AES

Advanced Encryption Standard

Inhalt

ENTSTEHUNG

Advanced Encryption Standard

Entwickelt von zwei belgischen Kryptographen, Vincent Rijmen und Joan Daemen

Oktober 2000 ausgewählt nach Wettbewerb um DES (Data Encryption Standard) Ablösung

Beruht auf Rijndael Block Cipher, wurde aber leicht modifiziert mit fixen Schlüsselgrössen

EIGENSCHAFTEN

Advanced Encryption Standard

Übersicht

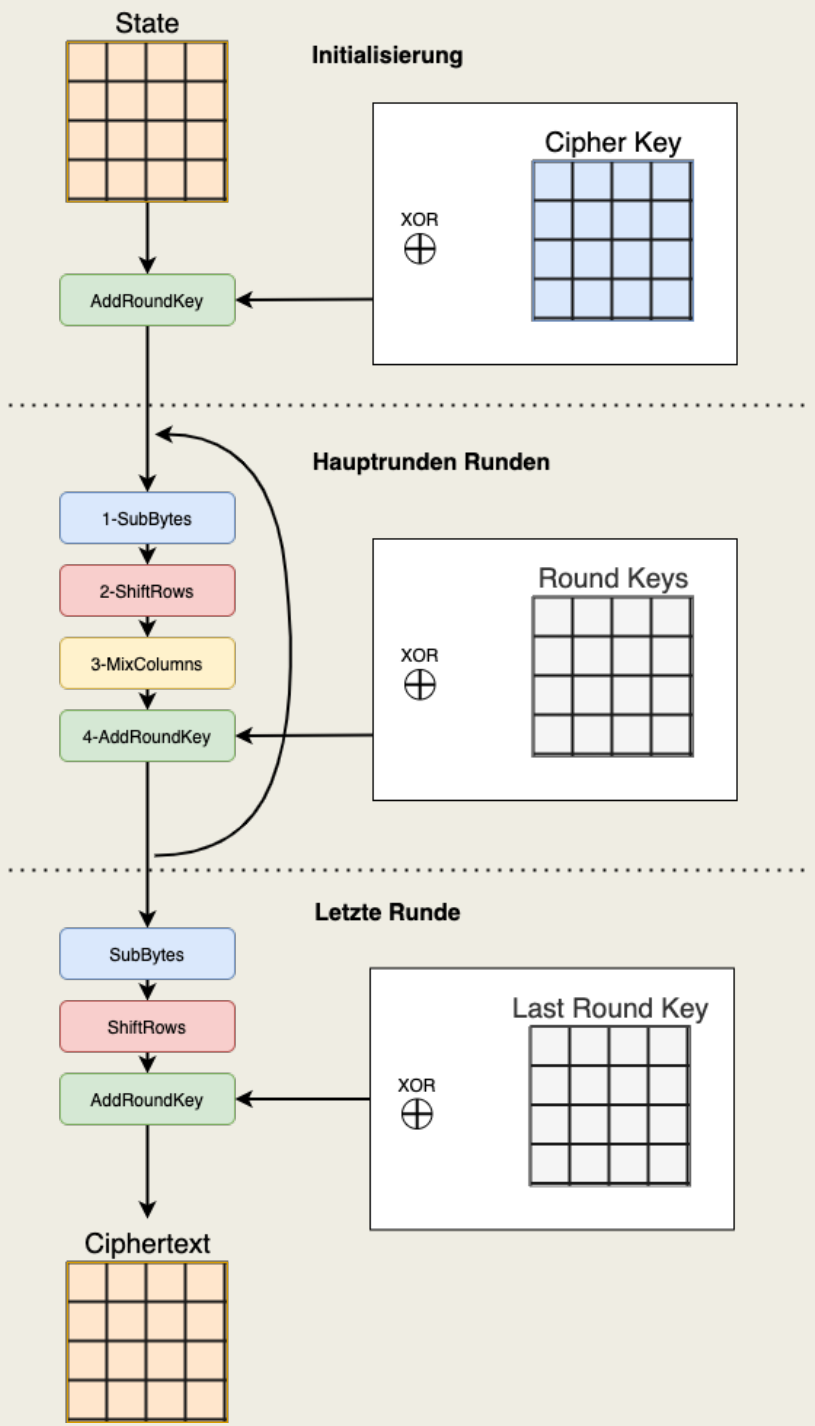
- Feste Blockgrösse 128 Bit
- Variable Schlüssellänge von 128 Bit, 192 Bit und 256 Bit – Symmetrisch verschlüsselt
- Variable Anzahl Runden gemäss Schlüssellänge
  - *128 Bit - 10 Runden, 192 Bit – 12 Runden, 256 Bit – 14 Runden*
- Operationen pro Runde
  - *SubByte*
  - *ShiftRow*
  - *MixColumn*
  - *AddRoundKey*
- Geschwindigkeit: Gleichmässig gute Performance über mehrere Plattformen wie z.b. 64-Bit, 32-Bit Prozessoren oder 8-Bit Mikrocontroller
- Speicherbedarf: Sehr geringe RAM- und ROM-Speicher Bedarf. Schlüsselerzeugung kann «on-the-fly» stattfinden. Ideal für Chipkarten.

FUNKTIONSWEISE

Advanced Encryption Standard

Übersicht

- Ein 128-Bit Block (State) wird eingelesen
- State wird mit dem Cipher Key XOR-Verknüpft
- Die Operationen SubByte, ShiftRows, MixColumns und AddRoundKey werden gemäss vorgegebener Anzahl Runden durchgeführt
- Der aktuelle State wird erneut XOR-Verknüpft mit dem Round Key
- In der letzten Runde wird die Operation MixColumns ausgelassen
- Die Ausgabe ist ein Ciphertext der mit den selben Cipher Key entschlüsselt werden muss



Beispiel Eingabe

Hexadecimalblock aus
Eingabe in 128 Bit

32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

Binärwerte = 8 4 2 1 | 8 4 2 1

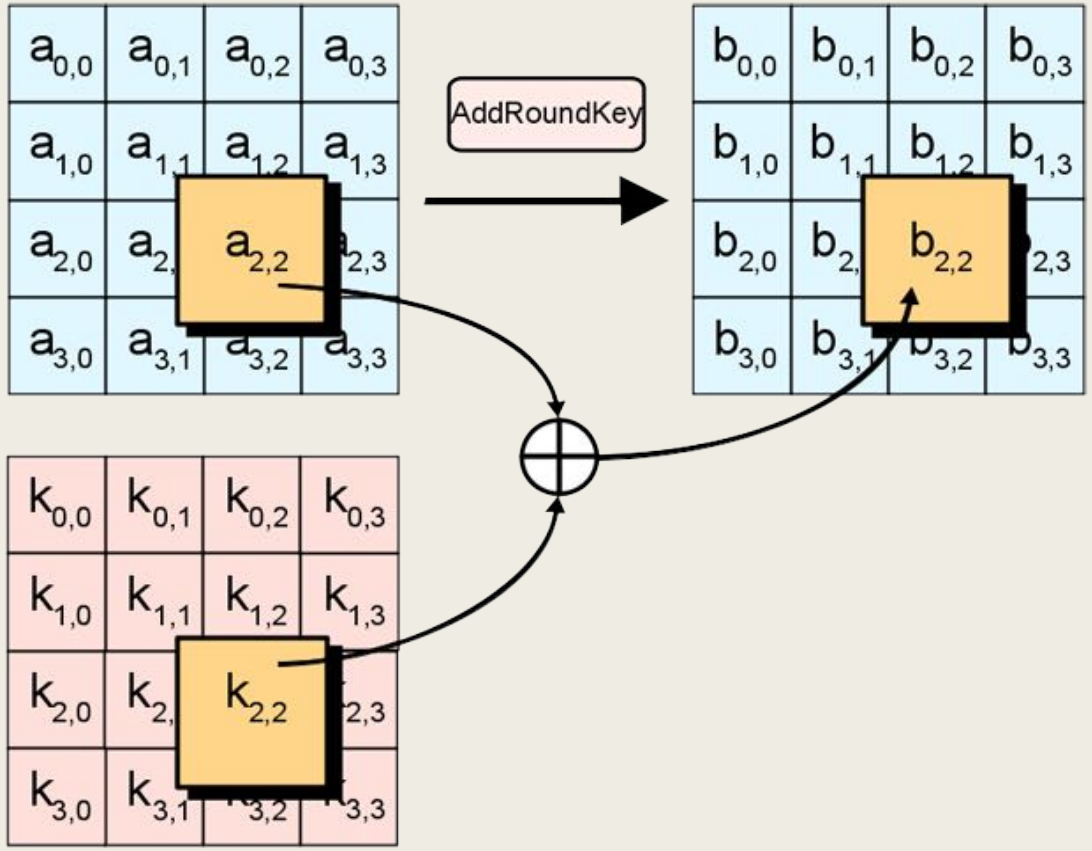
32 = 00110010 ← 1 Byte

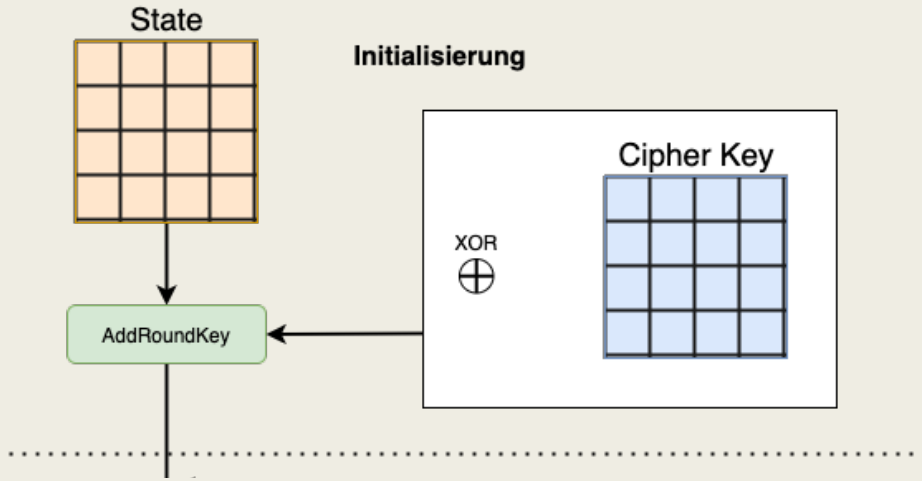
3 hex 2 hex

Hexadecimal = 0 1 2 3 5 6 7 8 9 a b c d e f

AddRoundKey - Initial

- Vor Beginn der Hauptrunden wird die Eingabe mit dem Cipher Key XOR-Verknüpft





AddRoundKey - Initial

State

32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

Cipher Key

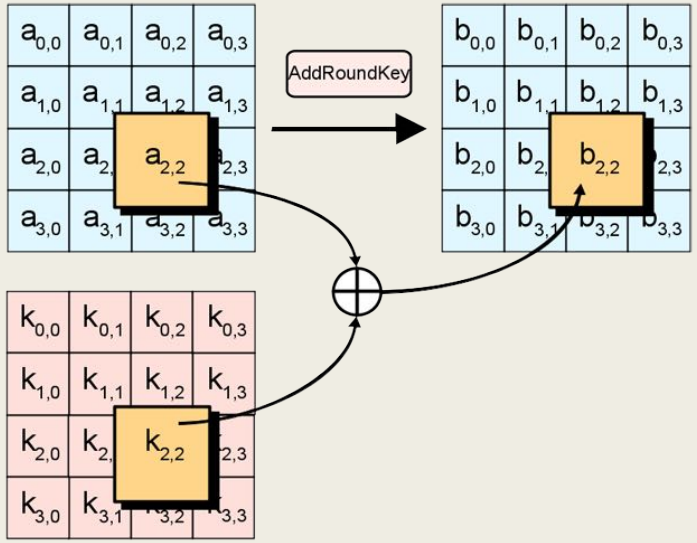
2b	28	ab	09
73	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

32 = 00110010

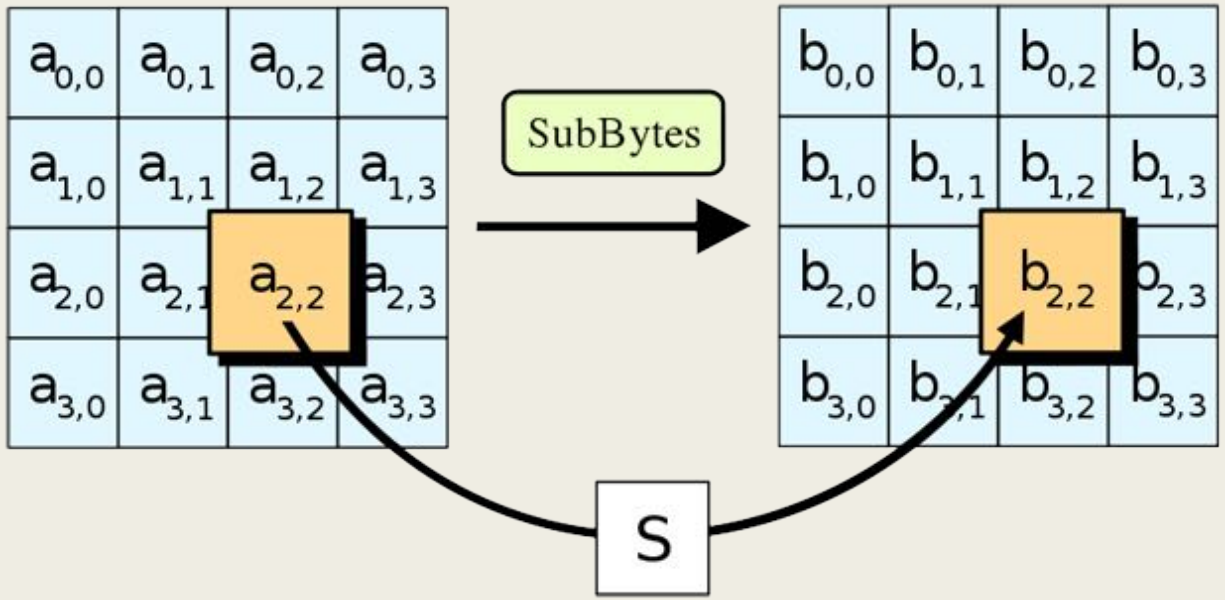
2b = 00101011

$\oplus$ = 00011001 = 19

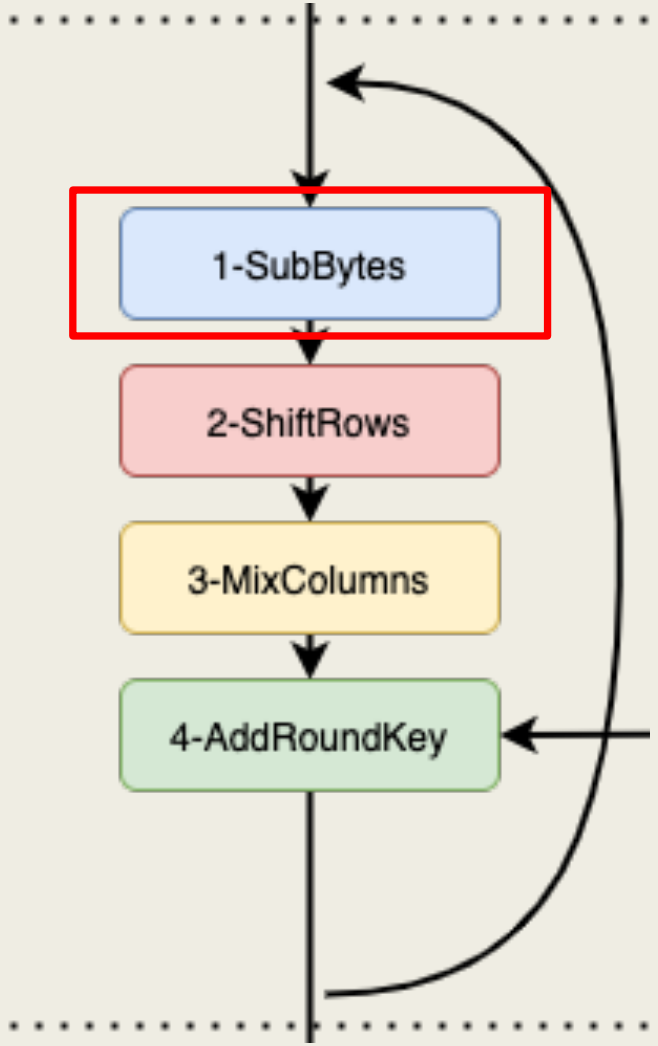


SubBytes

- State wird durch eine Rijndael S-Box substituiert
- S-Box ist unabhängig von der Eingabe
- Wird in vorberechneter Form verwendet falls genügend Speicher vorhanden ist (256 Bytes)



S = S-Box



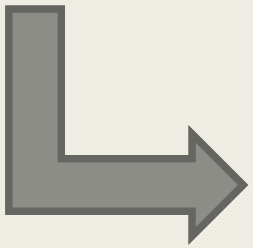
S-Box

19	→	d4
3d	→	27
e3	→	11
be	→	ae

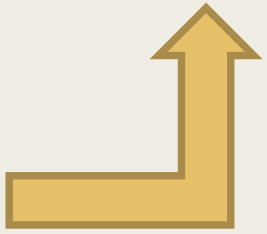
Hex	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3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

SubBytes

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

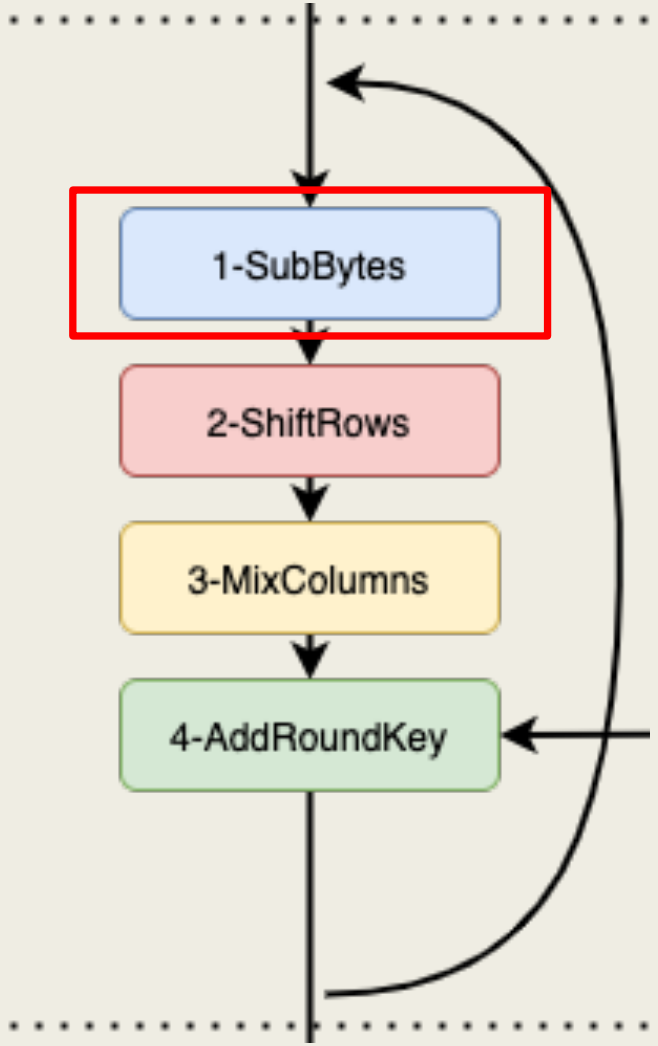


S-Box



d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

Der neue State wird weitergegeben an ShiftRows.



ShiftRows

- Jede Zeile des State wird um ihre Zeilennummer nach links verschoben.
- 0, 1, 2, 3 Bytes nach links