Input

State

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

This is a block from the plaintext message to be encrypted.

Cipher key

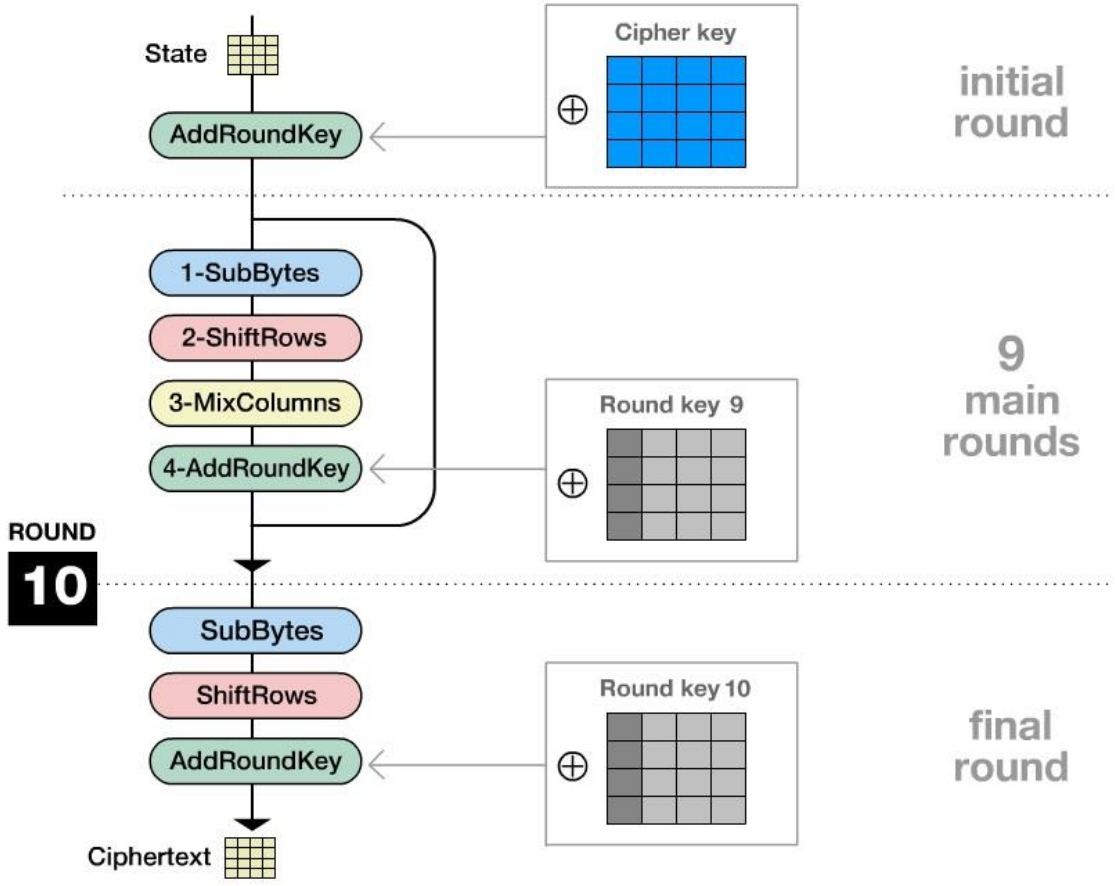
2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

Hexadecimal notation (sample):

32 = 00110010 (1 byte)

3hex 2hex

Encryption Process



1 - SubBytes

Round 1

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

hex	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

S-BOX byte substitution table

1 - SubBytes

Round 1

19

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

hex	y															
	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				2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0				af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc				f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a				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

x

d4

S-BOX byte substitution table

1 - SubBytes

Round 1

d4	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

hex	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

S-BOX byte substitution table

1 - SubBytes

Round 1

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

hex	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

S-BOX byte substitution table

2 - ShiftRows

Round 1

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

2 - ShiftRows

Round 1

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

rotate over 2 bytes

2 - ShiftRows

Round 1

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

rotate over 3 bytes

2 - ShiftRows

Round 1

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

3 - MixColumns

Round 1

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

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \bullet \begin{bmatrix} d4 \\ bf \\ 5d \\ 30 \end{bmatrix} = \begin{bmatrix} 04 \\ 66 \\ 81 \\ e5 \end{bmatrix}$$

The four numbers of one column are modulo multiplied in Rijndael's Galois Field by a given matrix.

3 - MixColumns

Round 1

04	e0	b8	1e
66	b4	41	27
81	52	11	98
e5	ae	f1	e5

The MixColumns step along with the ShiftRows step is the primary source of diffusion in Rijndael.

3 - MixColumns

Round 1

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

The MixColumns step along with the ShiftRows step is the primary source of diffusion in Rijndael.

4 - AddRoundKey

Round 1

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

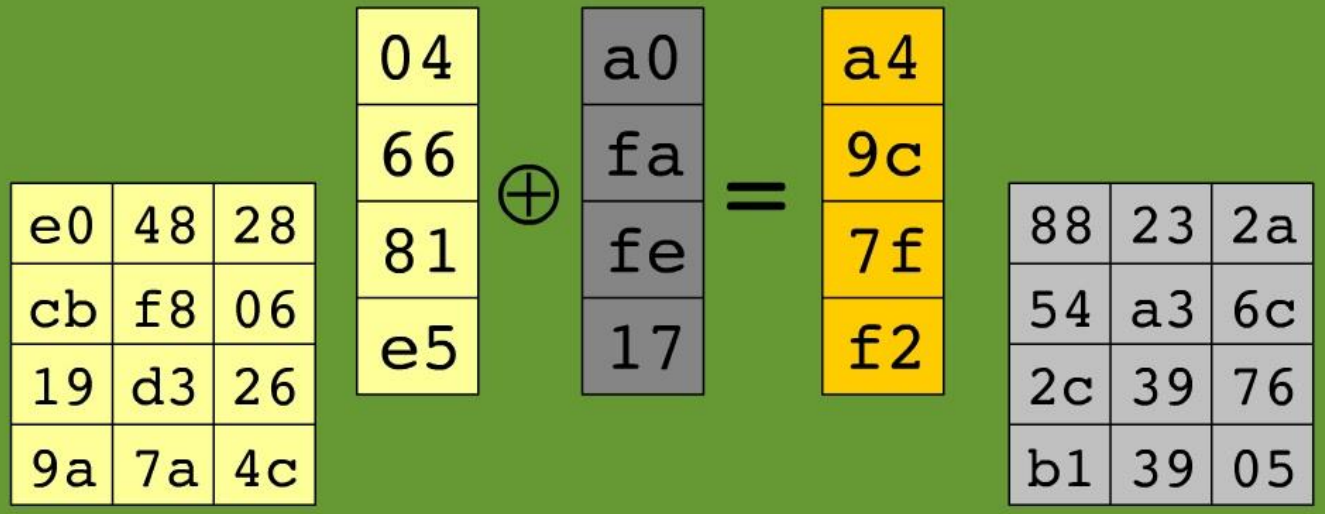
a0	88	23	2a
fa	54	a3	6c
fe	2c	39	76
17	b1	39	05

Round key

(produced as Round key 1 during the Key Schedule - see slide 19)

4 - AddRoundKey

Round 1



Round key

(produced as Round key 1 during the Key Schedule - see slide 19)

4 - AddRoundKey

Round 1

a4	e0	48	28
9c	cb	f8	06
7f	19	d3	26
f2	9a	7a	4c

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

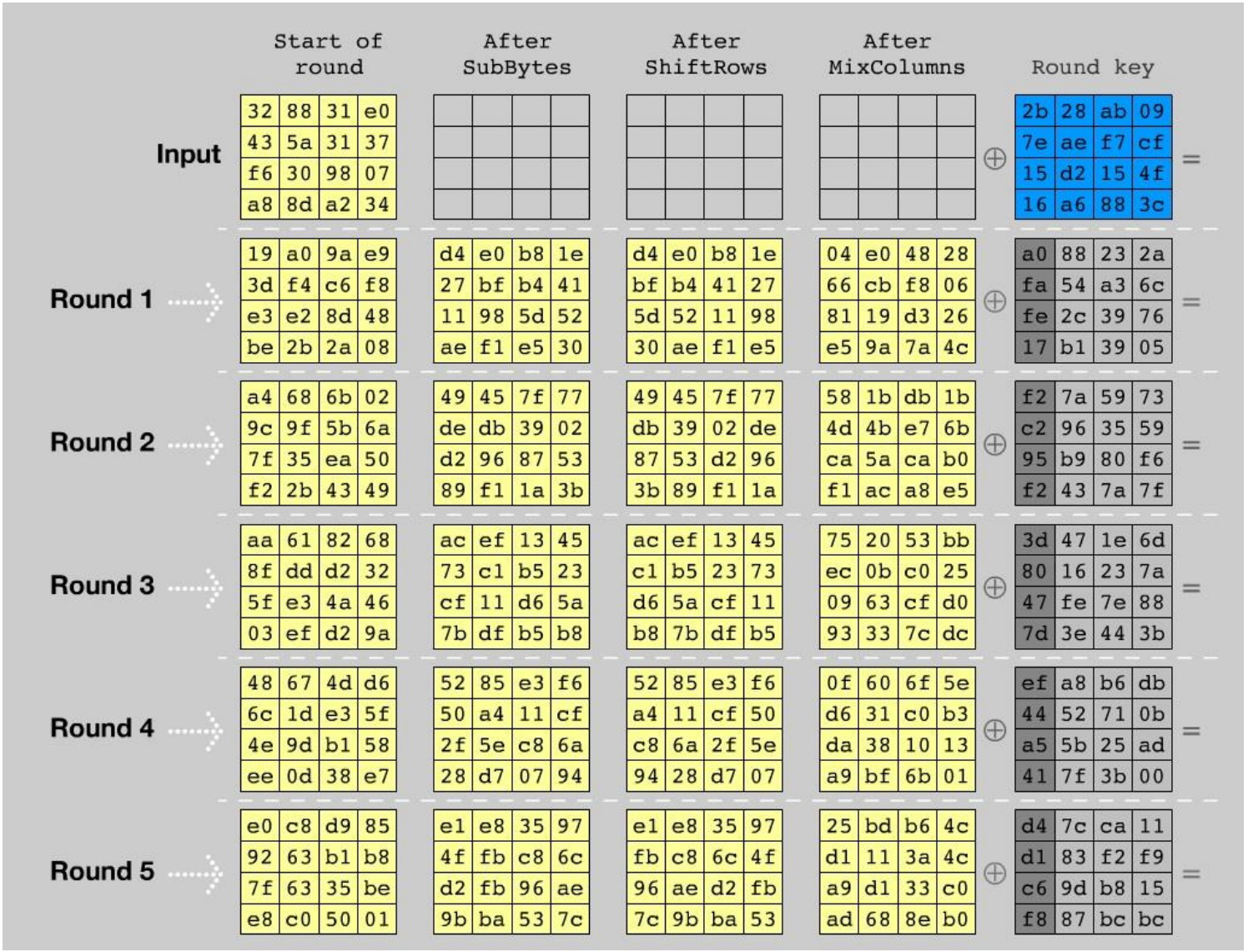
Round key

(produced as Round key 1 during the Key Schedule - see slide 19)

4 - AddRoundKey

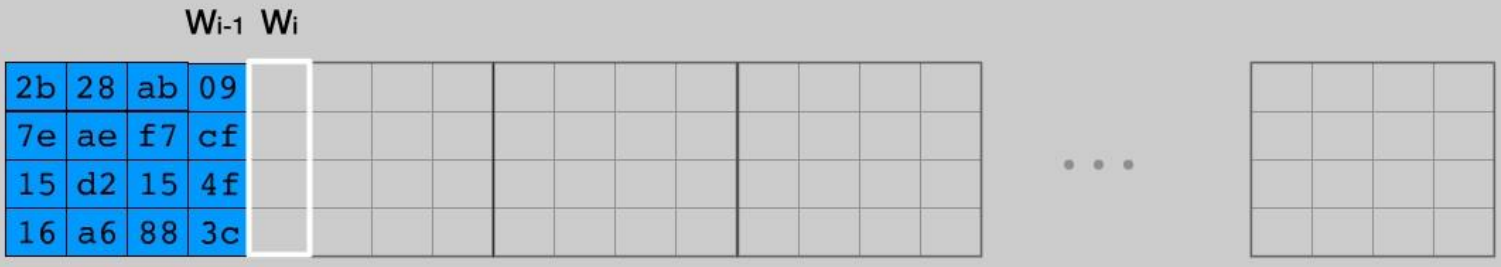
Round 1

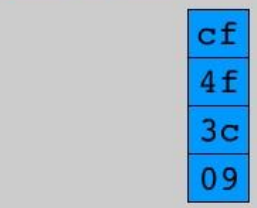
a4	68	6b	02
9c	9f	5b	6a
7f	35	ea	50
f2	2b	43	49


Start of round
After SubBytes
After ShiftRows
After MixColumns
Round key
Input
32 88 31 e0
43 5a 31 37
f6 30 98 07
a8 8d a2 34
2b 28 ab 09
7e ae f7 cf
15 d2 15 4f
16 a6 88 3c
Round 1
19 a0 9a e9
3d f4 c6 f8
e3 e2 8d 48
be 2b 2a 08
d4 e0 b8 1e
27 bf b4 41
11 98 5d 52
ae f1 e5 30
d4 e0 b8 1e
bf b4 41 27
5d 52 11 98
30 ae f1 e5
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
Round 2
a4 68 6b 02
9c 9f 5b 6a
7f 35 ea 50
f2 2b 43 49
49 45 7f 77
de db 39 02
d2 96 87 53
89 f1 1a 3b
49 45 7f 77
db 39 02 de
87 53 d2 96
3b 89 f1 1a
58 1b db 1b
4d 4b e7 6b
ca 5a ca b0
f1 ac a8 e5
f2 7a 59 73
c2 96 35 59
95 b9 80 f6
f2 43 7a 7f
Round 3
aa 61 82 68
8f dd d2 32
5f e3 4a 46
03 ef d2 9a
ac ef 13 45
73 c1 b5 23
cf 11 d6 5a
7b df b5 b8
ac ef 13 45
c1 b5 23 73
d6 5a cf 11
b8 7b df b5
75 20 53 bb
ec 0b c0 25
09 63 cf d0
93 33 7c dc
3d 47 1e 6d
80 16 23 7a
47 fe 7e 88
7d 3e 44 3b
Round 4
48 67 4d d6
6c 1d e3 5f
4e 9d b1 58
ee 0d 38 e7
52 85 e3 f6
50 a4 11 cf
2f 5e c8 6a
28 d7 07 94
52 85 e3 f6
a4 11 cf 50
c8 6a 2f 5e
94 28 d7 07
0f 60 6f 5e
d6 31 c0 b3
da 38 10 13
a9 bf 6b 01
ef a8 b6 db
44 52 71 0b
a5 5b 25 ad
41 7f 3b 00
Round 5
e0 c8 d9 85
92 63 b1 b8
7f 63 35 be
e8 c0 50 01
e1 e8 35 97
4f fb c8 6c
d2 fb 96 ae
9b ba 53 7c
e1 e8 35 97
fb c8 6c 4f
96 ae d2 fb
7c 9b ba 53
25 bd b6 4c
d1 11 3a 4c
a9 d1 33 c0
ad 68 8e b0
d4 7c ca 11
d1 83 f2 f9
c6 9d b8 15
f8 87 bc bc


Start of round
After SubBytes
After ShiftRows
After MixColumns
Round key
Round 6
f1 c1 7c 5d
00 92 c8 b5
6f 4c 8b d5
55 ef 32 0c
a1 78 10 4c
63 4f e8 d5
a8 29 3d 03
fc df 23 fe
a1 78 10 4c
4f e8 d5 63
3d 03 a8 29
fe fc df 23
4b 2c 33 37
86 4a 9d d2
8d 89 f4 18
6d 80 e8 d8
6d 11 db ca
88 0b f9 00
a3 3e 86 93
7a fd 41 fd
Round 7
26 3d e8 fd
0e 41 64 d2
2e b7 72 8b
17 7d a9 25
f7 27 9b 54
ab 83 43 b5
31 a9 40 3d
f0 ff d3 3f
f7 27 9b 54
83 43 b5 ab
40 3d 31 a9
3f f0 ff d3
14 46 27 34
15 16 46 2a
b5 15 56 d8
bf ec d7 43
4e 5f 84 4e
54 5f a6 a6
f7 c9 4f dc
0e f3 b2 4f
Round 8
5a 19 a3 7a
41 49 e0 8c
42 dc 19 04
b1 1f 65 0c
be d4 0a da
83 3b e1 64
2c 86 d4 f2
c8 c0 4d fe
be d4 0a da
3b e1 64 83
d4 f2 2c 86
fe c8 c0 4d
00 b1 54 fa
51 c8 76 1b
2f 89 6d 99
d1 ff cd ea
ea b5 31 7f
d2 8d 2b 8d
73 ba f5 29
21 d2 60 2f
Round 9
ea 04 65 85
83 45 5d 96
5c 33 98 b0
f0 2d ad c5
87 f2 4d 97
ec 6e 4c 90
4a c3 46 e7
8c d8 95 a6
87 f2 4d 97
6e 4c 90 ec
46 e7 4a c3
a6 8c d8 95
47 40 a3 4c
37 d4 70 9f
94 e4 3a 42
ed a5 a6 bc
ac 19 28 57
77 fa d1 5c
66 dc 29 00
f3 21 41 6e
Round 10
eb 59 8b 1b
40 2e a1 c3
f2 38 13 42
1e 84 e7 d2
e9 cb 3d af
09 31 32 2e
89 07 7d 2c
72 5f 94 b5
e9 cb 3d af
31 32 2e 09
7d 2c 89 07
b5 72 5f 94
d0 c9 e1 b6
14 ee 3f 63
f9 25 0c 0c
a8 89 c8 a6
Output
39 02 dc 19
25 dc 11 6a
84 09 85 0b
1d fb 97 32
Ciphertext

Key Schedule





RotWord

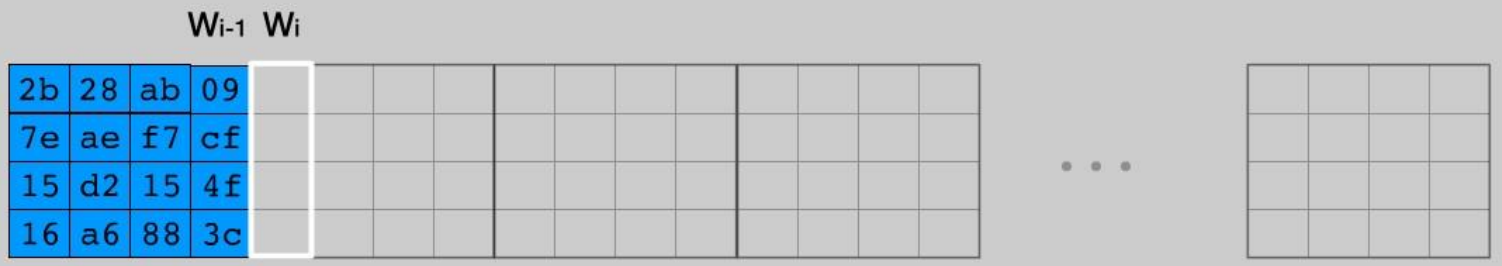
Words in positions that are a multiple of 4 (w_4 , w_8 ,..., w_{40}) are calculated by:

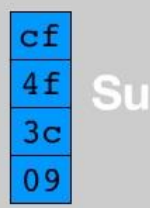
a) applying the **RotWord** and **SubBytes** transformation to the previous word w_{i-1} .

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

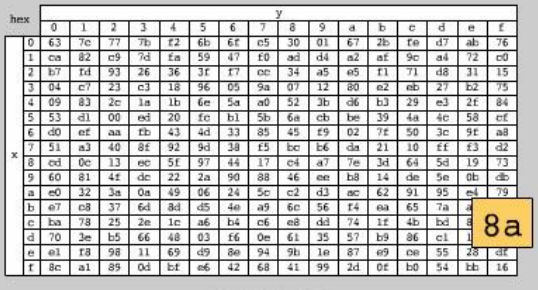
Rcon

Key Schedule



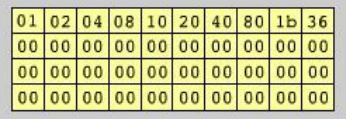


SubBytes



S-BOX byte substitution table

	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

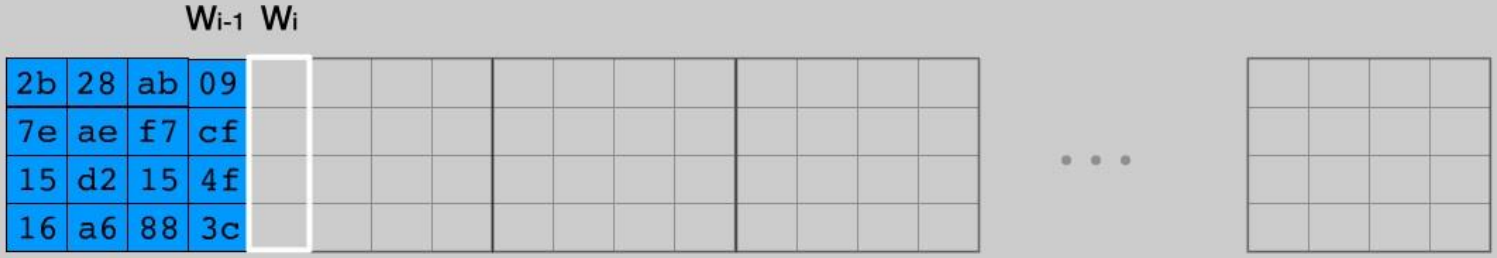


Rcon

Words in positions that are a multiple of 4 (w_4 , w_8 ,..., w_{40}) are calculated by:

a) applying the **RotWord** and **SubBytes** transformation to the previous word w_{i-1} .

Key Schedule



SubBytes

hex	y															
x	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

S-BOX byte substitution table

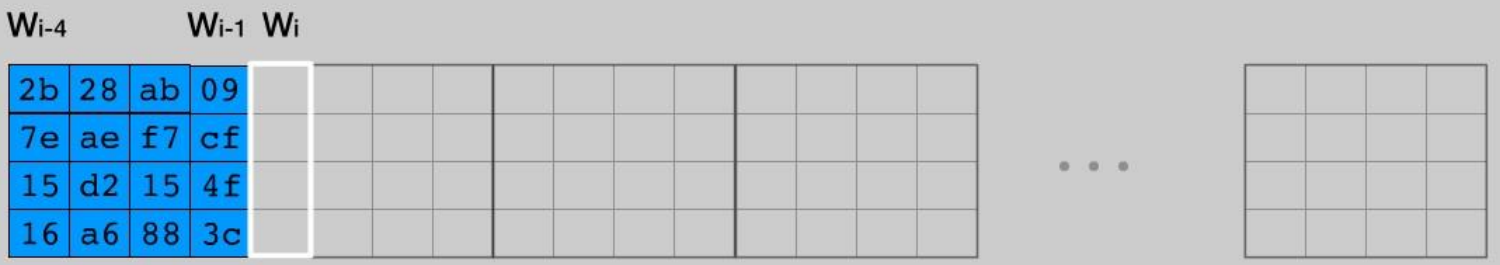
01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

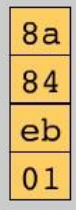
Rcon

Words in positions that are a multiple of 4 (w_4 , w_8 ,..., w_{40}) are calculated by:

a) applying the **RotWord** and **SubBytes** transformation to the previous word w_{i-1} .

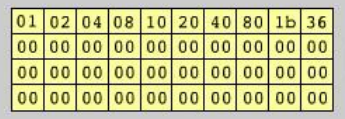
Key Schedule





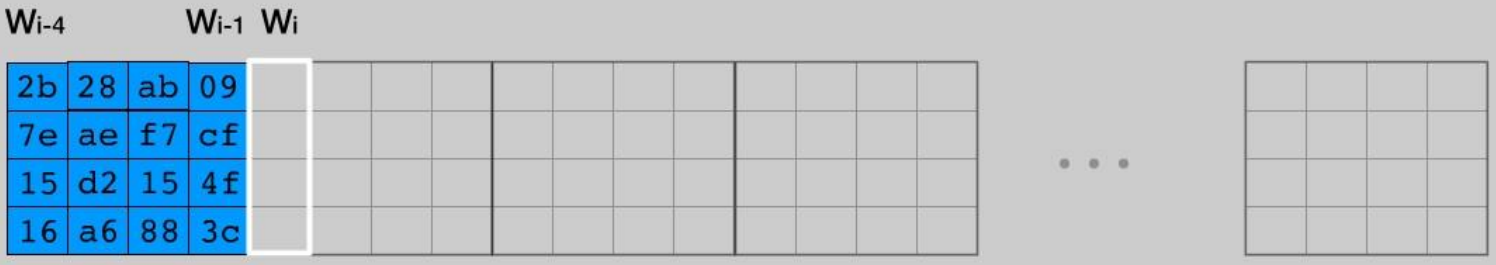
Words in positions that are a multiple of 4 (w_4 , w_8 ,..., w_{40}) are calculated by:

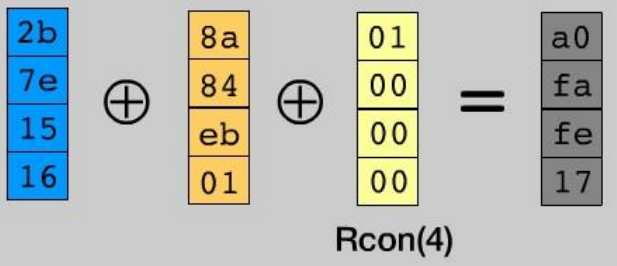
a) applying the **RotWord** and **SubBytes** transformation to the previous word w_{i-1} .



Rcon

Key Schedule





Words in positions that are a multiple of 4 (w_4 , w_8 ,..., w_{40}) are calculated by:

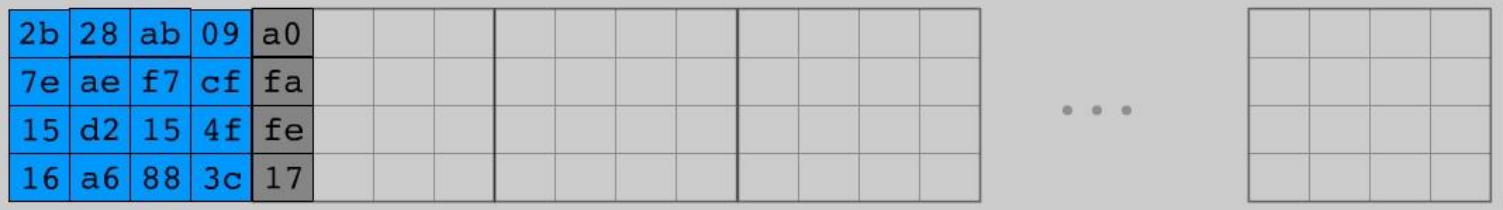
a) applying the **RotWord** and **SubBytes** transformation to the previous word w_{i-1} .

b) Adding (XOR) this result to the word 4 positions earlier w_{i-4} , plus a round constant **Rcon**.

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

Rcon

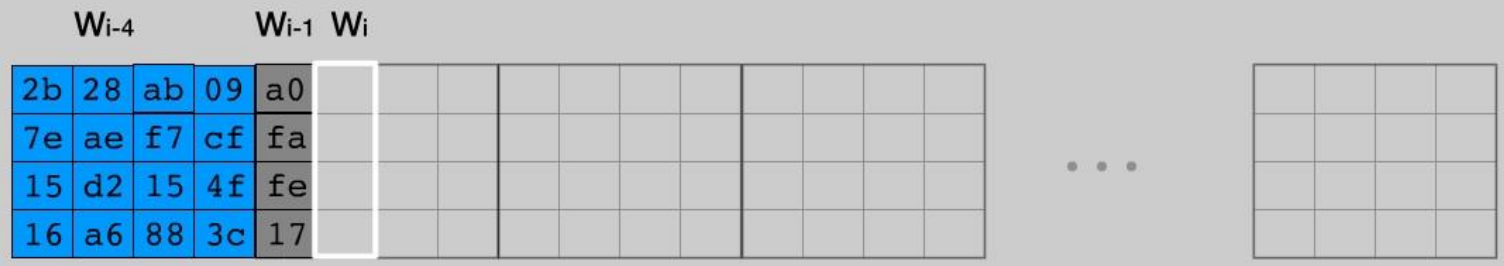
Key Schedule

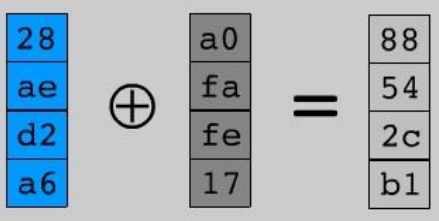


02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00

Rcon

Key Schedule



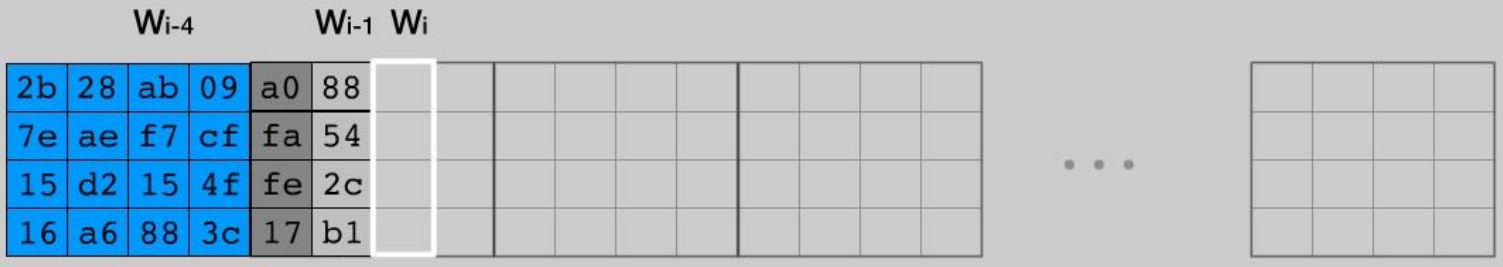


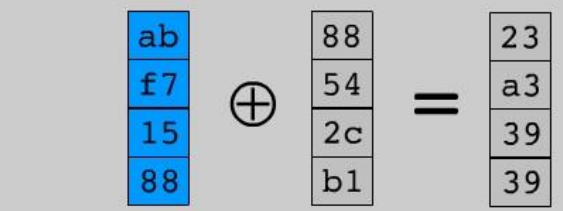
The remaining 32-bit words $\mathbf{w_i}$, are calculated by adding (XOR) the previous word $\mathbf{w_{i-1}}$, with the word 4 positions earlier $\mathbf{w_{i-4}}$.

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

Rcon

Key Schedule



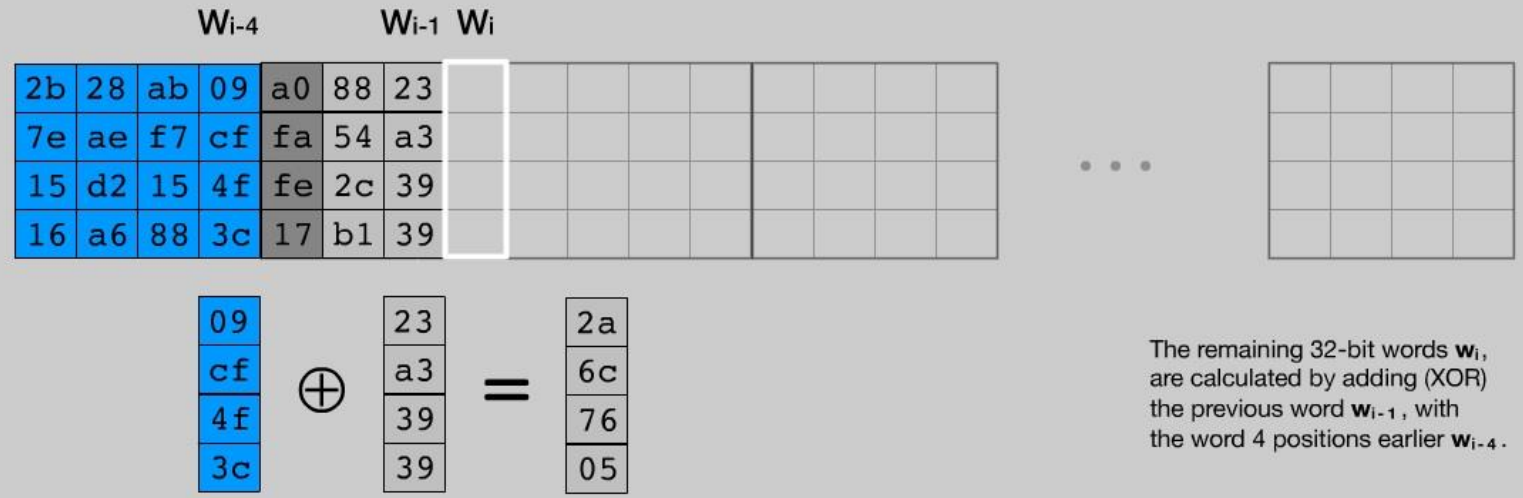


The remaining 32-bit words $\mathbf{w}_i$, are calculated by adding (XOR) the previous word $\mathbf{w}_{i-1}$, with the word 4 positions earlier $\mathbf{w}_{i-4}$.

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

Rcon

Key Schedule

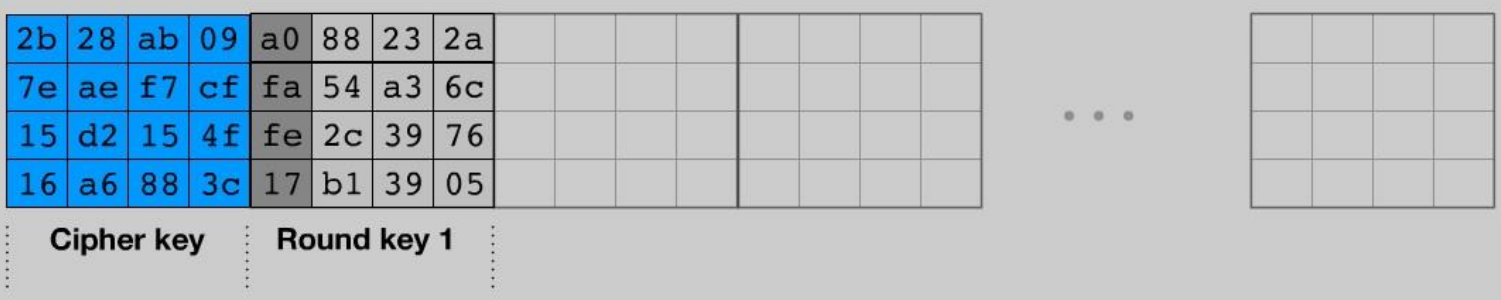


The remaining 32-bit words $\mathbf{w}_i$, are calculated by adding (XOR) the previous word $\mathbf{w}_{i-1}$, with the word 4 positions earlier $\mathbf{w}_{i-4}$.

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

Rcon

Key Schedule



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

Rcon

Key Schedule