

Verschlüsselung heute

- DES (Data Encryption Standard)
 - US-Standard ab 1976
 - Algorithmus unter Beteiligung der NSA entwickelt ¹
 - Schlüssellänge von 56 Bits
 - Blockchiffre, rundenbasiert
 - Heute noch im Einsatz (z. B. bei Geldautomaten)
- AES (Advanced Encryption Standard)
 - Nachfolger von DES
 - Blockchiffre, rundenbasiert
 - Größere Schlüssellängen

¹Das fanden nicht alle gut...

- Arbeitet auf Blöcken der Länge 128 Bit
- Mögliche Schlüssellängen: 128 Bit, 192 Bit, 256 Bit
- Aktueller Zustand (state) speichert einen (bearbeiteten) Block “tabellarisch”
 - Vier Zeilen
 - Vier Spalten
 - Ein Byte pro Zelle (und somit insgesamt 16 Byte = 128 Bit)
- Rundenbasiert
 - Verwendung eines neuen Rundenschlüssels pro Runde
 - Rundenzahl abhängig von der Schlüssellänge
 - Jede Runde verändert den aktuellen Zustand

Schaue Dir hier den Ablauf von AES an

- Einlesen eines Blocks aus der Eingabe
- Erzeugen der Rundenschlüssel (Key Expansion)
- Durchführen der einzelnen Runden
 1. SubBytes
 2. ShiftRows
 3. MixColumns (nicht bei letzter Runde)
 4. AddRoundKey
- Schreiben des Blocks in die Ausgabe

- Jedes Byte des aktuellen Zustands wird durch ein anderes Byte ersetzt
- Die Ersetzung wird durch eine sogenannte S-Box vorgegeben
- Die S-Box ist vordefiniert und **nicht** geheim!

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	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

Figure 7. S-box: substitution values for the byte xy (in hexadecimal format).

Abbildung 1: Die S-Box (Grafik aus dem Originalpaper)

- Zyklischer Byte-Shift der letzten drei Zeilen des Zustands

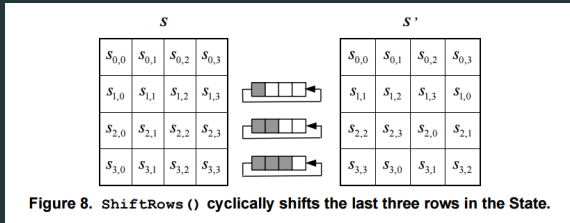


Abbildung 2: Zyklischer Byteshift (Grafik aus dem Originalpaper)

- Multiplikation jeder Spalte mit einer (vordefinierten) Matrix

- XOR-Verknüpfung jeder Spalte des Zustands mit einem Teil des aktuellen Rundenschlüssels

Erzeugen der Rundenschlüssel

- Alle Rundenschlüssel werden aus dem ursprünglich gewählten Schlüssel gewonnen
- Die Schlüsselteile werden erzeugt durch
 - S-Box
 - Byte-Rotation
 - XOR-Verknüpfung

- Anwendung der jeweils inversen Operationen in der richtigen Reihenfolge
 1. Inverse S-Box
 2. Rotation der Zeilen in die entgegengesetzte Richtung
 3. Multiplikation mit inverser Matrix
 4. AddRoundKey ist selbstinvers

- Die einzelnen Schritte des Verfahrens sind nicht allzu kompliziert
- Die Beurteilung der Sicherheit des Verfahrens ist Laien (auch mit Krypto-Grundkenntnissen) nicht mehr möglich

- Jedes betrachtete Verfahren setzt voraus, dass sich die beteiligten Kommunikationspartner auf einen gemeinsamen Schlüssel einigen
- Der Schlüssel sollte naheliegenderweise nicht über einen potentiell unsicheren Kanal übermittelt werden
 - Oder geht das doch irgendwie?