Advanced Encryption Standard

Folien zur Präsentation

Michael Dienert

Walther-Rathenau-Gewerbeschule Freiburg

9. Dezember 2021

Inhalt

AES

Blockchiffren

- AES gehört zu den *Blockchiffren*
- Blockchiffre bedeutet, dass immer Datenblöcke aus 128bit, 192bit oder 256bit in einem Durchgang verschlüsselt werden.
- Der symmetrische Schlüssel muss dabei exakt gleich lang wie ein einzelner Block sein.
- Die folgenden Beispiele beziehen sich auf 128bit Schlüssellänge. Damit können z.B. 16 Textzeichen verschlüsselt werden. Pro Textzeichen ergeben sich 8bit, je nach Zeichensatzcodierung. Z.B. UTF8. Die Darstellung erfolgt Hexadezimal
- “Das ist Klartext”
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Blockchiffren

- AES gehört zu den *Blockchiffren*
- Blockchiffre bedeutet, dass immer Datenblöcke aus 128bit, 192bit oder 256bit in einem Durchgang verschlüsselt werden.
- Der symmetrische Schlüssel muss dabei exakt gleich lang wie ein einzelner Block sein.
- Die folgenden Beispiele beziehen sich auf 128bit Schlüssellänge. Damit können z.B. 16 Textzeichen verschlüsselt werden. Pro Textzeichen ergeben sich 8bit, je nach Zeichensatzcodierung. Z.B. UTF8. Die Darstellung erfolgt Hexadezimal
- “Das ist Klartext”
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Blockchiffren

- AES gehört zu den *Blockchiffren*
- Blockchiffre bedeutet, dass immer Datenblöcke aus 128bit, 192bit oder 256bit in einem Durchgang verschlüsselt werden.
- Der symmetrische Schlüssel muss dabei exakt gleich lang wie ein einzelner Block sein.
- Die folgenden Beispiele beziehen sich auf 128bit Schlüssellänge. Damit können z.B. 16 Textzeichen verschlüsselt werden. Pro Textzeichen ergeben sich 8bit, je nach Zeichensatzcodierung. Z.B. UTF8. Die Darstellung erfolgt Hexadezimal
- “Das ist Klartext”
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Blockchiffren

- AES gehört zu den *Blockchiffren*
- Blockchiffre bedeutet, dass immer Datenblöcke aus 128bit, 192bit oder 256bit in einem Durchgang verschlüsselt werden.
- Der symmetrische Schlüssel muss dabei exakt gleich lang wie ein einzelner Block sein.
- Die folgenden Beispiele beziehen sich auf 128bit Schlüssellänge. Damit können z.B. 16 Textzeichen verschlüsselt werden. Pro Textzeichen ergeben sich 8bit, je nach Zeichensatzcodierung. Z.B. UTF8. Die Darstellung erfolgt Hexadezimal
- “Das ist Klartext”
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Blockchiffren

- AES gehört zu den *Blockchiffren*
- Blockchiffre bedeutet, dass immer Datenblöcke aus 128bit, 192bit oder 256bit in einem Durchgang verschlüsselt werden.
- Der symmetrische Schlüssel muss dabei exakt gleich lang wie ein einzelner Block sein.
- Die folgenden Beispiele beziehen sich auf 128bit Schlüssellänge. Damit können z.B. 16 Textzeichen verschlüsselt werden. Pro Textzeichen ergeben sich 8bit, je nach Zeichensatzcodierung. Z.B. UTF8. Die Darstellung erfolgt Hexadezimal
- “Das ist Klartext”
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

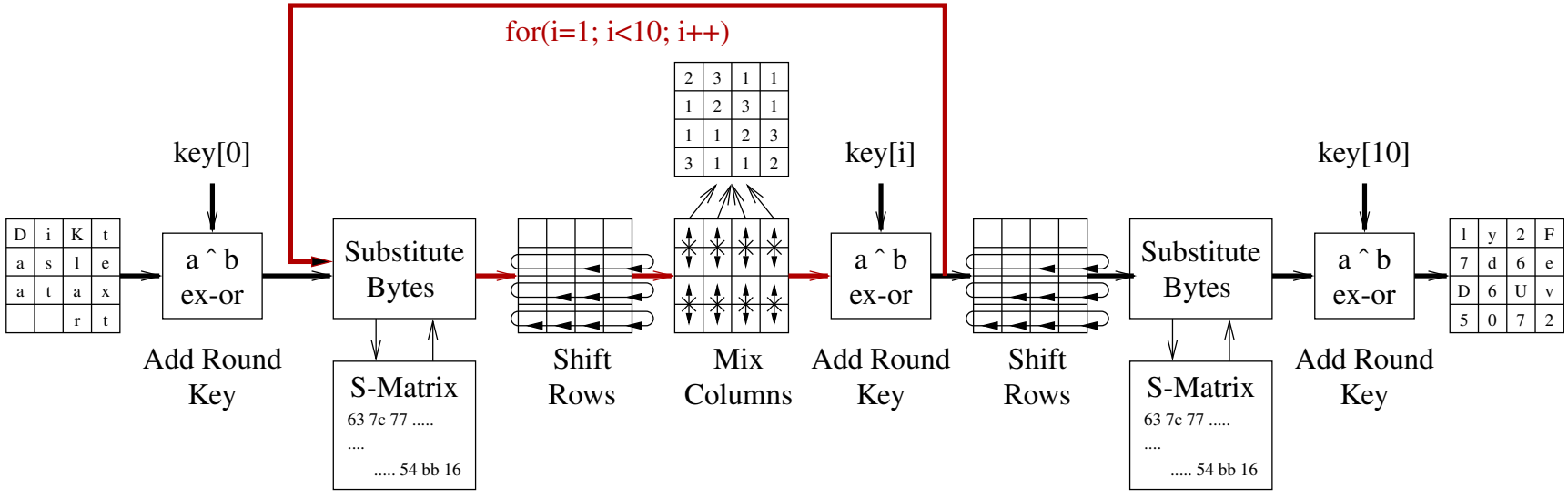
Blockchiffren

- AES gehört zu den *Blockchiffren*
- Blockchiffre bedeutet, dass immer Datenblöcke aus 128bit, 192bit oder 256bit in einem Durchgang verschlüsselt werden.
- Der symmetrische Schlüssel muss dabei exakt gleich lang wie ein einzelner Block sein.
- Die folgenden Beispiele beziehen sich auf 128bit Schlüssellänge. Damit können z.B. 16 Textzeichen verschlüsselt werden. Pro Textzeichen ergeben sich 8bit, je nach Zeichensatzcodierung. Z.B. UTF8. Die Darstellung erfolgt Hexadezimal
- “Das ist Klartext”
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Blockchiffren

- AES gehört zu den *Blockchiffren*
- Blockchiffre bedeutet, dass immer Datenblöcke aus 128bit, 192bit oder 256bit in einem Durchgang verschlüsselt werden.
- Der symmetrische Schlüssel muss dabei exakt gleich lang wie ein einzelner Block sein.
- Die folgenden Beispiele beziehen sich auf 128bit Schlüssellänge. Damit können z.B. 16 Textzeichen verschlüsselt werden. Pro Textzeichen ergeben sich 8bit, je nach Zeichensatzcodierung. Z.B. UTF8. Die Darstellung erfolgt Hexadezimal
- “Das ist Klartext”
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

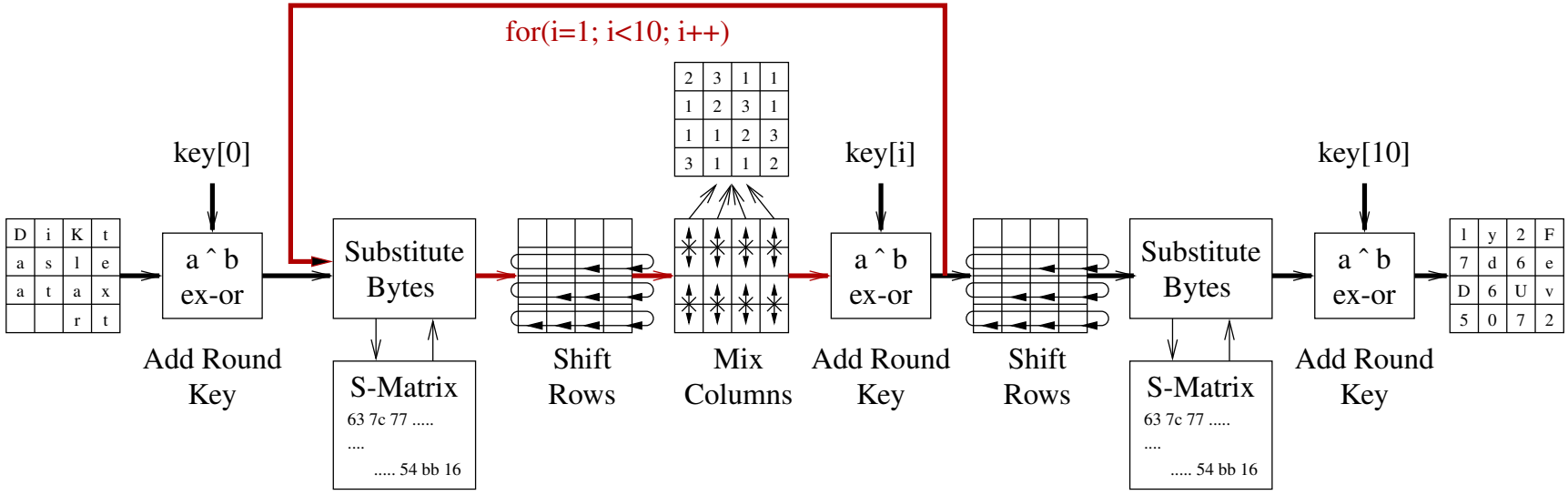
Übersicht AES-Algorithmus



Ablauf von AES

- Die Grafik zeigt AES am Beispiel von 128bit-Blockgrösse/Schlüssellänge
- AES-128 10 Runden
  AES-192 12 Runden
  AES-256 14 Runden

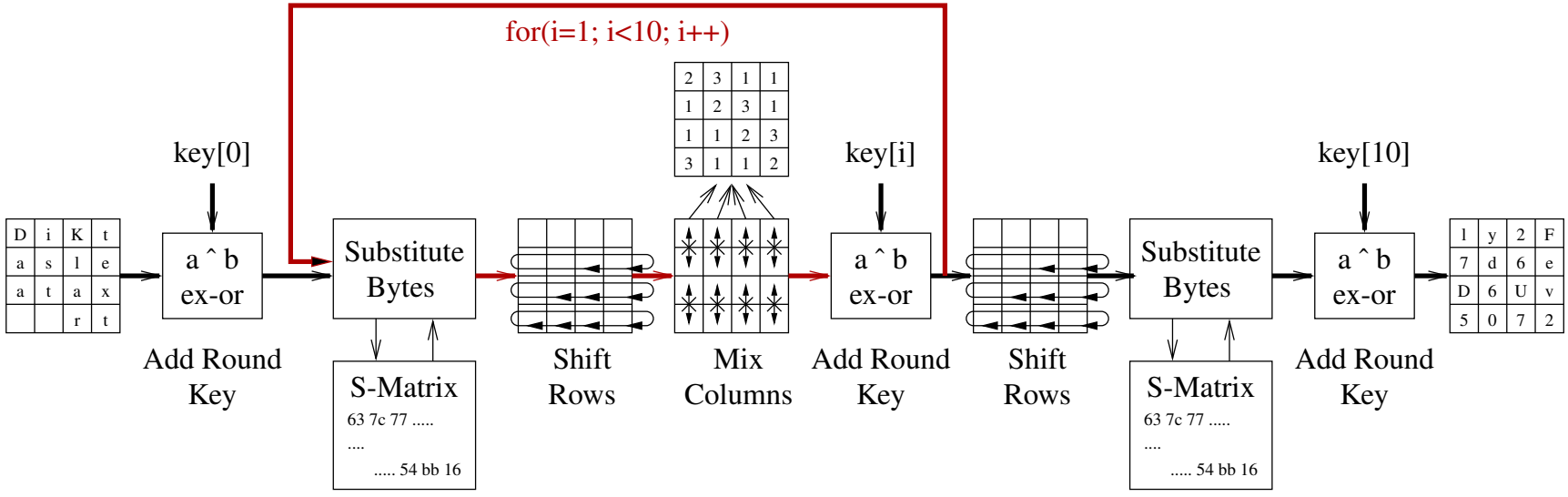
Übersicht AES-Algorithmus



Ablauf von AES

- Die Grafik zeigt AES am Beispiel von 128bit-Blockgrösse/Schlüssellänge
- AES-128 10 Runden
  AES-192 12 Runden
  AES-256 14 Runden

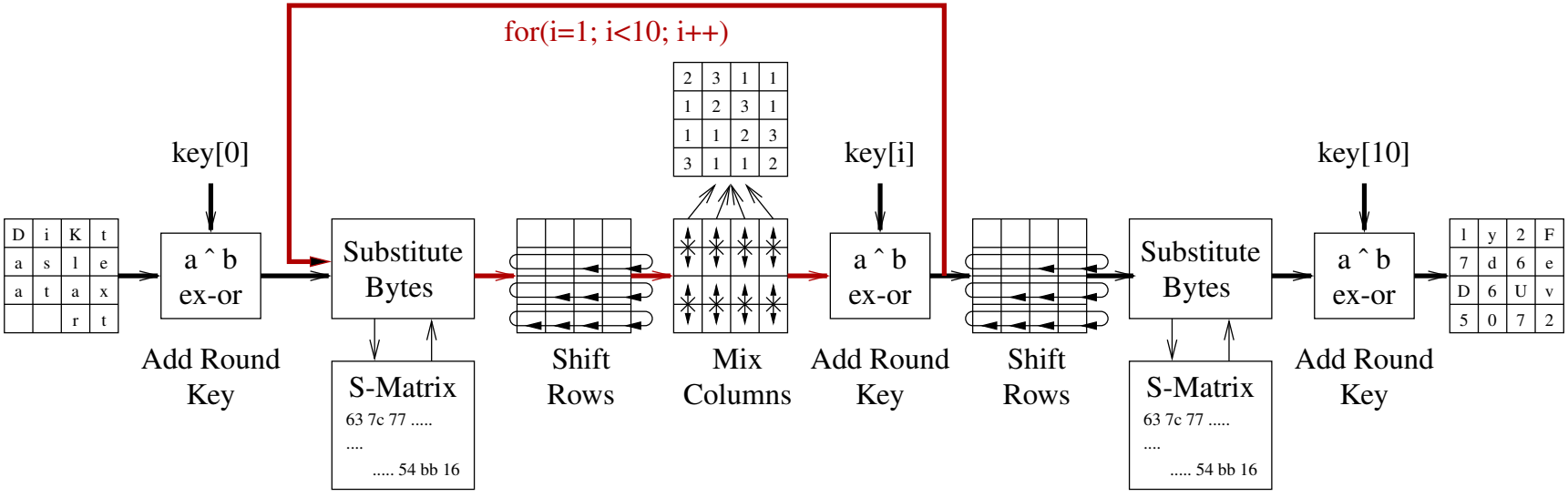
Übersicht AES-Algorithmus



Ablauf von AES

- Die Grafik zeigt AES am Beispiel von 128bit-Blockgrösse/Schlüssellänge
- AES-128 10 Runden
  AES-192 12 Runden
  AES-256 14 Runden

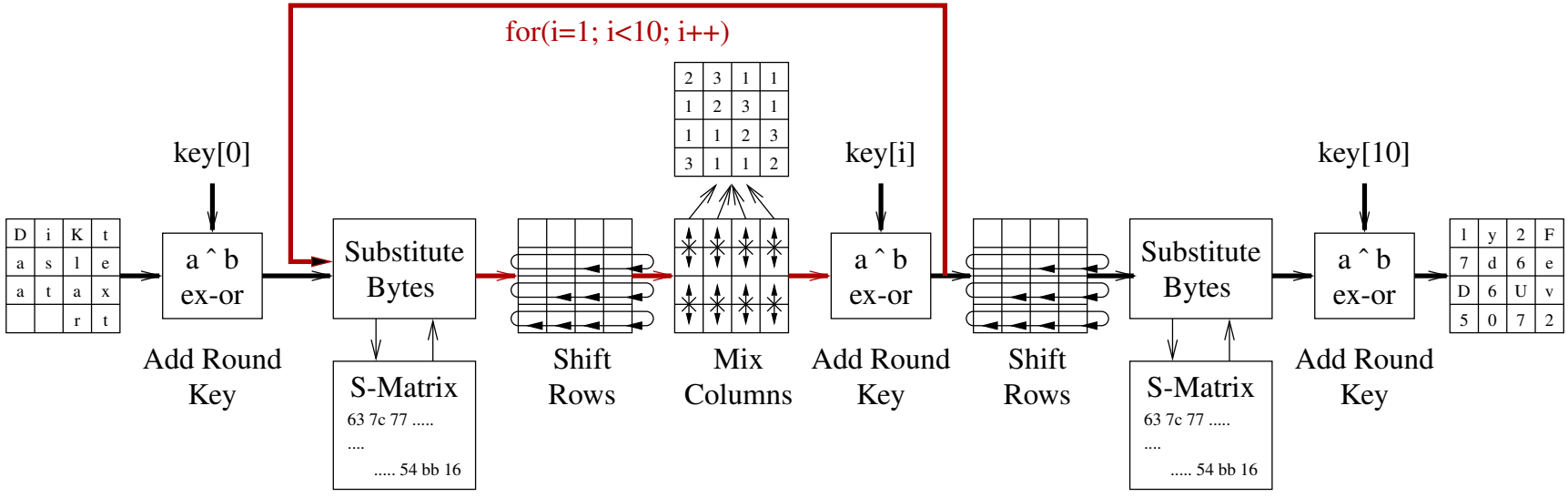
Übersicht AES-Algorithmus



Ablauf von AES

- Die Grafik zeigt AES am Beispiel von 128bit-Blockgrösse/Schlüssellänge
- AES-128 10 Runden
  AES-192 12 Runden
  AES-256 14 Runden

Übersicht AES-Algorithmus



Ablauf von AES

- Die Grafik zeigt AES am Beispiel von 128bit-Blockgrösse/Schlüssellänge
- AES-128 10 Runden
  AES-192 12 Runden
  AES-256 14 Runden

Key Expansion

- in jeder Runde wird ein anderer Schlüssel verwendet:
- Initialschlüssel k[0] und die Rundenschlüssel k[1] $\cdots$ k[10]
- der initiale Schlüssel k[0] ist eine Matrix aus 4x4 Bytes, also 4 Spalten
- aus den Spalten dieser Matrix werden durch Andwendung der Substitutionsmatrix, zyklischer Links-Verschiebung der Bytes einer Spalte und der EX-OR-Verknüpfung *rekursiv* zusätzliche Spalten errechnet.
- Im Beispiel der 4x4-Initial-Matrix (AES-128) werden rekursiv weitere 40 Spalten errechnet.
- 4 Spalten bilden immer einen Rundenschlüssel.

Substitute Bytes

	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

Substitute Bytes inverse

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Beispiel für Substitute Bytes

- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74
- Zeile 4, Spalte 4 $\Rightarrow$ 1b
- 1b ef 8f b7 f9 8f 92 b7 b3 50 ef 40 92 4d bc 92
- Inverse S-Box anwenden: 1b $\Rightarrow$ 44
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Beispiel für Substitute Bytes

- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74
- Zeile 4, Spalte 4 $\Rightarrow$ 1b
- 1b ef 8f b7 f9 8f 92 b7 b3 50 ef 40 92 4d bc 92
- Inverse S-Box anwenden: 1b $\Rightarrow$ 44
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Beispiel für Substitute Bytes

- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74
- Zeile 4, Spalte 4 $\Rightarrow$ 1b
- 1b ef 8f b7 f9 8f 92 b7 b3 50 ef 40 92 4d bc 92
- Inverse S-Box anwenden: 1b $\Rightarrow$ 44
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Beispiel für Substitute Bytes

- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74
- Zeile 4, Spalte 4 $\Rightarrow$ 1b
- 1b ef 8f b7 f9 8f 92 b7 b3 50 ef 40 92 4d bc 92
- Inverse S-Box anwenden: 1b $\Rightarrow$ 44
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Beispiel für Substitute Bytes

- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74
- Zeile 4, Spalte 4 $\Rightarrow$ 1b
- 1b ef 8f b7 f9 8f 92 b7 b3 50 ef 40 92 4d bc 92
- Inverse S-Box anwenden: 1b $\Rightarrow$ 44
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Beispiel für Substitute Bytes

- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74
- Zeile 4, Spalte 4 $\Rightarrow$ 1b
- 1b ef 8f b7 f9 8f 92 b7 b3 50 ef 40 92 4d bc 92
- Inverse S-Box anwenden: 1b $\Rightarrow$ 44
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Beispiel für Substitute Bytes

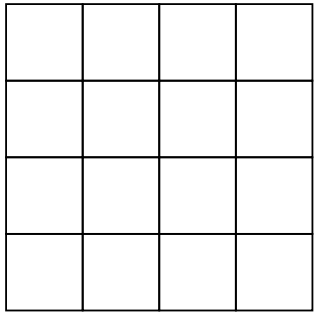
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74
- Zeile 4, Spalte 4 $\Rightarrow$ 1b
- 1b ef 8f b7 f9 8f 92 b7 b3 50 ef 40 92 4d bc 92
- Inverse S-Box anwenden: 1b $\Rightarrow$ 44
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

Beispiel für Substitute Bytes

- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74
- Zeile 4, Spalte 4 $\Rightarrow$ 1b
- 1b ef 8f b7 f9 8f 92 b7 b3 50 ef 40 92 4d bc 92
- Inverse S-Box anwenden: 1b $\Rightarrow$ 44
- 44 61 73 20 69 73 74 20 4b 6c 61 72 74 65 78 74

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit



ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

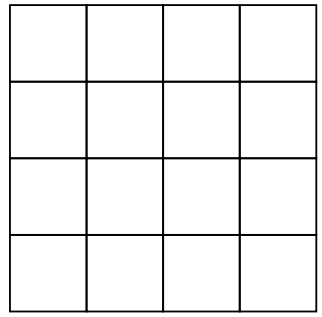
ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation



0 Byte zirkular links

1 Byte zirkular links

2 Bytes zirkular links

3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D			

0 Byte zirkular links

1 Byte zirkular links

2 Bytes zirkular links

3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i		

0 Byte zirkular links

1 Byte zirkular links

2 Bytes zirkular links

3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	

0 Byte zirkular links

1 Byte zirkular links

2 Bytes zirkular links

3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t	0 Byte zirkular links
				1 Byte zirkular links
				2 Bytes zirkular links
				3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t	0 Byte zirkular links
			a	1 Byte zirkular links
				2 Bytes zirkular links
				3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t	0 Byte zirkular links
s			a	1 Byte zirkular links
				2 Bytes zirkular links
				3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t
s	l		a

0 Byte zirkular links

1 Byte zirkular links

2 Bytes zirkular links

3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t	0 Byte zirkular links
s	l	e	a	1 Byte zirkular links
				2 Bytes zirkular links
				3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t
s	l	e	a
		s	

0 Byte zirkular links

1 Byte zirkular links

2 Bytes zirkular links

3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t	0 Byte zirkular links
s	l	e	a	1 Byte zirkular links
		s	t	2 Bytes zirkular links
				3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t	0 Byte zirkular links
s	l	e	a	1 Byte zirkular links
a		s	t	2 Bytes zirkular links
				3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t
s	l	e	a
a	x	s	t

0 Byte zirkular links

1 Byte zirkular links

2 Bytes zirkular links

3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t
s	l	e	a
a	x	s	t

0 Byte zirkular links

1 Byte zirkular links

2 Bytes zirkular links

3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t
s	l	e	a
a	x	s	t

0 Byte zirkular links

1 Byte zirkular links

2 Bytes zirkular links

3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t	0 Byte zirkular links
s	l	e	a	1 Byte zirkular links
a	x	s	t	2 Bytes zirkular links
			r	3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t	0 Byte zirkular links
s	l	e	a	1 Byte zirkular links
a	x	s	t	2 Bytes zirkular links
t			r	3 Bytes zirkular links

ShiftRows für 128bit-Blöcke

ShiftRows am 128bit Beispiel

Datenblock 16 Oktette = 128Bit

D	i	K	t
a	s	l	e
s	t	a	x
		r	t

ShiftRows Transformation

D	i	K	t
s	l	e	a
a	x	s	t
t			r

0 Byte zirkular links

1 Byte zirkular links

2 Bytes zirkular links

3 Bytes zirkular links

allgemein:

s00	s01	s02	s03
s10	s11	s12	s13
s20	s21	s22	s23
s30	s31	s32	s33

s00	s01	s02	s03
s11	s12	s13	s10
s22	s23	s20	s21
s33	s30	s31	s32

ShiftRows für 128bit-Blöcke

Mix-Columns

Mix-Columns

- Der Mix-Columns-Algorithmus verknüpft die Bytes einer Spalte untereinander, also *vertikal*.

Mix-Columns

- Der Mix-Columns-Algorithmus verknüpft die Bytes einer Spalte untereinander, also *vertikal*.
- Jedes Chiffre-Byte hängt somit von *allen* Klartextbytes der Spalte ab.

Mix-Columns

- Der Mix-Columns-Algorithmus verknüpft die Bytes einer Spalte untereinander, also *vertikal*.
- Jedes Chiffre-Byte hängt somit von *allen* Klartextbytes der Spalte ab.
- Durch den Shift-Rows-Algorithmus werden die Bytes in jeder Runde horizontal ausgetauscht.

Mix-Columns

- Der Mix-Columns-Algorithmus verknüpft die Bytes einer Spalte untereinander, also *vertikal*.
- Jedes Chiffre-Byte hängt somit von *allen* Klartextbytes der Spalte ab.
- Durch den Shift-Rows-Algorithmus werden die Bytes in jeder Runde horizontal ausgetauscht.
- Die Kombination aus Shift-Rows und Mix-Columns führt dazu, dass nach der Rundenzahl (z.B. 10 bei 128bit AES) ein Chiffre-Byte mehrfach von allen Klartext-Bytes abhängt.

Mix-Columns

- Der Mix-Columns-Algorithmus verknüpft die Bytes einer Spalte untereinander, also *vertikal*.
- Jedes Chiffre-Byte hängt somit von *allen* Klartextbytes der Spalte ab.
- Durch den Shift-Rows-Algorithmus werden die Bytes in jeder Runde horizontal ausgetauscht.
- Die Kombination aus Shift-Rows und Mix-Columns führt dazu, dass nach der Rundenzahl (z.B. 10 bei 128bit AES) ein Chiffre-Byte mehrfach von allen Klartext-Bytes abhängt.
- Ändert sich 1 bit im Klartext, ändern sich etwas 50% aller Bits in der Chiffre und umgekehrt!

Mix-Columns

- Der Mix-Columns-Algorithmus verknüpft die Bytes einer Spalte untereinander, also *vertikal*.
- Jedes Chiffre-Byte hängt somit von *allen* Klartextbytes der Spalte ab.
- Durch den Shift-Rows-Algorithmus werden die Bytes in jeder Runde horizontal ausgetauscht.
- Die Kombination aus Shift-Rows und Mix-Columns führt dazu, dass nach der Rundenzahl (z.B. 10 bei 128bit AES) ein Chiffre-Byte mehrfach von allen Klartext-Bytes abhängt.
- Ändert sich 1 bit im Klartext, ändern sich etwas 50% aller Bits in der Chiffre und umgekehrt!
- Diese Eigenschaft von Verschlüsselungsalgorithmen wird *Diffusion* genannt.

Mix-Columns

Mix-Columns

- Bei Mix-Columns wird eine Spalte als *Vektor* betrachtet und mit einer speziellen *Matrix* multipliziert.

Mix-Columns

- Bei Mix-Columns wird eine Spalte als *Vektor* betrachtet und mit einer speziellen *Matrix* multipliziert.
- Dabei werden spezielle Mutlitplikations- und Additionsoperationen angewendet, bei denen das Ergebnis immer als ein Byte dargestellt werden kann, also eine Zahl zwischen 0 und 255 ist (Galois-Körper).

Mix-Columns

- Bei Mix-Columns wird eine Spalte als *Vektor* betrachtet und mit einer speziellen *Matrix* multipliziert.
- Dabei werden spezielle Mutlitplikations- und Additionsoperationen angewendet, bei denen das Ergebnis immer als ein Byte dargestellt werden kann, also eine Zahl zwischen 0 und 255 ist (Galois-Körper).
- Das Multiplizieren des Vektors mit der Matrix und die Reduktion auf 8bit ist umkehrbar, damit eine Entschlüsselung möglich wird.

Mix-Columns

- Bei Mix-Columns wird eine Spalte als *Vektor* betrachtet und mit einer speziellen *Matrix* multipliziert.
- Dabei werden spezielle Mutlitplikations- und Additionsoperationen angewendet, bei denen das Ergebnis immer als ein Byte dargestellt werden kann, also eine Zahl zwischen 0 und 255 ist (Galois-Körper).
- Das Multiplizieren des Vektors mit der Matrix und die Reduktion auf 8bit ist umkehrbar, damit eine Entschlüsselung möglich wird.
- alle Rechenoperationen sind wegen der Binär-Arithmetik sehr einfach oder lassen sich über Tabellen schnell bestimmen.

Mix-Columns

- Bei Mix-Columns wird eine Spalte als *Vektor* betrachtet und mit einer speziellen *Matrix* multipliziert.
- Dabei werden spezielle Mutlitplikations- und Additionsoperationen angewendet, bei denen das Ergebnis immer als ein Byte dargestellt werden kann, also eine Zahl zwischen 0 und 255 ist (Galois-Körper).
- Das Multiplizieren des Vektors mit der Matrix und die Reduktion auf 8bit ist umkehrbar, damit eine Entschlüsselung möglich wird.
- alle Rechenoperationen sind wegen der Binär-Arithmetik sehr einfach oder lassen sich über Tabellen schnell bestimmen.
- Moderne Prozessoren enthalten ein spezielles AES-Rechenwerk, dass die Chiffrierung/Dechiffrierung in Hardware umsetzt.

Multiplikation Matrix mit Vektor

Multiplikation einer Matrix mit einem Vektor:

b0	b1	b2	b3

a00	a01	a02	a03	b0	=	a00·b0 + a01·b1 + a02·b2 + a03·b3
a10	a11	a12	a13	b1		a10·b0 + a11·b1 + a12·b2 + a13·b3
a20	a21	a22	a23	b2		a20·b0 + a21·b1 + a22·b2 + a23·b3
a30	a31	a32	a33	b3		a30·b0 + a31·b1 + a32·b2 + a33·b3

Multiplikation Matrix mit Vektor

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

MixColumns-Matrix beim Verschlüsseln

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

inverse MixColumns-Matrix beim Entschlüsseln

Multiplikation Matrix mit Vektor

Multiplikation einer Matrix mit einem Vektor:

b0	b1	b2	b3

a00	a01	a02	a03	b0	=	a00·b0 + a01·b1 + a02·b2 + a03·b3
a10	a11	a12	a13	b1		a10·b0 + a11·b1 + a12·b2 + a13·b3
a20	a21	a22	a23	b2		a20·b0 + a21·b1 + a22·b2 + a23·b3
a30	a31	a32	a33	b3		a30·b0 + a31·b1 + a32·b2 + a33·b3

Multiplikation Matrix mit Vektor

MixColumns-Matrix beim Verschlüsseln

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

inverse MixColumns-Matrix beim Entschlüsseln

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

Multiplikation Matrix mit Vektor

Multiplikation einer Matrix mit einem Vektor:

b0	b1	b2	b3

a00	a01	a02	a03		b0	a00·b0 + a01·b1 + a02·b2 + a03·b3
a10	a11	a12	a13	×	b1	a10·b0 + a11·b1 + a12·b2 + a13·b3
a20	a21	a22	a23		b2	a20·b0 + a21·b1 + a22·b2 + a23·b3
a30	a31	a32	a33		b3	a30·b0 + a31·b1 + a32·b2 + a33·b3

Multiplikation Matrix mit Vektor

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

MixColumns-Matrix beim Verschlüsseln

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

inverse MixColumns-Matrix beim Entschlüsseln

Multiplikation Matrix mit Vektor

Multiplikation einer Matrix mit einem Vektor:

b0	b1	b2	b3

a00	a01	a02	a03		b0	a00·b0 + a01·b1 + a02·b2 + a03·b3
a10	a11	a12	a13	×	b1	a10·b0 + a11·b1 + a12·b2 + a13·b3
a20	a21	a22	a23		b2	a20·b0 + a21·b1 + a22·b2 + a23·b3
a30	a31	a32	a33		b3	a30·b0 + a31·b1 + a32·b2 + a33·b3

Multiplikation Matrix mit Vektor

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

MixColumns-Matrix beim Verschlüsseln

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

inverse MixColumns-Matrix beim Entschlüsseln

Multiplikation Matrix mit Vektor

Multiplikation einer Matrix mit einem Vektor:

b0	b1	b2	b3

a00	a01	a02	a03	b0	=	a00·b0 + a01·b1 + a02·b2 + a03·b3
a10	a11	a12	a13	b1		a10·b0 + a11·b1 + a12·b2 + a13·b3
a20	a21	a22	a23	b2		a20·b0 + a21·b1 + a22·b2 + a23·b3
a30	a31	a32	a33	b3		a30·b0 + a31·b1 + a32·b2 + a33·b3

Multiplikation Matrix mit Vektor

MixColumns-Matrix beim Verschlüsseln

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

inverse MixColumns-Matrix beim Entschlüsseln

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

Multiplikation Matrix mit Vektor

Multiplikation einer Matrix mit einem Vektor:

b0	b1	b2	b3

a00	a01	a02	a03	b0	=	a00·b0 + a01·b1 + a02·b2 + a03·b3
a10	a11	a12	a13	b1		a10·b0 + a11·b1 + a12·b2 + a13·b3
a20	a21	a22	a23	b2		a20·b0 + a21·b1 + a22·b2 + a23·b3
a30	a31	a32	a33	b3		a30·b0 + a31·b1 + a32·b2 + a33·b3

Multiplikation Matrix mit Vektor

MixColumns-Matrix beim Verschlüsseln

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

inverse MixColumns-Matrix beim Entschlüsseln

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

Multiplikation Matrix mit Vektor

Multiplikation einer Matrix mit einem Vektor:

b0	b1	b2	b3

a00	a01	a02	a03	b0	a00·b0 + a01·b1 + a02·b2 + a03·b3
a10	a11	a12	a13	b1	a10·b0 + a11·b1 + a12·b2 + a13·b3
a20	a21	a22	a23	b2	a20·b0 + a21·b1 + a22·b2 + a23·b3
a30	a31	a32	a33	b3	a30·b0 + a31·b1 + a32·b2 + a33·b3

Multiplikation Matrix mit Vektor

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

MixColumns-Matrix beim Verschlüsseln

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

inverse MixColumns-Matrix beim Entschlüsseln

Multiplikation Matrix mit Vektor

Multiplikation einer Matrix mit einem Vektor:

b0	b1	b2	b3

a00	a01	a02	a03	b0	=	a00·b0 + a01·b1 + a02·b2 + a03·b3
a10	a11	a12	a13	b1		a10·b0 + a11·b1 + a12·b2 + a13·b3
a20	a21	a22	a23	b2		a20·b0 + a21·b1 + a22·b2 + a23·b3
a30	a31	a32	a33	b3		a30·b0 + a31·b1 + a32·b2 + a33·b3

Multiplikation Matrix mit Vektor

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

MixColumns-Matrix beim Verschlüsseln

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

inverse MixColumns-Matrix beim Entschlüsseln

Multiplikation Matrix mit Vektor

Multiplikation einer Matrix mit einem Vektor:

b0	b1	b2	b3

a00	a01	a02	a03	b0	=	a00·b0 + a01·b1 + a02·b2 + a03·b3
a10	a11	a12	a13	b1		a10·b0 + a11·b1 + a12·b2 + a13·b3
a20	a21	a22	a23	b2		a20·b0 + a21·b1 + a22·b2 + a23·b3
a30	a31	a32	a33	b3		a30·b0 + a31·b1 + a32·b2 + a33·b3

Multiplikation Matrix mit Vektor

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

MixColumns-Matrix beim Verschlüsseln

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

inverse MixColumns-Matrix beim Entschlüsseln

Multiplikation Matrix mit Vektor

Multiplikation einer Matrix mit einem Vektor:

b0	b1	b2	b3

a00	a01	a02	a03	b0	=	a00·b0 + a01·b1 + a02·b2 + a03·b3
a10	a11	a12	a13	b1		a10·b0 + a11·b1 + a12·b2 + a13·b3
a20	a21	a22	a23	b2		a20·b0 + a21·b1 + a22·b2 + a23·b3
a30	a31	a32	a33	b3		a30·b0 + a31·b1 + a32·b2 + a33·b3

Multiplikation Matrix mit Vektor

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

MixColumns-Matrix beim Verschlüsseln

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

inverse MixColumns-Matrix beim Entschlüsseln