V. Substitution-Permutation und AES

Entwurfsprinzipien moderner Blockchiffren

- Einfache Operationen, aber nicht nur linear (Effizienz)
- Mehrere Runden um Sicherheit zu garantieren (Konfusion und Diffusion)
- 2 wesentliche Strukturen
  - Feistel-Chiffren (DES)
  - Substitution-Permutations Netzwerke (AES).

V. Substitutions-Permutations-Chiffren

SP-Chiffren definiert durch

1. Klartextraum $P = \{0,1\}^n$, Chiffretextraum $C = \{0,1\}^n$, Schlüsselraum $K = \{0,1\}^n$, wobei $n = t \cdot b$.

2. Rundenzahl $r > 1$.

3. Methode zur Erzeugung von Rundenschlüssel
   KeySchedule: $\{0,1\}^n \to \{0,1\}^{(r+1)\cdot n}$
   $K \mapsto K_0,\ldots,K_r$

4. Bijektive Substitutionen $S_i : \{0,1\}^b \to \{0,1\}^b, i = 1,\ldots,t$.

5. Permutation $P : \{1,\ldots,n\} \to \{1,\cdots,n\}$.

V. Substitutions-Permutations-Chiffren

SP-Chiffren definiert durch

1. Klartextraum $P = \{0,1\}^n$, Chiffretextraum $C = \{0,1\}^n$, Schlüsselraum $K = \{0,1\}^n$, wobei $n = t \cdot b$.

2. Rundenzahl $r > 1$.

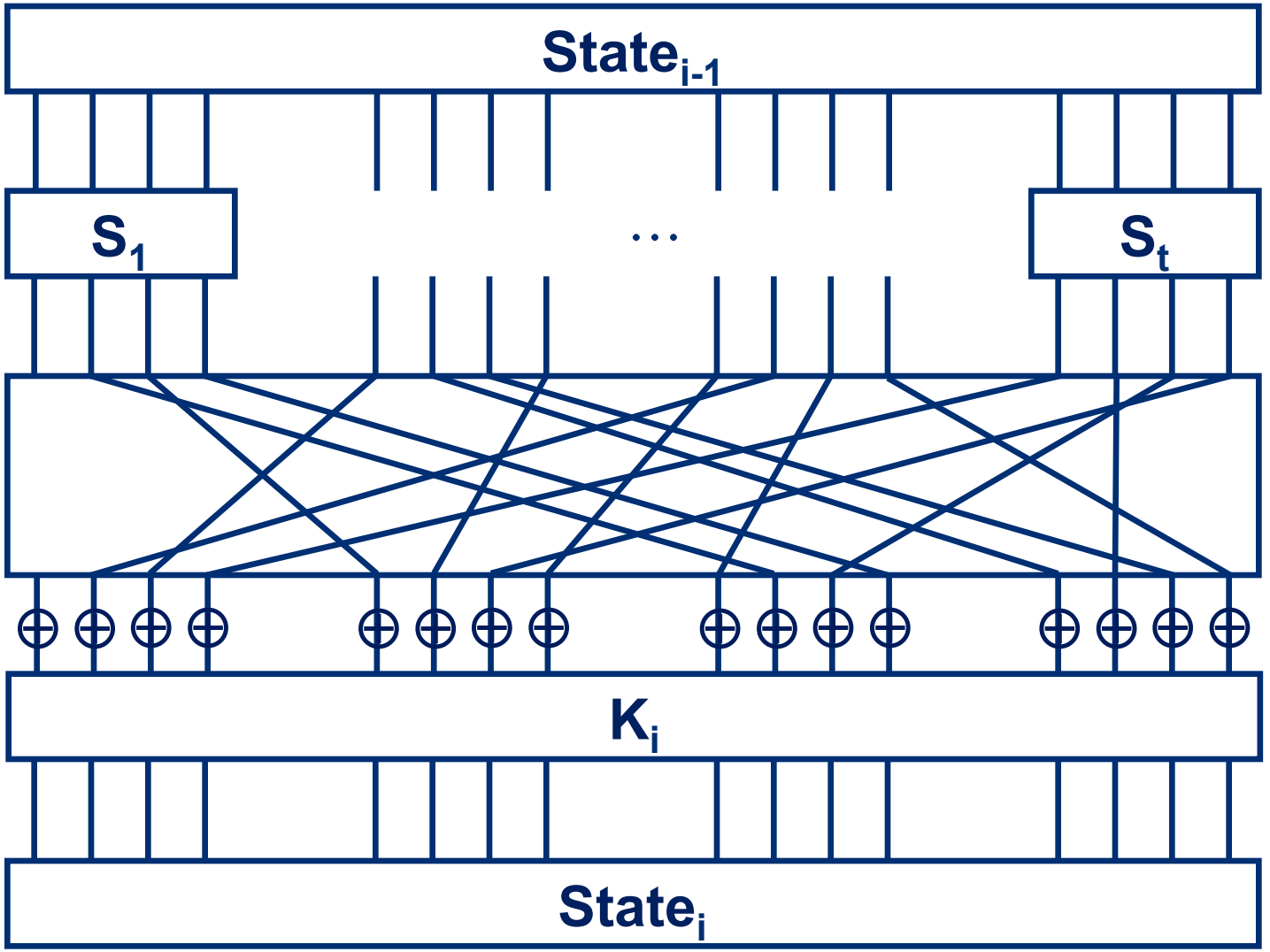
3. Methode zur Erzeugung von Rundenschlüssel
   KeySchedule: $\{0,1\}^n \to \{0,1\}^{(r+1)\cdot n}$
   $K \mapsto K_0,\ldots,K_r$

4. Bijektive Substitutionen $S_i : \{0,1\}^b \to \{0,1\}^b, i = 1,\ldots,t$.

5. Bijektive lineare Funktion $P : \{0,1\}^n \to \{0,1\}^n$.

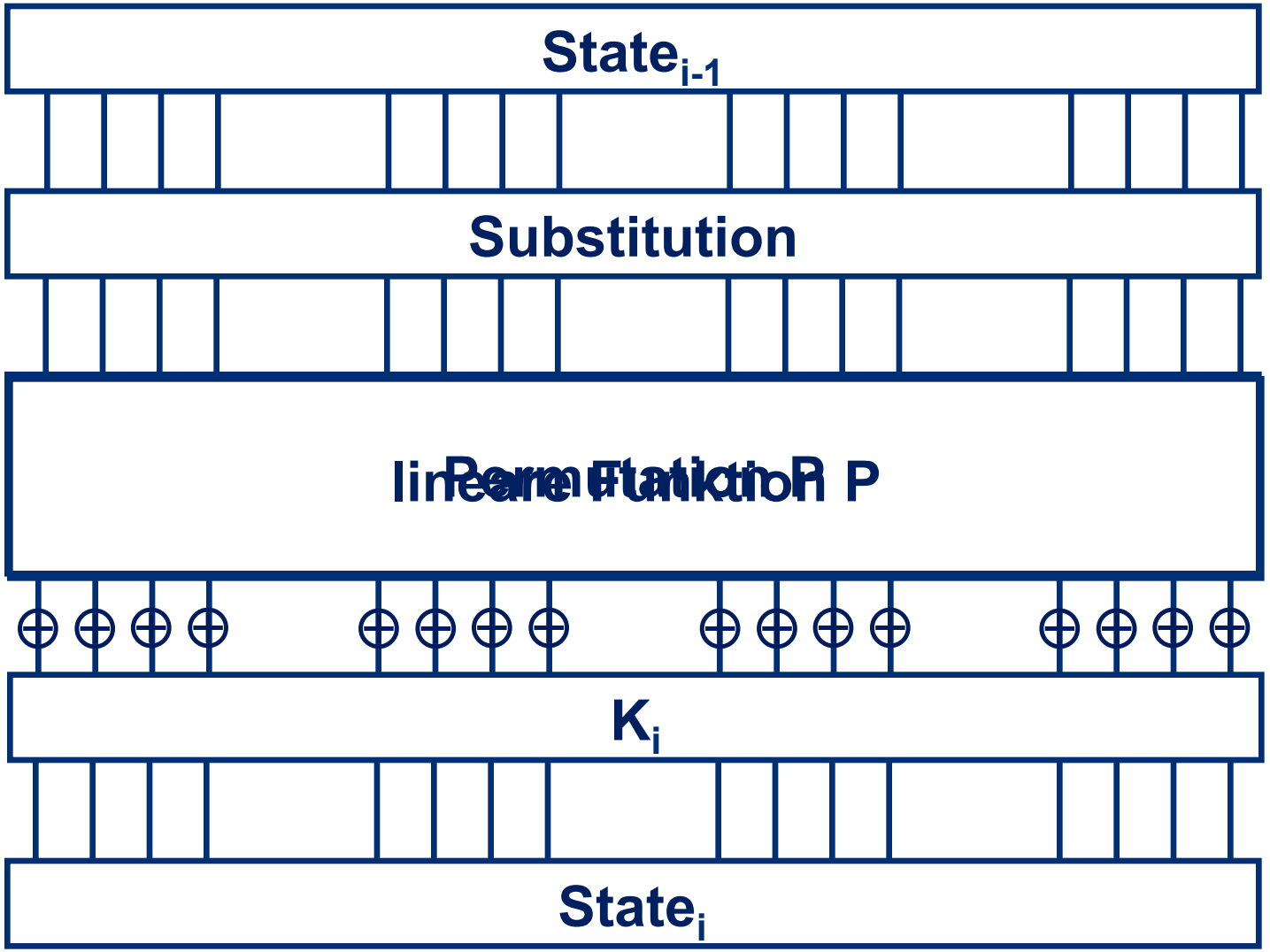
SPN – eine Runde

$$State_i := P\left(S\left(State_{i-1}\right)\right) \oplus K_i, \quad State_j \in \{0,1\}^n$$



SPN – eine Runde

$$State_i := P\left(S\left(State_{i-1}\right)\right) \oplus K_i,\ State_j \in \{0,1\}^n$$



SPN – Initialisierung

- Klartext $p \in \{0,1\}^n$
- $State_0 = p \oplus K_0$
- Whitening genannt
- Damit hängen alle Zustände vom (geheimen) Schlüssel ab.

V.2 Advanced Encryption Standard AES

- 1997 Ausschreibung des NIST für Nachfolger von DES.
- Bei Abgabeschluss 1998 15 Vorschläge.
- 1999 werden 5 Kandidaten ausgesucht (MARS, RC6, Rijndael, Serpent, Twofish).
- 2. Oktober 2000 wird Rijndael als Sieger bekannt gegeben.
- Am 26. November 2001 wird DES durch AES-Rijndael als Standard der NIST ersetzt.
- AES-Rijndael entwickelt von Joan Daemen und Vincent Rijmen.

AES - Parameter

AES ist SP-Chiffre mit

1. Klartextraum $P = \{0,1\}^{128}$, Chiffretextraum $C = \{0,1\}^{128}$, Schlüsselraum $K = \{0,1\}^{128}$, wobei b=8, t=16 .

2. Rundenzahl r = 10.

3. Methode zur Erzeugung von Rundenschlüssel
   KeySchedule: $\{0,1\}^{128} \rightarrow \{0,1\}^{11 \cdot 128}$
   $K \mapsto K_0,\ldots,K_{10}$

4. Bijektive Substitution $S : \{0,1\}^8 \rightarrow \{0,1\}^8$

5. Lineare Bijektion $P : \{0,1\}^{128} \rightarrow \{0,1\}^{128}$.

AES – Varianten

Varianten von AES besitzen

- Nachrichtenlänge 192 und 256.
- Schlüssellänge 192 und 256
- Rundenzahl 12 und 14.
- Betrachten nur den Standardfall,
- Operationen in Varianten identisch.

Zustände in AES

$$\mathbf{State}_j = \begin{array}{|c|c|c|c|} \hline B_{00} & B_{01} & B_{02} & B_{03} \\ \hline B_{10} & B_{11} & B_{12} & B_{13} \\ \hline B_{20} & B_{21} & B_{22} & B_{23} \\ \hline B_{30} & B_{31} & B_{32} & B_{33} \\ \hline \end{array}$$

$$B_{ij} \in \{0,1\}^8, i,j \in \{0,1,2,3\}$$

AES Rundenstruktur

State ← p

State ←AddRoundKey(State, K_0)

For i=1 to 9 do

    State SubBytes(State)

    State ← ShiftRows(State)

    State ← MixColumns(State)

    State ← AddRoundKey(State, K_i)

} Funktion P

State ← SubBytes(State)

State ← ShiftRows(State)

State ← AddRoundKey(State, K_{10})

return State

V.3 Bytes – Interpretationen

$$\{0,1\}^8 = \{0,1\}^4 \times \{0,1\}^4$$

$$\{0,1\}^4 \doteq \{0,1,2,\ldots,15\}$$

Schreiben Elemente aus $\{0,1,2,\ldots,15\}$ in Hexadezimaldarstellung.

$$\{0,1,2,\ldots,14,15\} \doteq \{0,1,2,\ldots,9,A,B,C,D,E,F\}$$

Bytes – Hexadezimalzahlen

$$\{0,1\}^8 = \{0,1\}^4 \times \{0,1\}^4$$

$$\{0,1\}^4 \doteq \{0,1,2,\ldots,15\}$$

Schreiben Elemente aus $\{0,1,2,\ldots,15\}$ in Hexadezimaldarstellung.

$$\{0,1,2,\ldots,14,15\} \doteq \{0,1,2,\ldots,9,A,B,C,D,E,F\}$$

Beispiele

- 00000000 ≐ 00
- 00000001 ≐ 01
- 00000010 ≐ 02
- 10110101 ≐ B5

Bytes und der endliche Körper $\mathbb{F}_{256}$

Satz 5.1 Es gibt einen endlichen Körper $\mathbb{F}_{256}$ mit 256 Elementen. Dieser Körper ist bis auf Isomorphie eindeutig.

$256 = 2^8$, daher können wir Bytes und Paare von Hexadezimalzahlen mit Elementen von $\mathbb{F}_{256}$ identifizieren.

AES Rundenstruktur

State ← p

State ←AddRoundKey(State, K_0)

For i=1 to 9 do

    State SubBytes(State)

    State ← ShiftRows(State)

    State ← MixColumns(State)

    State ← AddRoundKey(State, K_i)

} Funktion P

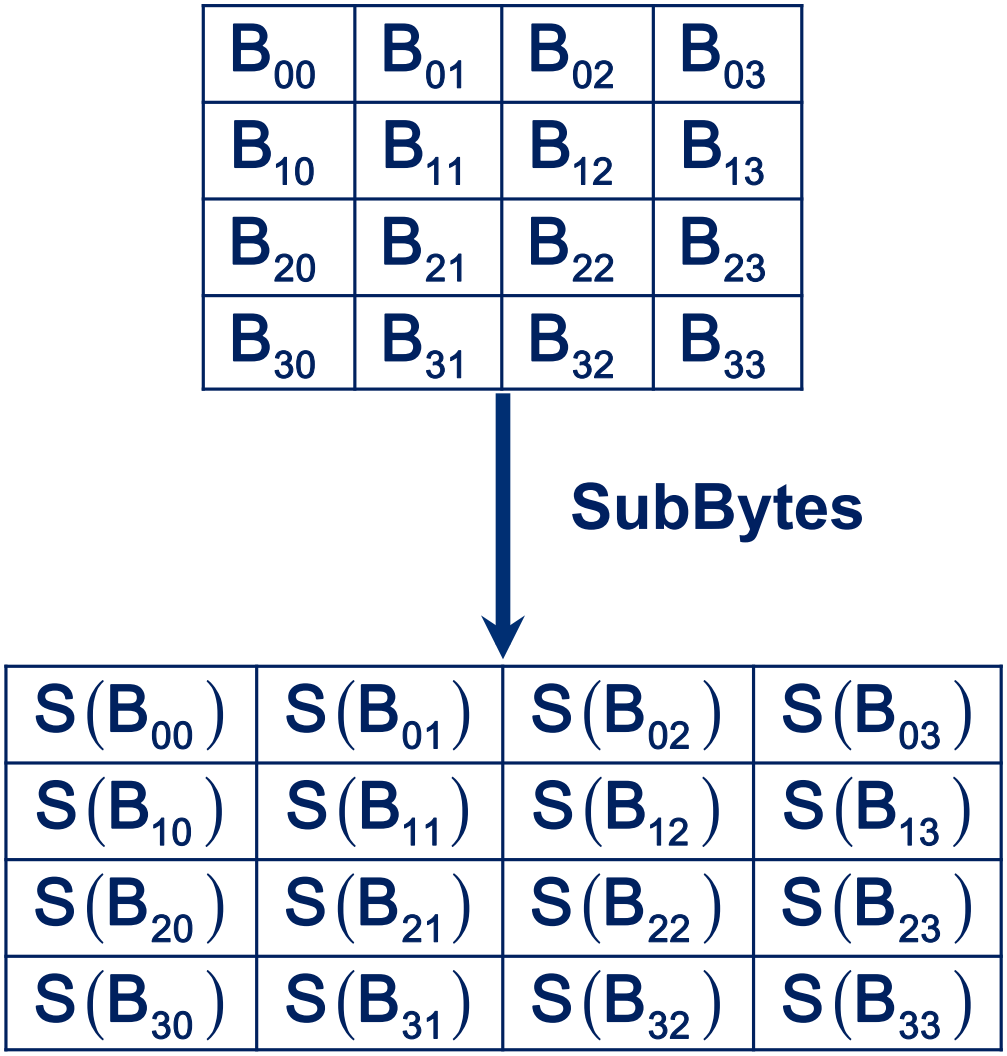
State ← SubBytes(State)

State ← ShiftRows(State)

State ← AddRoundKey(State, K_{10})

return State

V.4 SubBytes



$S : \{0,1\}^8 \rightarrow \{0,1\}^8$ **nicht-lineare Abbildung**

SubBytes

	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

AES Rundenstruktur

State ← p

State ←AddRoundKey(State, K_0)

For i=1 to 9 do

    State ← SubBytes(State)

    State ← ShiftRows(State)

    State ← MixColumns(State)

    State ← AddRoundKey(State, K_i)

} Funktion P

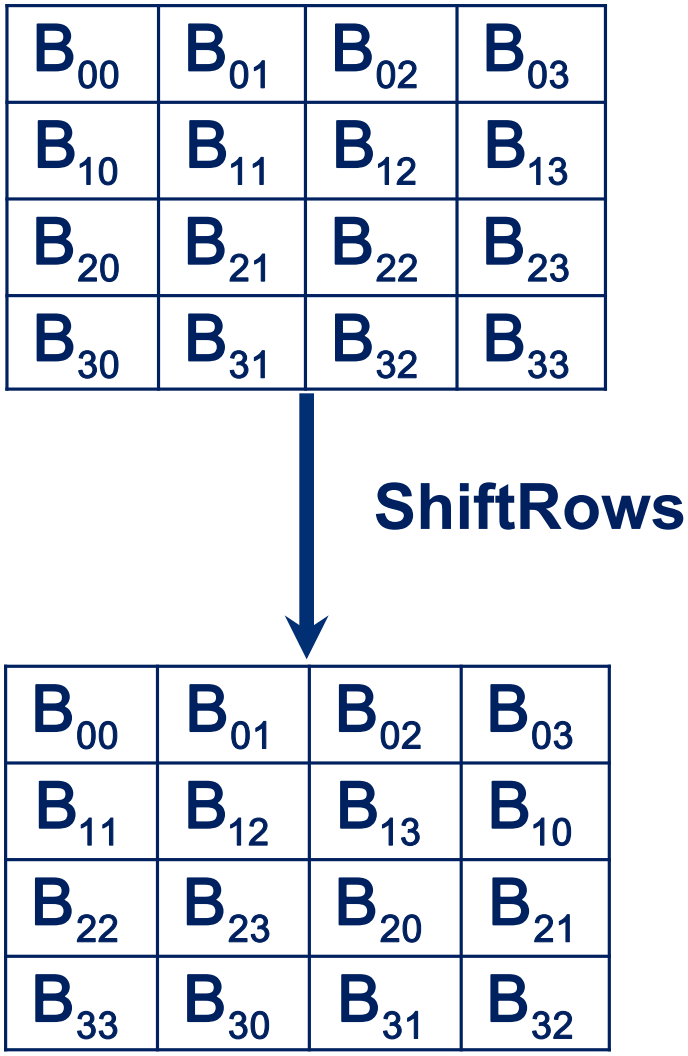
State ← SubBytes(State)

State ← ShiftRows(State)

State ← AddRoundKey(State, K_{10})

return State

V.5 ShiftRows und MixColumns



MixColumns

B_{00}	B_{01}	B_{02}	B_{03}
B_{10}	B_{11}	B_{12}	B_{13}
B_{20}	B_{21}	B_{22}	B_{23}
B_{30}	B_{31}	B_{32}	B_{33}

MixColumns

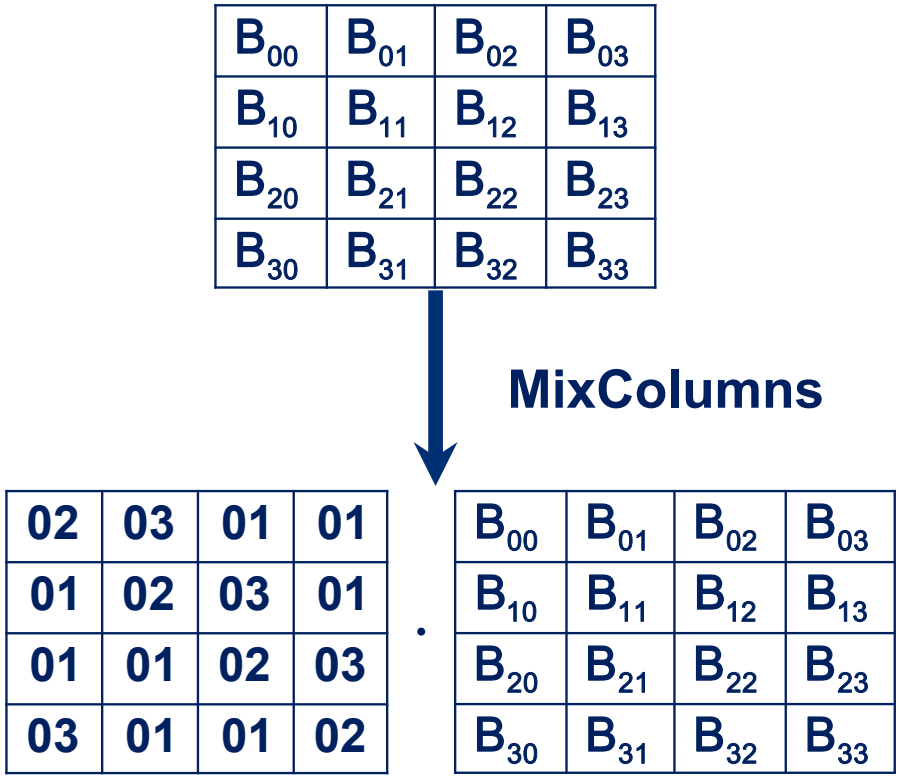
02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

$\cdot$

B_{00}	B_{01}	B_{02}	B_{03}
B_{10}	B_{11}	B_{12}	B_{13}
B_{20}	B_{21}	B_{22}	B_{23}
B_{30}	B_{31}	B_{32}	B_{33}

Dabei finden alle Operationen in $\mathbb{F}_{256}$ statt!

MixColumns



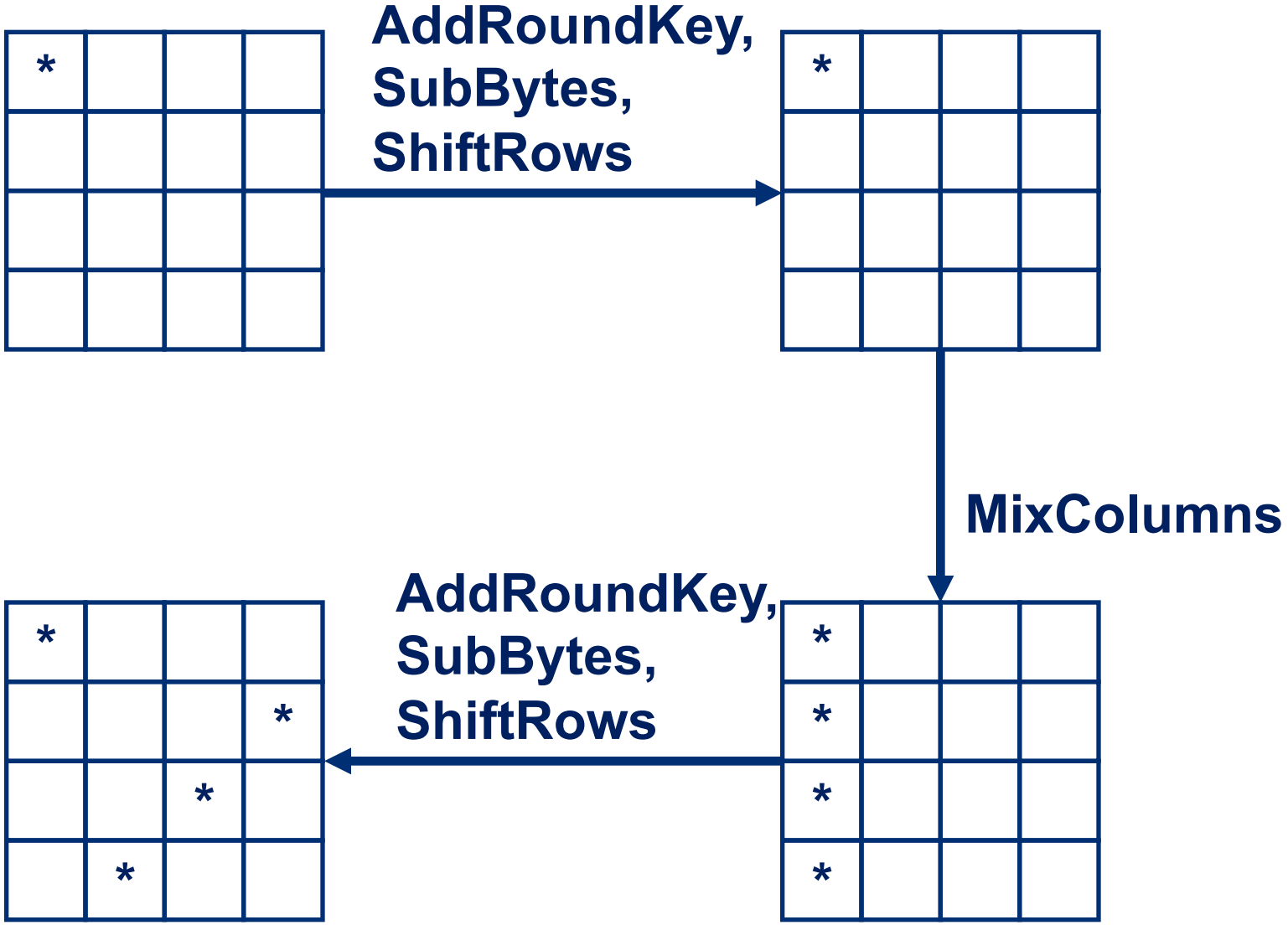
Lemma 5.6 **Sei v ein Vektor aus 4 Bytes mit t von Null verschiedenen Bytes, t > 0. MixColumns bildet diesen Vektor auf einen Vektor ab, der mindestens 5-t von Null verschiedene Bytes besitzt..**

(MDS-Eigenschaft = maximum distance separable)

V.6 Diffusion in AES

Diffusion **Die Diffusion einer Blockchiffre ist groß, wenn jedes Bit des Klartextes und jedes Bit des Schlüssels möglichst viele Bits des Chiffretexts beeinflusst.**

Diffusion in AES



* := Positionen mit unterschiedlichen Bytes in zwei Zuständen

Diffusion in AES

* := Positionen mit unterschiedlichen Bytes in zwei Zuständen

AES – KeySchedule

… ist nicht sehr erhellend!

Entschlüsselung in AES

State ← p

State AddRoundKey(State, K_{10})

For i=9 downto 1 do

    State ← InvShiftRows(State)

    State ← InvSubBytes(State)

    State ← AddRoundKey(State, K_i)

    State ← InvMixColumns(State)

State ← InvShiftRows(State)

State ← InvSubBytes(State)

State ← AddRoundKey(State, K_0)

return State

Inv*:= inverse Operation zu *

Entschlüsselung in AES

AES Verschlüsselung

State ← p
State ←AddRoundKey(State, K_0)
For i=1 to 9 do
    State SubBytes(State)
    State ← ShiftRows(State)
    State ← MixColumns(State)
    State ← AddRoundKey(State, K_i)
State ← SubBytes(State)
State ← ShiftRows(State)
State ← AddRoundKey(State, K_{10})
return State

AES Entschlüsselung

State ← p
State AddRoundKey(State, K_{10})
For i=9 downto 1 do
    State ← InvShiftRows(State)
    State ← InvSubBytes(State)
    State ← AddRoundKey(State, K_i)
    State ← InvMixColumns(State)
State ← InvShiftRows(State)
State ← InvSubBytes(State)
State ← AddRoundKey(State, K_0)
return State

Zusammenfassung symmetrische Verschlüsselung

- Sicherheit nur beruhend auf geheimen Schlüssel (Kerckhoffs Prinzip)
- Diffusion und Konfusion
- Einfache Chiffren wie
  - Substitutions-Chiffren
  - Permutations-Chiffen
  - affine Chiffren

  nicht sicher
- Chiffre darf nicht linear sein

Zusammenfassung symmetrische Verschlüsselung

- Verschlüsselungsmodi und Blockchiffren
- perfekte Sicherheit
- Feistel-Chiffren und DES
- SP-Chiffren und AES