Advance Encryption Standard

AES Events in Chronological Order

- NIST issued call for a standard cipher in 1997
 - international
 - 15 candidates (out of 21) accepted in Jun 98
- A shortlist of 5 selected in Aug-99
 - MARS (IBM)
 - RC6 (USA)
 - Rijndael (Belgium)
 - Serpent (Europe)
 - Twofish (USA)
- Europe vs. USA
- commercial vs. academic
 - US based ones were all of commercial origin
- Rijndael (from Belgium) was selected as the AES in Oct-2000
 - issued as FIPS PUB 197 standard in Nov-2001

AES Evaluation Criteria

- final criteria (used to select the winner)
 - general security
 - NIST relied on evaluation done by cryptographic community
 - software implementation performance
 - execution speed, performance across different platforms (8 to 64 bit platforms), variation of speed with key size
 - hardware implementation
 - not only timings, but also cost is important
 - especially for restricted space environments (such as smartcards)
 - implementation (timing and power) attacks
 - key agility
 - ability to change keys quickly and with minimum resources

AES Requirements

- private key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- stronger & faster than Triple-DES
- active life of 20-30 years
- both C & Java implementations
- provide full specification and design details

The AES Cipher - Rijndael

- designed by Vincent Rijmen and Joan Daemen in Belgium (UCL)
- Characteristics
 - an iterative rather than feistel cipher
 - processes data as block of 4 columns of 4 bytes
 - operates on entire data block in every round
 - 128/192/256 bit keys, 128 bit data
 - expanded key size of 44, 52 or 60 words
 - algorithm is <u>Not</u> a Feistel structure
 - processes entire data block in parallel
 - designed to be
 - resistant against known attacks
 - speed and code compactness on many CPUs
 - design simplicity

The AES Cipher vrs Rijndael

- AES is not precisely Rijndael
- Rijndael
 - supports a larger range of block and key sizes
 - the key and block sizes in any multiple of 32 bits, with a minimum of 128 bits and a maximum of 256 bits.
- > AES
 - has a fixed block size of 128 bits and a key size of 128, 192 or 256 bits,

AES - overview

- > Hence, overall
 - an initial round (AddRoundKey),
 - r standard rounds, r is 10,12 or 14
 - The first r-1 rounds are similar consisting of
 - ByteSub
 - ShiftRow
 - MixColumn
 - AddRoundKey
 - The last round only perform the transformations
 - ByteSub
 - ShiftRow
 - AddRoundKey

AES - overview

- Simple Repeating structure
- Cipher begins and ends with Add Round Key,
 - forms a Vernam Cipher or "One Time Pad"
 - any other stage applied at the beginning or end is reversible without the key
- other three stages provide confusion, diffusion and nonlinearity

GF(28)

Byte b₇b₆b₅b₄b₃b₂b₁b₀ will have the representation as

$$b(x) = b_7 x^7 + b_6 x^6 + b_5 x^5 + b_4 x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0$$

> Therefore, 01010111 would have the representation as

$$01010111 \rightarrow x^6 + x^4 + x^2 + x + 1$$

AES inputs

Input

State

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



to Encryption Process

Cipher Key

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c



to Key Schedule

State array from input

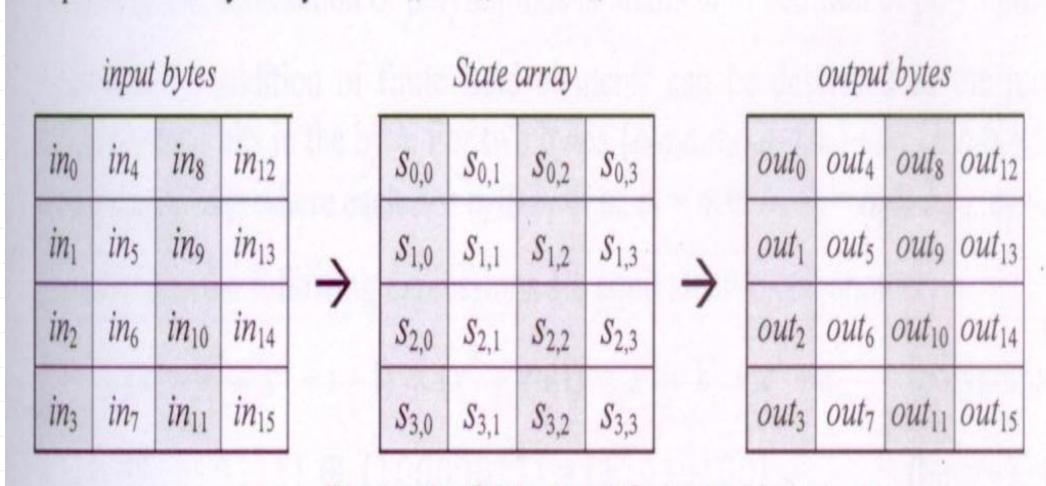
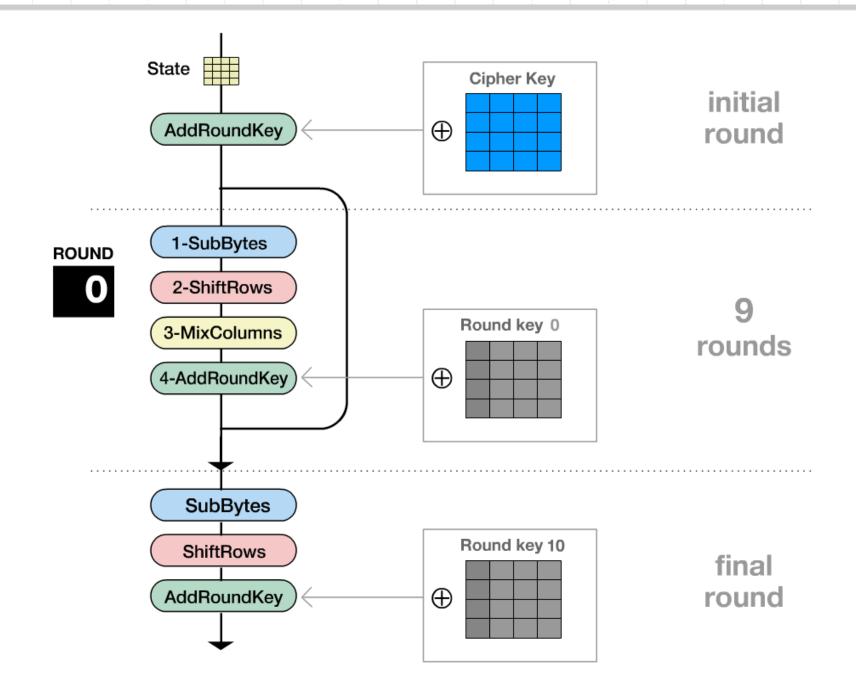
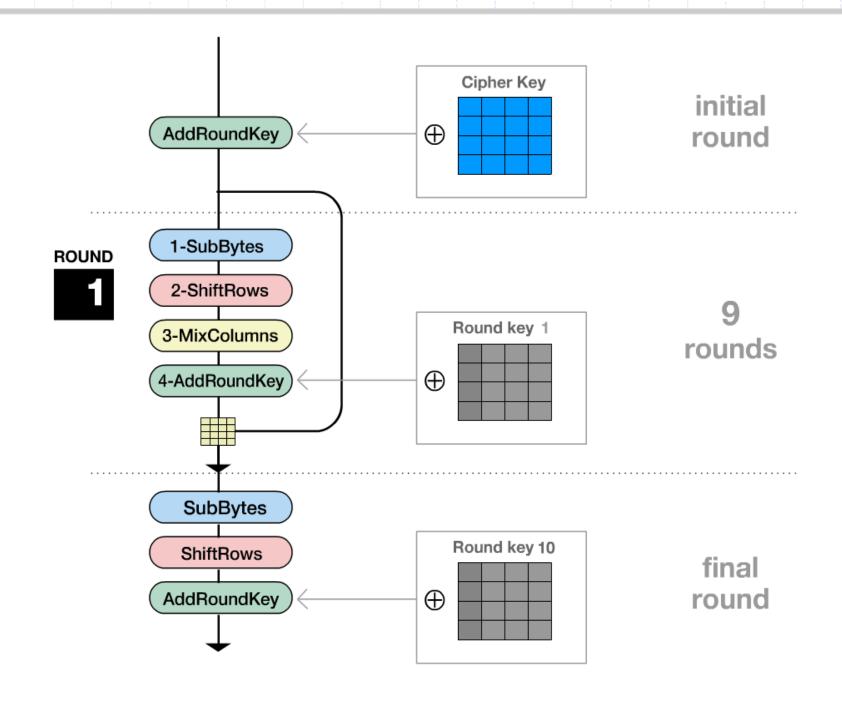
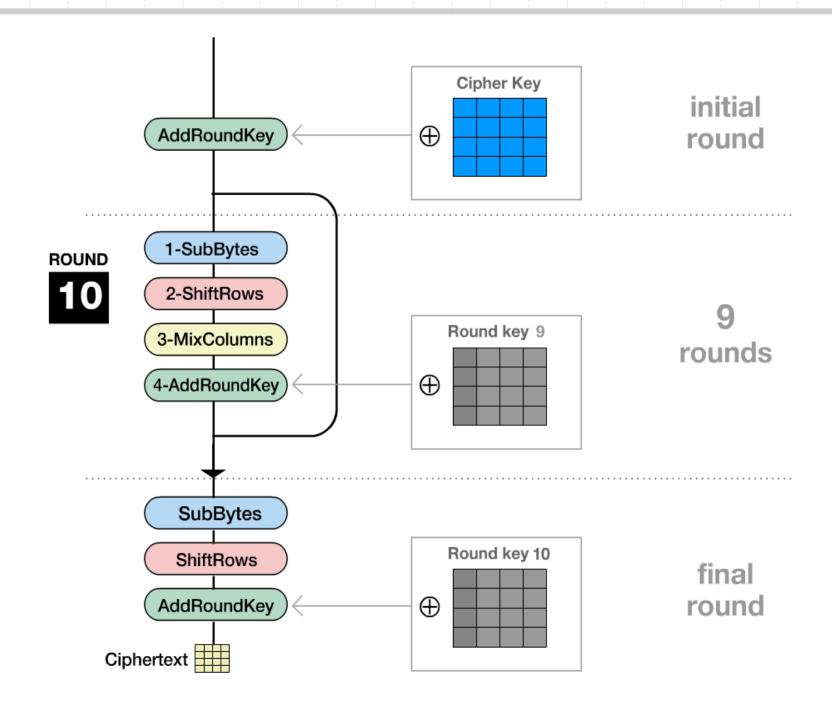
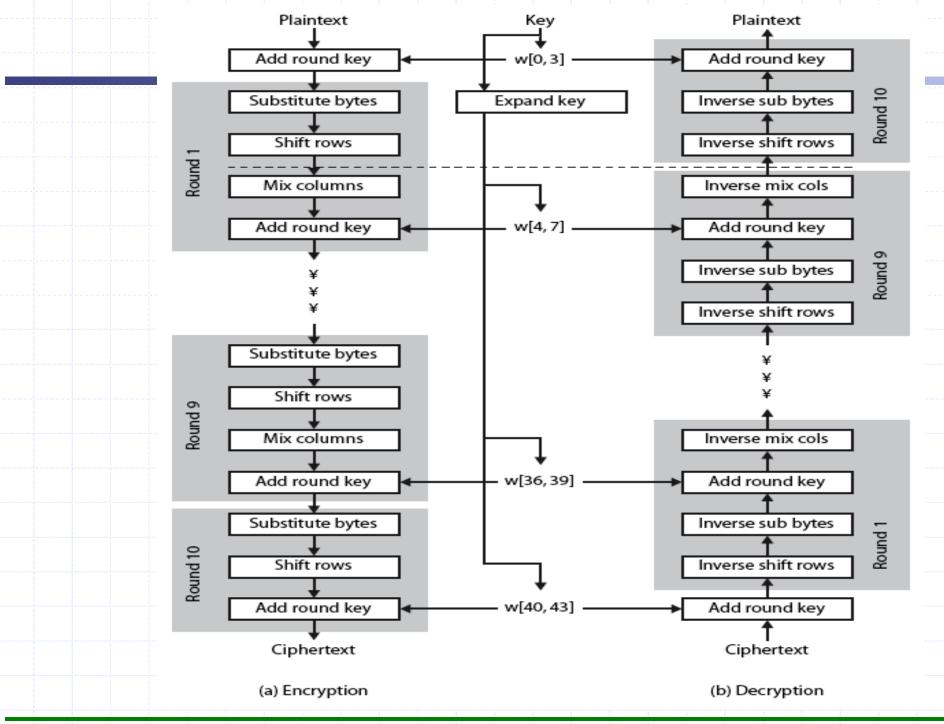


Figure 3. State array input and output.









AES pseudocode

```
#define Nb 4 /* for blocksize=keysize=128 bits and Nr=10 */
Rijndael (byte in [4*Nb], byte out [4*Nb], word
                                            ExpandedKey[Nb*(Nr+1)]))
   byte state[4, Nb];
    int i, j, r, c;
    for (i=0; i<=4; i++) {
        for (j=0; j<=Nb; j++)
                state[i,j] = in[i + 4j];
    KeyExpansion(CipherKey, ExpandedKey);
    AddRoundKey(state, ExpandedKey);
    for(i=1; i<N<sub>r</sub>; i++)
        Round(State, ExpandedKey[i]);
    FinalRound(State, ExpandedKey[N]);
    out = state;
```

Word

a _{0,0}	a _{0,1}	a _{0,2}	a _{0,3}
a _{1,0}	a _{1,1}	a _{1,2}	a _{1,3}
a _{2,0}	a _{2,1}	a _{2,2}	a _{2,3}
a _{3,0}	a _{3,1}	a _{3,2}	a _{3,3}

AES pseudocode

```
Round (State, ExpandedKey[i])
   SubBytes (State);
   ShiftRows(State);
   MixColumns (State);
   AddRoundKey(State, ExpandedKey[i]);
FinalRound(State, ExpandedKey[N])
   SubBytes (State);
   ShiftRows(State);
   AddRoundKey (State, ExpandedKey [N,]);
```

AES illustration - input key value

Input

State

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



to Encryption Process

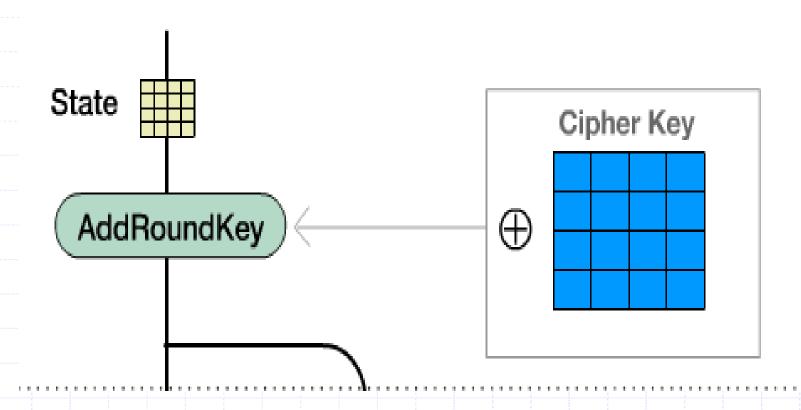
Cipher Key

2b	28	ab	09
7e	ae	£7	cf
15	d2	15	4f
16	a6	88	3c



to Key Schedule

First: the Initial Round



initial round

First: the Initial Round.....

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





19	a0	9a	е9
3d	f4	С6	f8
е3	e2	8d	48
be	2b	2a	08

Cipher Key

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

Now Round transformations

4 transformations:

1-SubBytes

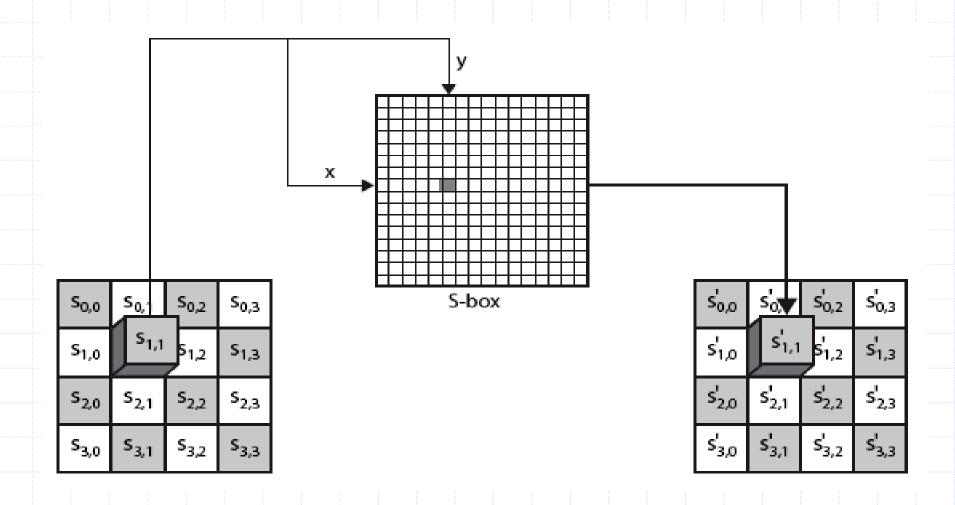
2-ShiftRows

3-MixColumns

4-AddRoundKey

9 times

- Appears as a simple substitution of each byte
 - uses one table of 16x16 bytes containing a permutation of all 256 8-bit values
 - each byte of state is replaced by byte indexed by row (left 4-bits) & column (right 4-bits)
 - eg. byte {95} is replaced by byte in row 9 column 5
 - which has value {2A}
- However, S-box is constructed
 - using defined algebraic transformations of values in GF(28)
 - designed to be resistant to all known attacks



19	a0	9a	e9
3d	f4	С6	f8
e3	e2	8d	48
be	2b	2a	08

h	ex									7							
"	EX	0	1	2	3	4	5	6	7	8	9	a	b	С	d	е	f
	0	63	7c	77	7b	f2	6b	6f	с5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	с9	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
, x	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
ı î	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
	а	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	80
	С	ba	78	25	2e	1c	a6	b4	с6	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
	е	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	се	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

a09ae93df4c6f8e3e28d48be2b2a08

19

l l	ex									Y							
1.	EX	0	1	2	3	4	5	6	7	8	9	a	b	С	d	е	f
	0	63	7c	77	7b	f2	6b	6f	с5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	с9	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
١,	. 7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
ľ	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
	а	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	80
	С	ba	78	25	2e	1c	a6	b4	с6	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
	е	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e 6	42	68	41	99	2d	0f	b0	54	bb	16

a09ae93df4c6f8e3e28d48be2b2a08

19

h	ex								3	7							
**	e.	0	1	2	3	4	5	6	7	8	9	а	b	С	d	е	f
	0	63	7c	77	7b	f2	6b	6f	с5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	с9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	00	34	a5	e 5	f1	71	d8	31	15
	3	04	c 7	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
l î	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
	а	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	đ5	4e	a9	6c	56	f4	ea	65	7a	ae	80
	С	ba	78	25	2e	1c	a6	b4	с6	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
	е	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

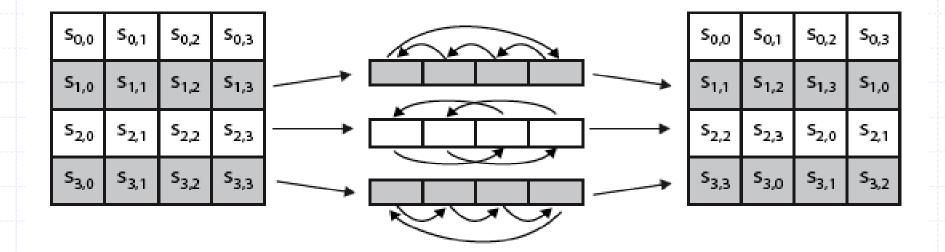
d4	a0	9a	e9
3d	f4	С6	f8
e3	e2	8d	48
be	2b	2a	08

١,	ex									7							
ľ	iex	0	1	2	3	4	5	6	7	8	9	а	b	С	d	е	f
	0	63	7c	77	7b	f2	6b	6f	с5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	с9	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	e 3	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
١,	, 7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
1	` 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
	а	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	đ5	4e	a9	6c	56	f4	ea	65	7a	ae	80
	С	ba	78	25	2e	1c	a6	b4	с6	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
	е	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	се	55	28	df
	f	8c	al	89	0d	bf	e 6	42	68	41	99	2d	0f	b0	54	bb	16

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

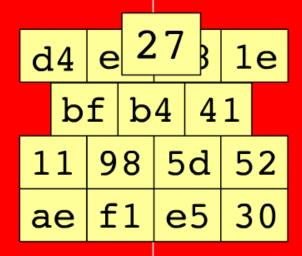
he									3	Z.							
116		0	1	2	3	4	5	6	7	8	9	а	b	С	d	е	f
	0	63	7c	77	7b	f2	6b	6f	с5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	с9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	00	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	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
1^	8	cd	00	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
	а	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e 7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	80
	С	ba	78	25	2e	1c	a6	b4	с6	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
	е	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	се	55	28	df
	f	8c	al	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

- A circular byte shift in each each
 - 1st row is unchanged
 - 2nd row does 1 byte circular shift to left
 - 3rd row does 2 byte circular shift to left
 - 4th row does 3 byte circular shift to left
- decrypt inverts using shifts to right
- since state is processed by columns, this step permutes bytes between the columns



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

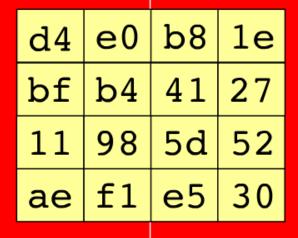




..... rotate over 1 byte

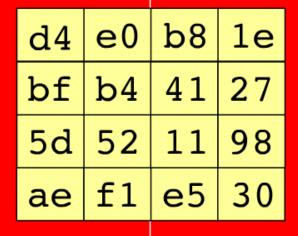
d4	e0	b8	1e
bf	b4	41	27
11	98	5d	52
ae	f1	e5	30





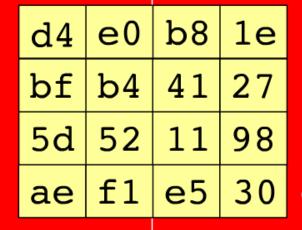
rotate over 2 bytes





rotate over 2 bytes





30

image: rotate over 3 bytes

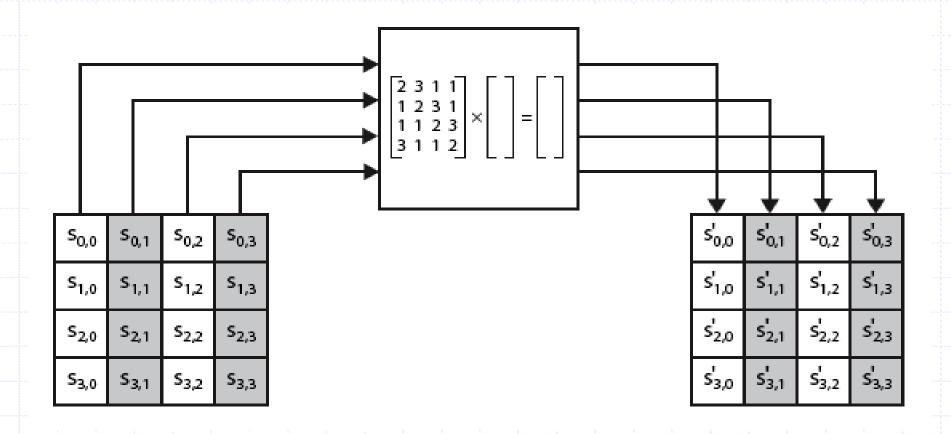
AES Round - Shift Rows transformation



f1 e5 rotate over 3 bytes

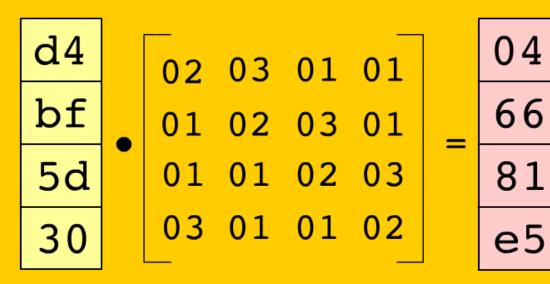
- the MixColumns stage is a substitution that makes use of arithmetic over GF(2^8).
- > each column is processed separately
- each byte is replaced by a value dependent on all 4 bytes in the column
- effectively a matrix multiplication in GF(28) using prime poly $m(x) = x^8 + x^4 + x^3 + x + 1$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$



- can express each col as 4 equations
 - to derive each new byte in col
- decryption requires use of inverse matrix
 - with larger coefficients, hence a little harder
- have an alternate characterisation
 - each column a 4-term polynomial
 - with coefficients in GF(28)
 - and polynomials multiplied modulo (x⁴+1)

e0	b8	1e
b4	41	27
52	11	98
ae	f1	e5



04	e	b8	1e
66	cb	41	27
81	19	11	98
e5	9a	f1	e5

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

- XOR state with 128-bits of the round key
- again processed by column (though effectively a series of byte operations)
- inverse for decryption identical
 - since XOR own inverse, with reversed keys
- designed to be as simple as possible
 - a form of Vernam cipher on expanded key
 - requires other stages for complexity / security

S _{0,0}	S _{0,1}	S _{0,2}	S _{0,3}
S _{1,0}	S _{1,1}	S _{1,2}	S _{1,3}
S _{2,0}	S _{2,1}	S _{2,2}	S _{2,3}
S _{3,0}	S _{3,1}	S _{3,2}	S _{3,3}

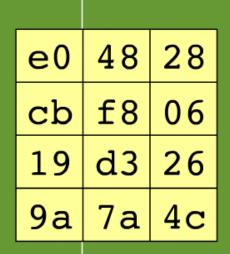


Wi	W _{i+1}	W _{i+2}	W _{i+3}
----	------------------	------------------	------------------

04	e e	48	28
66	င်	f8	06
81	19	d3	26
e5	9a	7a	4c

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

a0	88	23	2a
fa	54	a3	6c
fe	2c	39	76
17	b1	39	05



04		a0		a4
66	\oplus	fa	-	9c
81	Θ	fe		7f
e5		17		f2

88	23	2a
54	a3	6c
2c	39	76
b1	39	05

a4	68	48	28
9c	9f	f8	06
7f	35	d3	26
f2	2b	7a	4c

23	2a
a3	6c
39	76
39	05

a4	68	6b	02
9c	9f	5b	6a
7f	35	ea	50
f2	2b	43	49

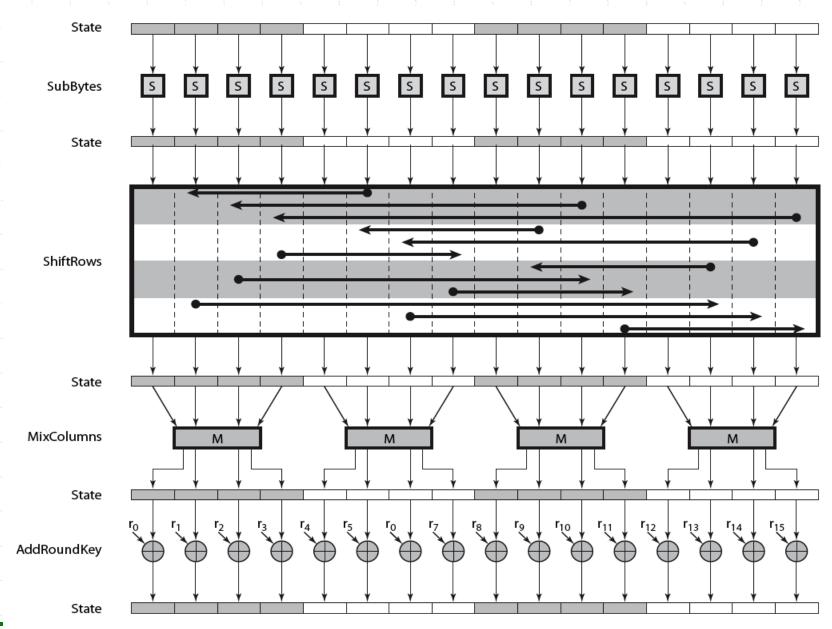
AES Round – ciphertext generation

	Round 2	Round 3	Round 4	Round 5	Round 6
2.51	49 45 7f 77	ac ef 13 45	52 85 e3 f6	e1 e8 35 97	a1 78 10 4c
After SubBytes	de db 39 02 d2 96 87 53	73 c1 b5 23 cf 11 d6 5a	50 a4 11 cf 2f 5e c8 6a	4f fb c8 6c d2 fb 96 ae	63 4f e8 d5 a8 29 3d 03
-	89 f1 1a 3b	7b df b5 b8	28 d7 07 94	9b ba 53 7c	fc df 23 fe
	49 45 7f 77	ac ef 13 45	52 85 e3 f6	e1 e8 35 97	a1 78 10 4c
After	db 39 02 de	c1 b5 23 73	a4 11 cf 50	fb c8 6c 4f	4f e8 d5 63
ShiftRows	87 53 d2 96	d6 5a cf 11	c8 6a 2f 5e	96 ae d2 fb	3d 03 a8 29
	3b 89 f1 1a	b8 7b df b5	94 28 d7 07	7c 9b ba 53	fe fc df 23
	58 1b db 1b	75 20 53 bb	0f 60 6f 5e	25 bd b6 4c	4b 2c 33 37
After	4d 4b e7 6b	ec 0b c0 25	d6 31 c0 b3	d1 11 3a 4c	86 4a 9d d2
MixColumns	ca 5a ca b0	09 63 cf d0	da 38 10 13	a9 d1 33 c0	8d 89 f4 18
	f1 ac a8 e5	93 33 7c dc	a9 bf 6b 01	ad 68 8e b0	6d 80 e8 d8
	\oplus	\oplus	\oplus	\oplus	\oplus
	f2 7a 59 73	3d 47 1e 6d	ef a8 b6 db	d4 7c ca 11	6d 11 db ca
Round Key	c2 96 35 59	80 16 23 7a	44 52 71 0b	d1 83 f2 f9	88 0b f9 00
	95 b9 80 f6 f2 43 7a 7f	47 fe 7e 88 7d 3e 44 3b	a5 5b 25 ad 41 7f 3b 00	c6 9d b8 15 f8 87 bc bc	a3 3e 86 93 7a fd 41 fd
	12 43 /a /1	7d 3e 44 3b	41 /1 35 00	f8 87 bc bc	/α 1α 41 1α
	aa 61 82 68	48 67 4d d6	e0 c8 d9 85	f1 c1 7c 5d	26 3d e8 fd
- - .	8f dd d2 32	6c 1d e3 5f	92 63 b1 b8	00 92 c8 b5	0e 41 64 d2
After AddRoundKey	5f e3 4a 46	4e 9d b1 58	7f 63 35 be	6f 4c 8b d5	2e b7 72 8b
riddiculariey	03 ef d2 9a	ee 0d 38 e7	e8 c0 50 01	55 ef 32 0c	17 7d a9 25

AES Round – ciphertext generation

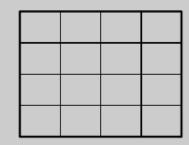
After SubBytes Fractal Property Fractal Propert					
After SubBytes		Round 7	Round 8	Round 9	Round 10
After SubBytes					
## After MixColumns After MixColumns After Mix					
After ShiftRows After ShiftRows After ShiftRows After MixColumns After MixColumn	SubBytes				
ShiftRows					
After MixColumns 14 46 27 34 51 64 62 26 89 6d 99 94 64 3a 42 62 65 84 4e 64 65 84 65 66 67 67 69 47 67 67 68 68 68 68 68 6		40 3d 31 a9	d4 f2 2c 86	46 e7 4a c3	7d 2c 89 07
After MixColumns b5 15 56 d8 2f 89 6d 99 d1 ff cd ea ed a5 a6 bc ed a5 a6 a6 a6 a6 a6 a6 a6					D3 72 31 34
Round Key 4e 5f 84 4e					
Round Key 54 5f a6 a6 f7 c9 4f dc 0e f3 b2 4f 5a 19 a3 7a 41 49 e0 8c AddRoundKey 42 dc 19 04 62 8d 2b 8d 77 fa d1 5c 66 dc 29 00 f3 21 41 6e 77 fa d1 5c 66 dc 29 00 f3 21 41 6e 78 fa d1 5c 66 dc 29 00 f3 21 41 6e 88 89 c8 a6 88 89 c8 a6 89 c8 a6 80 d2 8d 2b 8d 77 fa d1 5c 66 dc 29 00 f3 21 41 6e 88 89 c8 a6		bf ec d7 43	d1 ff cd ea	ed a5 a6 bc —	\oplus
f7 c9 4f dc 0e f3 b2 4f 73 ba f5 29 21 d2 60 2f 66 dc 29 00 f3 21 41 6e f9 25 0c 0c a8 89 c8 a6 Sa 19 a3 7a AddRoundKey ea 04 65 85 83 45 5d 96 5c 33 98 b0 eb 59 8b 1b 40 2e a1 c3 f2 38 13 42 39 02 dc 19 25 dc 11 6a 84 09 85 0b	Round Kev	54 5f a6 a6	d2 8d 2b 8d	77 fa d1 5c	14 ee 3f 63
After 41 49 e0 8c 83 45 5d 96 40 2e a1 c3 42 dc 19 04 5c 33 98 b0 f2 38 13 42 25 dc 11 6a 84 09 85 0b	,				
AddRoundKey 42 dc 19 04 5c 33 98 b0 f2 38 13 42 84 09 85 0b		5a 19 a3 7a		eb 59 8b 1b	39 02 dc 19
b1 1f 65 0c f0 2d ad c5 1e 84 e7 d2 1d fb 97 32		42 dc 19 04	5c 33 98 b0	f2 38 13 42	84 09 85 0b
		b1 1f 65 0c	f0 2d ad c5	1e 84 e7 d2	1d fb 97 32

AES Round summarizing



2b	28	ab	09						
7e	ae	f7	cf						
15	d2	15	4f						
16	a6	88	3c						





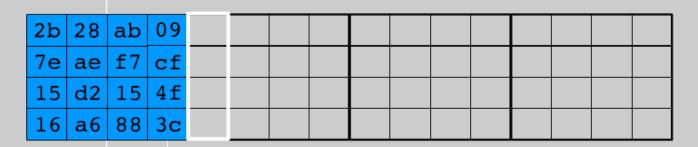
 01
 02
 04
 08
 10
 20
 40
 80
 1b
 36

 00
 00
 00
 00
 00
 00
 00
 00
 00
 00

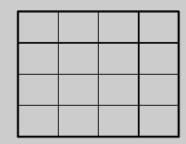
 00
 00
 00
 00
 00
 00
 00
 00
 00
 00

 00
 00
 00
 00
 00
 00
 00
 00
 00
 00









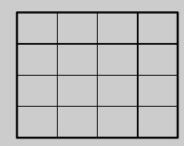
09 cf 4f 3c

RotWord

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

2b	28	ab	09						
7e	ae	f7	cf						
15	d2	15	4f						
16	a6	88	3c						





cf 4f 3c 09

SubBytes

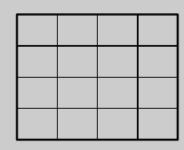
he									3	У							
110		0	1	2	3	4	5	6	7	8	9	a	b	С	d	е	£
	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	с9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	ъ7	fd	93	26	36	3f	£7	CC	34	a5	e5	fl	71	d8	31	15
	3	04	c:7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	ь2	75
	4	09	83	2c	la	1ь	6e	5a	a0	52	3Ъ	d6	Ь3	29	e3	2f	84
	5	53	dl	00	ë	20	fc	ы	5Ь	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7£	50	3с	9f	a8
×	7	51	a3	40	8f	92	9d	38	£5	bc	b6	da	21	10	ff	f3	d2
1 ^	8	cd	0c	13	ec	5£	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	de	22	2a	90	88	46	ee	p8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	ь	e7	c8	37	6d	8d	d5	4e	a9	6c	56	£4	ea	65	7a	ae	08
	С	ba	78	25	2e	1c	a6	b4	с6	e8	dd	74	1f	4b	bd	8b	:8a:
	d	70	3е	b5	66	48	0.3	f6	0e	61	35	57	b9	86	c1	1d	9e
	е	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

S-BOX

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

2b	28	ab	09						
7e	ae	f7	cf						
15	d2	15	4f						
16	a6	88	3c						





8a 84 3c

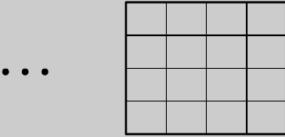
SubBytes

he										y							
110		0	1	2	3	4	5	6	7	8	9	a	b	С	d	е	f
	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	с9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	Ъ7	fd	93	26	36	3f	£7	cc	34	a5	e5	fl	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	ь2	75
	4	09	83	2c	la	1ь	6e	5a	a0	52	3Ь	d6	ь3	29	e3	2f	84
	5	53	dl	00	ë	20	fc	ы	5Ь	6а	сь	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7£	50	3с	9f	a8
×	7	51	a3	40	8f	92	9d	38	£5	bc	b6	da	21	10	ff	f3	d2
1^	8	cd	0c	13	ec	5£	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	p8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	Sc	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6с	56	£4	ea	65	7a	ae	0.8
	C	ba	78	25	2e	1c	a6	b4	с6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	0.3	f6	0e	61	35	57	b9	86	c1	1d	9e
	е	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	Of	b0	54	bb	16

S-BOX

	01	02	04	08	10	20	40	80	1b	36
ı	00	00	00	00	00	00	00	00	00	00
ı	00	00	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00	00	00

2b	28	ab	09						
7e	ae	f7	cf						
15	d2	15	4f						
16	a6	88	3c						



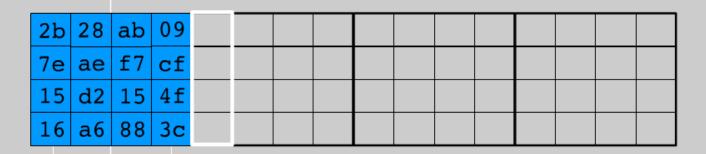
8a 84 eb 01

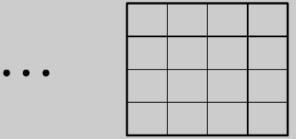
SubBytes

la es									7	y							
he	:X	0	1	2	3	4	5	6	7	8	9	a	b	С	d	е	f
	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	с9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	ь7	fd	93	26	36	3f	£7	cc	34	a5	e5	fl	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	ь2	75
	4	09	83	2c	la	1b	6e	5a	a0	52	3Ъ	d6	b3	29	e3	2f	84
	5	53	dl	00	ed	20	fc	ы	5Ь	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7£	50	3c	9f	a8
×	7	51	a3	40	8f	92	9d	38	£5	bc	b6	da	21	10	ff	f3	d2
11^	8	cd	0c	13	ec	5£	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	p8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	Sc	c2	d3	ac	62	91	95	e4	79
	ь	e7	c8	37	6d	8d	d5	4e	a9	6с	56	f4	ea	65	7a	ae	08
	С	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3е	b5	66	48	0.3	f6	0e	61	35	57	b9	86	c1	1d	9e
	е	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	е6	42	68	41	99	2d	0f	b0	54	bb	16

S-BOX

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00





2b		8a		01		a0
7e	\oplus	84	\oplus	00	_	fa
15	W	eb	U	00		fe
16		01		00		17
			R	con	(4)	

02	04	08	10	20	40	80	1b	36
00	00	00	00	0	00	00	00	00
00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00

2b	28	ab	09	a0						
7e	ae	f7	cf	fa						
15	d2	15	4f	fe						
16	a6	88	3c	17						



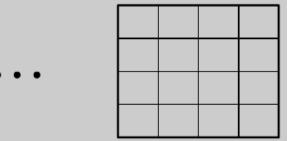
 02
 04
 08
 10
 20
 40
 80
 1b
 36

 00
 00
 00
 00
 00
 00
 00
 00
 00

 00
 00
 00
 00
 00
 00
 00
 00
 00

 00
 00
 00
 00
 00
 00
 00
 00
 00

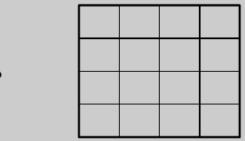
2b	28	ab	09	a0						
7e	ae	f7	cf	fa						
15	d2	15	4f	fe						
16	a6	88	3c	17						



28		a0		88
ae	\oplus	fa	_	54
d2	lack	fe		20
a6		17		b.

02	04	08	10	20	40	80	1b	36
00	00	00	00	0	00	00	00	00
00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00

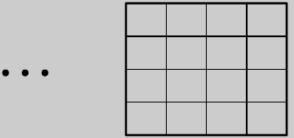
2b	28	ab	09	a0	88					
7e	ae	f7	cf	fa	54					
15	d2	15	4f	fe	2c					
16	a6	88	3c	17	b1					



ab		88		23
f7	\oplus	54	_	a3
15	lacktriangle	2c		39
88		b1		39

02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00

2b	28	ab	09	a0	88	23					
7e	ae	f7	cf	fa	54	a3					
15	d2	15	4f	fe	2c	39					
16	a6	88	3c	17	b1	39					

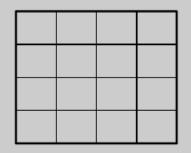


09		23		2a
cf	\oplus	a3	_	6c
4f	Ψ	39		76
3c		39		05

		08						
00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00

2b	28	ab	09	a0	88	23	2a				
7e	ae	f7	cf	fa	54	a3	6c				
15	d2	15	4f	fe	2c	39	76				
16	a6	88	3c	17	b1	39	05				





6c 76 05 2a

SubBytes

la es									7	у							\neg
he	×	0	1	2	3	4	5	6	7	8	9	a	b	С	d	е	£
Г	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
ш	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
ш	2	ь7	fd	93	26	36	3f	£7	cc	34	a5	e5	fl	71	d8	31	15
ш	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
ш	4	09	83	2c	la	1ь	6e	5a	a0	52	3Ь	d6	ь3	29	e3	2f	84
ш	5	53	dl	00	ed	20	fc	Ыl	5Ъ	6a	cb	be	39	4a	4c	58	cf
ш	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
l _×	7	51	a3	40	8f	92	9d	38	£5	bc	b6	da	21	10	ff	f3	d2
l^	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
ш	9	60	81	4f	dc	22	2a	90	88	46	ee	p8	14	de	5e	0b	ďb
ш	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
ш	ь	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
ш	c	ba	78	25	2e	1c	а6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3е	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ш	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
L	f	8c	a1	89	0d	bf	еб	42	68	41	99	2d	0f	b0	54	bb	16

S-BOX

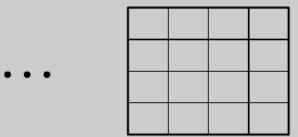
 02
 04
 08
 10
 20
 40
 80
 1b
 36

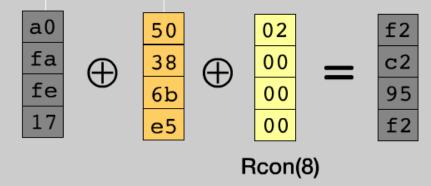
 00
 00
 00
 00
 00
 00
 00
 00

 00
 00
 00
 00
 00
 00
 00
 00

 00
 00
 00
 00
 00
 00
 00
 00

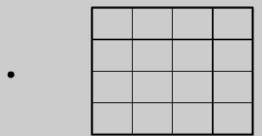
2b	28	ab	09	a0	88	23	2a				
7e	ae	f7	cf	fa	54	a3	6c				
15	d2	15	4f	fe	2c	39	76				
16	a6	88	3c	17	b1	39	05				





04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

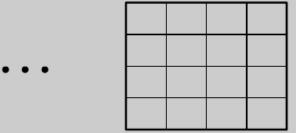
2b	28	ab	09	a0	88	23	2a	f2				
7e	ae	f7	cf	fa	54	a3	6c	с2				
15	d2	15	4f	fe	2c	39	76	95				
16	a6	88	3c	17	b1	39	05	f2				



88		f2		7a
54	\oplus	с2	_	96
2c	Θ	95		b9
b1		f2		43

04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

2b	28	ab	09	a0	88	23	2a	f2	7a			
7e	ae	f7	cf	fa	54	a3	6c	c2	96			
15	d2	15	4f	fe	2c	39	76	95	b9			
16	a6	88	3c	17	b1	39	05	f2	43			



23	
a3	6
39	
39	

04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

Round key 2

Round key 3

2b	28	ab	09	a0	88	23	2a	f2	7a	23	73	3d	47	1e	6d
7e	ae	f7	cf	fa	54	a3	6c	c2	96	a3	59	80	16	23	7a
15	d2	15	4f	fe	2c	39	76	95	b9	39	f6	47	fe	7e	88
16	a6	88	3c	17	b1	39	05	f2	43	39	7f	7d	3e	44	3b

Round key 1

Cipher Key

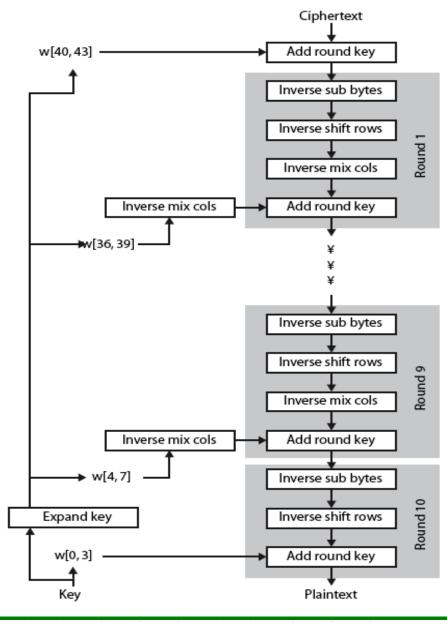
. . .

d0	с9	e1	b6
14	ee	3f	63
f9	25	0c	0c
a8	89	c8	a6

AES Decryption

- AES decryption is not identical to encryption since steps done in reverse
- but can define an equivalent inverse cipher with steps as for encryption
 - but using inverses of each step
 - with a different key schedule
- works since result is unchanged when
 - swap byte substitution & shift rows
 - swap mix columns & add (tweaked) round key

AES Decryption



Inverse S-Box

			y														
		0	1	2	3	4	5	6	7	8	9	Α	В	С	D	Е	F
	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9 B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A 6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A 1	66	28	D9	24	B 2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	В6	92
	5	6C	70	48	50	FD	ED	B 9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	$_{\mathrm{BC}}$	D3	0A	F7	E4	58	05	B 8	В3	45	06
	7	$\mathbf{D}0$	2C	1E	8F	CA	3F	0 F	02	C1	AF	$^{\mathrm{BD}}$	03	01	13	8A	6B
X	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B 4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	Α	47	F1	1A	71	1D	29	C5	89	6F	B 7	62	0E	AA	18	BE	1B
	В	FC	56	3E	4B	C6	D2	79	20	9A	DB	CO	FE	78	CD	5A	F4
	С	1F	DD	A8	33	88	07	C7	31	B 1	12	10	59	27	80	EC	5F
	D	60	51	7F	A 9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A 0	E0	3B	4D	AE	2A	F5	B 0	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

(b) Inverse S-box

Inverse Shift Row Transformation

- Performs the circular shifts in opposite direction
 - Circular right shift
- 1-byte circular right shift for second
- > 2-byte circular right shift for third and so on....

Inverse Mix-columns operation

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Implementation Aspects

- can efficiently implement on 8-bit CPU
 - byte substitution works on bytes using a table of 256 entries
 - shift rows is simple byte shift
 - add round key works on byte XOR's
 - mix columns requires matrix multiply in GF(28) which works on byte values, can be simplified to use table lookups & byte XOR's

Implementation Aspects

- can efficiently implement on 32-bit CPU
 - redefine steps to use 32-bit words
 - can precompute 4 tables of 256-words
 - then each column in each round can be computed using 4 table lookups + 4 XORs
 - at a cost of 4Kb to store tables
- designers believe this very efficient implementation was a key factor in its selection as the AES cipher

AES Example

Plaintext:	0123456789abcdeffedcba9876543210
Key:	0f1571c947d9e8590cb7add6af7f6798
Ciphertext:	ff0b844a0853bf7c6934ab4364148fb9

AES Key Expansion

Table 5.3 Key Expansion for AES Example

Key Words	Auxiliary Function
w0 = 0f 15 71 c9	RotWord(w3) = 7f 67 98 af = x1
w1 = 47 d9 e8 59	SubWord $(x1) = d2 85 46 79 = y1$
w2 = 0c b7 ad d6	Rcon(1) = 01 00 00 00
w3 = af 7f 67 98	y1 ⊕ Rcon(1) = d3 85 46 79 = z1
$w4 = w0 \oplus z1 = dc 90 37 b0$	RotWord(w7) = 81 15 a7 38 = x2
$w5 = w4 \oplus w1 = 9b \ 49 \ df \ e9$	SubWord $(x2) = 0c 59 5c 07 = y2$
$w6 = w5 \oplus w2 = 97 \text{ fe } 72 \text{ 3f}$	Rcon(2) = 02 00 00 00
w7 = w6	y2 ⊕ Rcon(2) = 0e 59 5c 07 = z2
$w8 = w4 \oplus z2 = d2 c9 6b b7$	RotWord(w11) = ff d3 c6 e6 = $x3$
w9 = w8 ⊕ w5 = 49 80 b4 5e	SubWord(x3) = 16 66 b4 83 = y3
w10 = w9 ⊕ w6 = de 7e c6 61	Rcon(3) = 04 00 00 00
$w11 = w10 \oplus w7 = e6 \text{ ff d3 c6}$	y3 (+) Rcon(3) = 12 66 b4 8e = z3
w12 = w8 + z3 = c0 af df 39	RotWord(w15) = ae 7e c0 b1 = $x4$
$w13 = w12 \oplus w9 = 89 2f 6b 67$	SubWord(x4) = e4 f3 ba c8 = y4
w14 = w13 ⊕ w10 = 57 51 ad 06	Rcon(4) = 08 00 00 00
$w15 = w14 \oplus w11 = b1$ ae 7e c0	y4 ⊕ Rcon(4) = ec f3 ba c8 = 4

AES Key Expansion...

Key Words	Auxiliary Function
$w16 = w12 \oplus z4 = 2c$ 5c 65 f1	RotWord(w19) = 8c dd 50 43 = x5
$w17 = w16 \oplus w13 = a5 73 0e 96$	SubWord(x5) = 64 c1 53 1a = y5
$w18 = w17 \oplus w14 = f2 22 a3 90$	Rcon(5) = 10 00 00 00
$w19 = w18 \oplus w15 = 43$ 8c dd 50	$y5 \oplus Rcon(5) = 74 c1 53 1a = z5$
$w20 = w16 \oplus z5 = 58 \text{ 9d } 36 \text{ eb}$	RotWord (w23) = 40 46 bd 4c = x6
$w21 = w20 \oplus w17 = fd$ ee 38 7d	SubWord (x6) = 09 5a 7a 29 = y6
$w22 = w21 \oplus w18 = 0f \text{ cc } 9b \text{ ed}$	Rcon(6) = 20 00 00 00
$w23 = w22 \oplus w19 = 4c \ 40 \ 46 \ bd$	y6 (Rcon(6) = 29 5a 7a 29 = z6
w24 = w20 ⊕ z6 = 71 c7 4c c2	RotWord (w27) = a5 a9 ef cf = x7
$w25 = w24 \oplus w21 = 8c 29 74 bf$	SubWord $(x7) = 06 \text{ d3 bf } 8a = y7$
$w26 = w25 \oplus w22 = 83 \text{ e5 ef } 52$	Rcon (7) = 40 00 00 00
$w27 = w26 \oplus w23 = cf \ a5 \ a9 \ ef$	$y7 \oplus Rcon(7) = 46 d3 df 8a = z7$
$w28 = w24 \oplus z7 = 37 14 93 48$	RotWord (w31) = 7d a1 4a f7 = x8
$w29 = w28 \oplus w25 = bb \ 3d \ e7 \ f7$	SubWord (x8) = ff 32 d6 68 = y8
$w30 = w29 \oplus w26 = 38 d8 08 a5$	Rcon (8) = 80 00 00 00
$w31 = w30 \oplus w27 = f7 7d a1 4a$	y8 \oplus Rcon(8) = 7f 32 d6 68 = z8
w32 = w28 + z8 = 48 26 45 20	RotWord (w35) = be 0b 38 3c = x9
$w33 = w32 \oplus w29 = f3$ 1b a2 d7	SubWord (x9) = ae 2b 07 eb = y9
$w34 = w33 \oplus w30 = cb \ c3 \ aa \ 72$	Rcon (9) = 1B 00 00 00
$w35 = w34 \oplus w32 = 3c$ be 0b 3	y9 ⊕ Rcon (9) = b5 2b 07 eb = z9
w36 = w32 ⊕ z9 = fd 0d 42 cb	RotWord (w39) = 6b 41 56 f9 = x10
$w37 = w36 \oplus w33 = 0e 16 e0 1c$	SubWord (x10) = 7f 83 b1 99 = y10
w38 = w37 ⊕ w34 = c5 d5 4a 6e	Rcon (10) = 36 00 00 00
$w39 = w38 \oplus w35 = f9 \ 6b \ 41 \ 56$	y10 ⊕ Rcon (10) = 49 83 b1 99 = z10
w40 = w36 \oplus z10 = b4 8e f3 52	
w41 = w40 ⊕ w37 = ba 98 13 4e	
$w42 = w41 \oplus w38 = 7f \ 4d \ 59 \ 20$	
$w43 = w42 \oplus w39 = 86 \ 26 \ 18 \ 76$	

AES Example

Example

AES – Avalanche Effect

Table 5.5 Avalanche Effect in AES: Change in Plaintext

Round		Number of Bits that Differ
	0123456789abcdeffedcba9876543210 0023456789abcdeffedcba9876543210	1
0	0e3634aece7225b6f26b174ed92b5588 0f3634aece7225b6f26b174ed92b5588	1
1	657470750fc7ff3fc0e8e8ca4dd02a9c c4a9ad090fc7ff3fc0e8e8ca4dd02a9c	20
2	5c7bb49a6b72349b05a2317ff46d1294 fe2ae569f7ee8bb8c1f5a2bb37ef53d5	58
3	7115262448do747e5cdac7227da9bd9c ec093dfb7c45343d689017507d485e62	59
4	f867aee8b437a5210c24c1974cffeabc 43efdb697244df808e8d9364ee0ae6f5	61
5	721eb200ba06206dcbd4bce704fa654e 7b28a5d5ed643287e006c099bb375302	68
6	0ad9d85689f9f77bclc5f71185e5fb14 3bc2d8b6798d8ac4fe36ald891ac181a	64
7	db18a8ffa16d30d5f88b08d777ba4eaa 9fb8b5452023c70280e5c4bb9e5555a4b	67
8	f91b4fbfe934c9bf8f2f85812b084989 20264e1126b219aef7feb3f9b2d6de40	65
9	ccal04al3e678500ff59025f3bafaa34 b56a034lb2290ba7dfdfbddcd8578205	61
10	ff0b844a0853bf7c6934ab4364148fb9 612b89398d0600cda116227ca72433f0	58

Hypothecation.....

- Prof.(Dr.) D. C. Jinwala, SVNIT, Surat
- •Ueli Maurer, ETH Zurich
- •C.R. Subramanian, ISEA Workshop, IMSc, Chennai
- Boaz Barak, Princeton University
- José de Jesús Angel Angel
- AES Demos

To teach is to learn twice!!

