hline k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} & k_{1,4} & k_{1,5} \\ \hline k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} & k_{2,4} & k_{2,5} \\ \hline k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} & k_{3,4} & k_{3,5} \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|c|}\hline b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,4} & b_{0,5} \\ \hline b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} \\ \hline b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & b_{2,5} \\ \hline b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5} \\ \hline \end{array}$$

Rijndael: Kryptanalyse

- Sicher gegen alle bekannten Attacken
- Insbesondere: differentielle und lineare Kryptanalyse
- Attacken gegen „rundenreduzierte” Versionen
  - SQUARE–Attacke
  - Chosen Plaintext Attacke

SQUARE–Attacke: Λ–Mengen

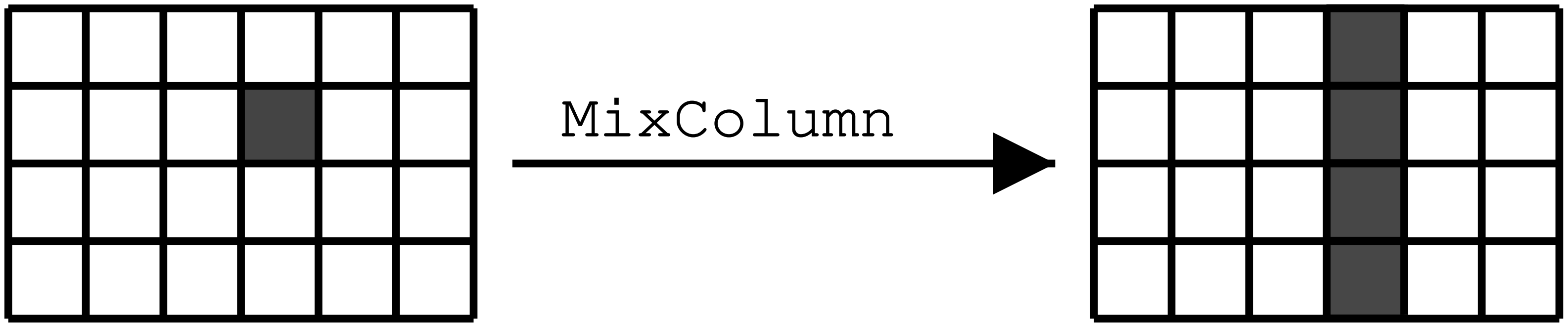
- SQUARE: Blockcipher von Rijmen und Daemen
- Attacke auf SQUARE:
  Funktioniert auch für 4–Runden–Rijndael
- Λ–Mengen:
  - Menge von 256 Blöcken
  - Byte an Position (i,j):
    - Entweder paarweise verschieden: „aktiv"
    - Oder in allen Blöcken gleich: „passiv"

$$x_{i,j} \neq y_{i,j} \ \forall x, y \in \Lambda (x \neq y) \text{ (aktiv)}$$

$$x_{i,j} = y_{i,j} \ \forall x, y \in \Lambda \text{ (passiv)}$$

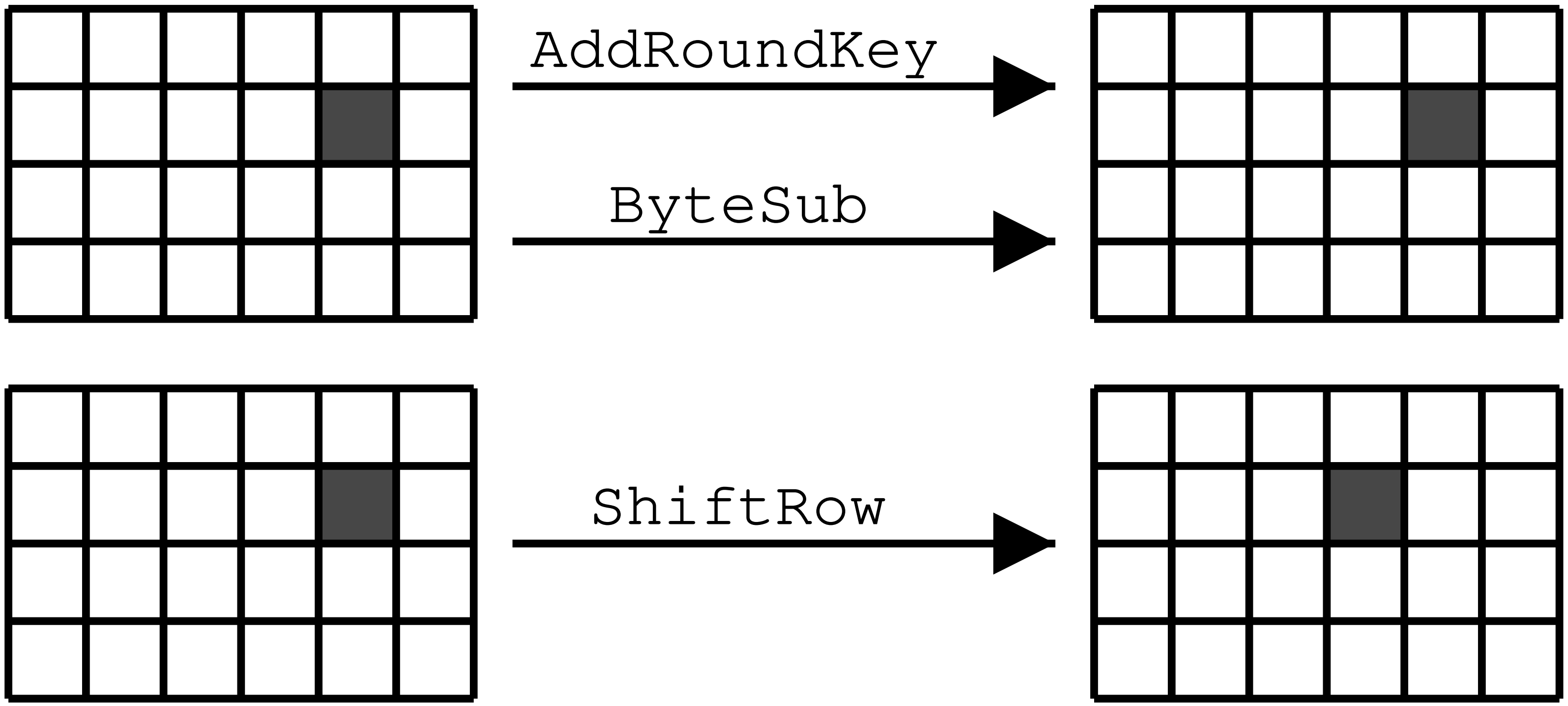
Rijndael–Operationen auf Λ–Mengen

- **AddRoundKey**: aktive Bytes bleiben aktiv, passive bleiben passiv
- **ByteSub**: ebenso
- **ShiftRow**: aktive Bytes werden verschoben
- **MixColumn**: liefert nicht unbedingt eine Λ–Menge!
- Aber: Λ–Menge mit genau einem aktiven Byte:
  $\Rightarrow$ Λ–Menge mit einer aktiven Spalte

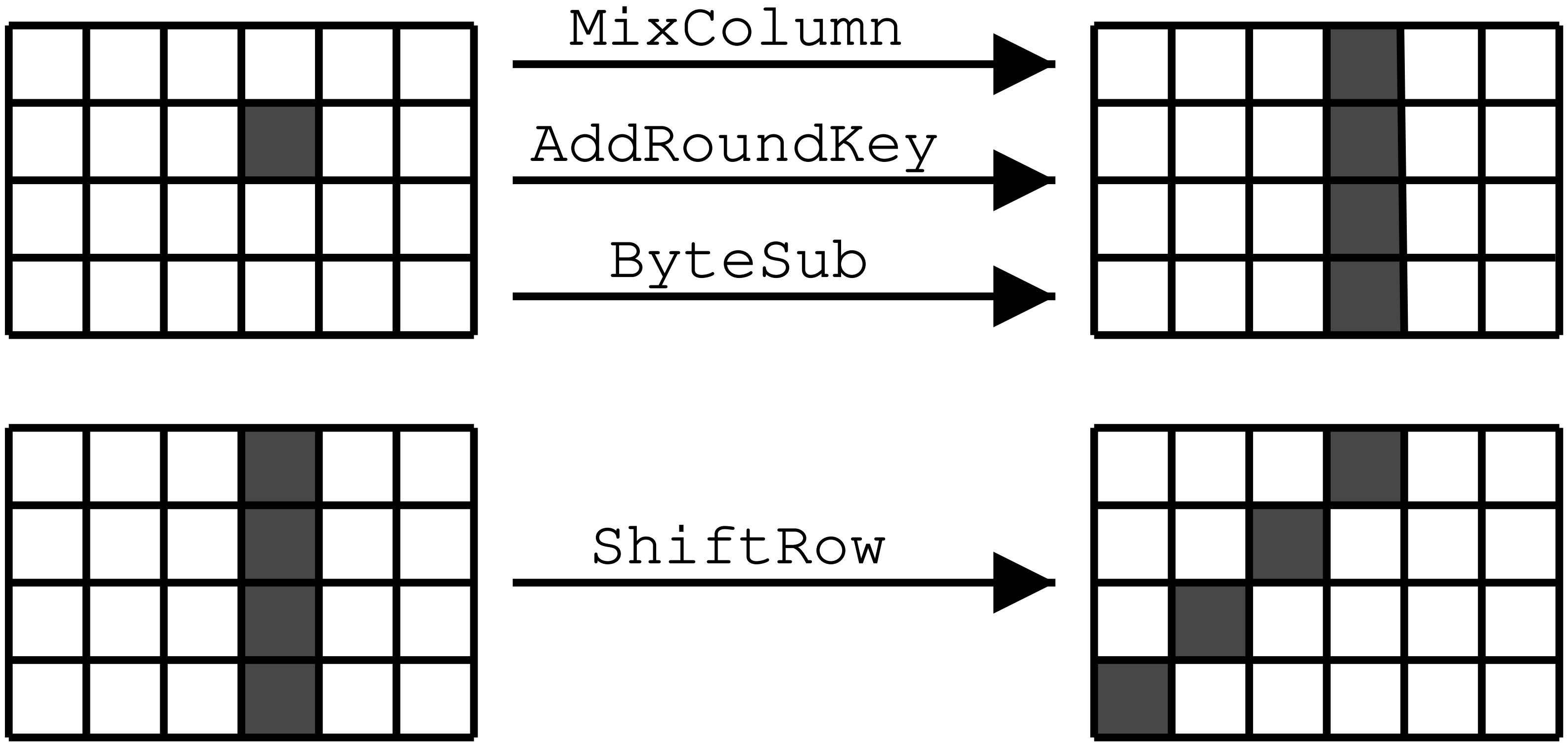


SQUARE–Attacke [1]

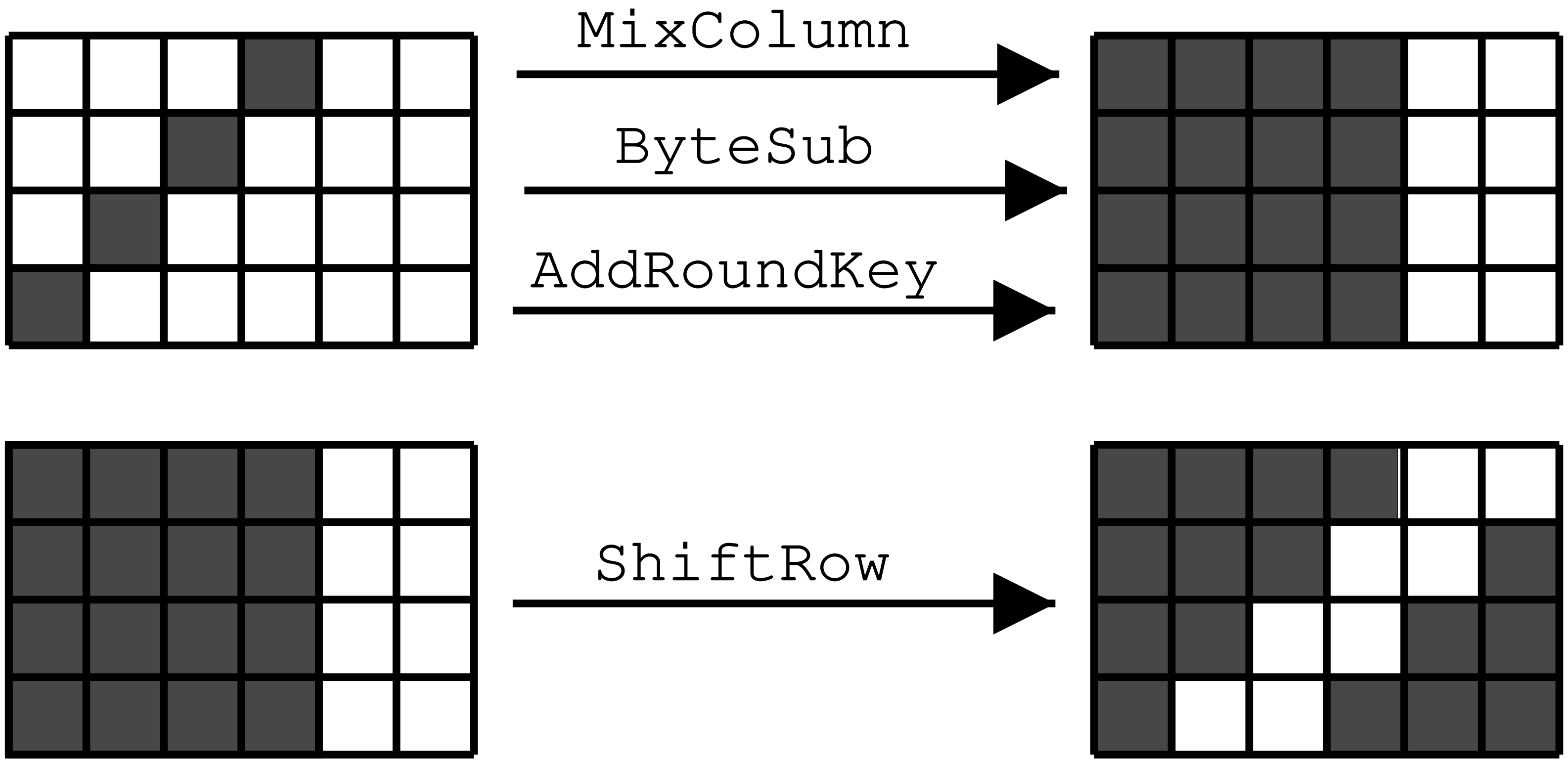
- Λ–Menge mit genau einem aktiven Byte
  (256 chosen plaintexts)



SQUARE–Attacke [2]



SQUARE–Attacke [3]



SQUARE–Attacke [4]

- Vor **MixColumn** in der 3.Runde:
  Λ–Menge
- Passive Bytes: in jedem Block gleich
  $\Rightarrow$ Summe über alle Blöcke 0 an jeder passiven Position
- Aktive Bytes: paarweise verschieden
  $\Rightarrow$ Summe auch 0
- Also:

$$\bigoplus_{a \in \Lambda} a_{i,j} = 0$$

SQUARE–Attacke [5]

- Nach **MixColumn** gilt: $b := MixColumn(a), a \in \Lambda$

$$\bigoplus_{b \in \Lambda'} b_{i,j} = \bigoplus_{a \in \Lambda} \left('02' \cdot a_{i,j} \oplus' 03' \cdot a_{i+1,j} \oplus a_{i+2,j} \oplus a_{i+3,j}\right) = 0$$

- Beginn der 4.Runde: Summe ist 0
- Diese Eigenschaft wird durch **ByteSub** zerstört
- Letzte Runde: ohne Operation MixColumn
  $\Rightarrow$ Jedes Output–Byte hängt von 1 Input–Byte ab

$$a_{i,j} = ByteSub(b_{i',j'}) \oplus k_{i,j}$$

$$b_{i',j'} = InvByteSub(a_{i,j} \oplus k_{i,j})$$

SQUARE–Attacke: Vorgehen

- 256 Klartexte (eine Λ–Menge) wählen
- Verschlüsseln lassen
- 1 Keybyte $k_{i,j}$ „raten”
- $b_{i',j'}$ berechnen
- Testen, ob die Summe über alle 256 $b_{i',j'}$ 0 ergibt
- Wenn nein: Keybyte falsch
- Wenn ja: Keybyte mit großer Wahrscheinlichkeit richtig
- Keybytes können einzeln ermittelt werden
- Man kann gesamten Rundenschlüsssel ermitteln

SQUARE–Attacke: Erweiterungen

- Kann auf mehr Runden erweitert werden (erfordert dann mehr *chosen plaintexts*)
- Bis zu 8 Runden für 192 und 256 Bit Schlüssel
  - Besser als *brute force*
  - Impraktikabel
    - Zu viele *chosen plaintexts* nötig
    - Zu langsam (Komplexität!)

Algebraische Repräsentation: ByteSub [1]

- S–Box: 256 Werte $\Rightarrow$ Polynominterpolation:

$$S(x) =' 63' +' 8F' \cdot x^{127} +' B5' \cdot x^{191} +' 01' \cdot x^{223} +' F4' \cdot x^{239} +$$
$$+' 25' \cdot x^{247} +' F9' \cdot x^{251} +' 09' \cdot x^{253} +' 05' \cdot x^{254}$$

- Kurze Schreibweise:

$$S(a_{i,j}) = w_8 + \sum_{d=0}^{7} w_d \cdot a_{i,j}^{255-2^d}$$

(Wobei die w_i die Konstanten aus der obigen Formel sind.)

Algebraische Repräsentation: ByteSub [2]

$$S(a_{i,j}) = w_8 + \sum_{d=0}^{7} w_d \cdot a_{i,j}^{255-2^d}$$

- Die Konstante w_8 bringt man in einem modifizierten Rundenschlüssel unter.
- Da $x^{255} \equiv 1 \ \forall x \in \mathbb{F}_{2^8}^*$ erhält man:

$$S(a_{i,j}) = \sum_{d=0}^{7} w_d \cdot a_{i,j}^{-2^d}$$

(Für $x = 0$ kommt man mit der Konvention $a/0 := 0$ aus.)

Alg. Rep.: ShiftRow/MixColumn

- ShiftRow

$$t_{i,j} = S(a_{i,i+j}) = \sum_{d=0}^{7} w_d \cdot a_{i,i+j}^{-2^d}$$

- MixColumn (für passende Konstanten $w_{i,e,d}$)

$$m_{i,j} = \sum_{e=0}^{3} v_{i,e} \cdot t_{e,j} = \sum_{e=0}^{3} (v_{i,e} \cdot \sum_{d=0}^{7} w_d \cdot a_{e,e+j}^{-2^d}) = \sum_{e=0}^{3} \sum_{d=0}^{7} w_{i,e,d} \cdot a_{e,e+j}^{-2^d}$$

$$(v_{i,j}) = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

Alg. Rep.: 1,2 Runden

- Eine Runde insgesamt (nach AddRoundKey):

$$r_{i,j} = m_{i,j} + k_{i,j} = k_{i,j} + \sum_{e=0}^{3} \sum_{d=0}^{7} w_{i,e,d} \cdot a_{e,e+j}^{-2^d}$$

- Zwei Runden:

$$a = k + \sum \frac{w}{(k + \sum \frac{w}{a^{2^{d_1}}})^{2^{d_2}}}$$

Da $(a+b)^c = a^c + b^c$ in Körpern mit Charakteristik c („freshman's dream"):

$$a = k + \sum \frac{w}{k^{2^{d_2}} + \sum \frac{w^{2^{d_2}}}{a^{2(d_1+d_2)}}}$$

Algebraische Repräsentation

- N Runden:

$$c = k + \sum \frac{w}{k^{2^d} + \sum \frac{w^{2^d}}{k^{2^d} + \sum \frac{w^{2^d}}{k^{2^d} + \ldots + \sum \frac{w^{2^d}}{k^{2^d} + p^{2^d}}}}}$$

- Geschlossene, relativ einfache Formel für den kompletten Cipher!
- Ohne Summationszeichen: groß, nicht riesig (10 Runden: $\approx 2^{50}$ Brüche der Form $\frac{w}{K+p}$)
- Nach den ps auflösen???

Alg. Rep.: Konsequenzen

- Algebraische Attacken?
- Keine Angriffe bekannt
- Wenig Kenntnisse über derartige Gleichungen
- **Härtebedingung:**
  Gleichungen dieser Art müssen schwer lösbar sein!
  sonst wäre Rijndael zu schwach
- Dies war zum Zeitpunkt, als Rijndael für AES ausgewählt wurde, nicht bekannt.

Algebraische Eigenschaften

- Umgruppierung der Operationen:
  - Non–linear Layer: nur Inversion $x \mapsto x^{-1}$
  - Matrix: linear
  - Konstante: in den Rundenschlüssel
- Linear Mixing Layer: durch eine Matrix beschreibbar (für 128 Bit Blöcke: binäre 128×128–Matrix)
- Es zeigt sich: Matrix hat lediglich Ordnung 16!
- Vergleich: DES: 1020
- Vielleicht ein Schwachpunkt?
- Es ist kein Angriff bekannt, der dies ausnutzt...

Résumée

- AES/Rijndael hat große Bedeutung in der Praxis
  (vor allem in der Zukunft)
- Gilt heutzutage als sicher
- Es gibt jedoch gewisse Bedenken
- Einige mögliche Problempunkte wurden hier gezeigt
- Noch relativ neu

Niemand weiß, wie sicher Rijndael wirklich ist!