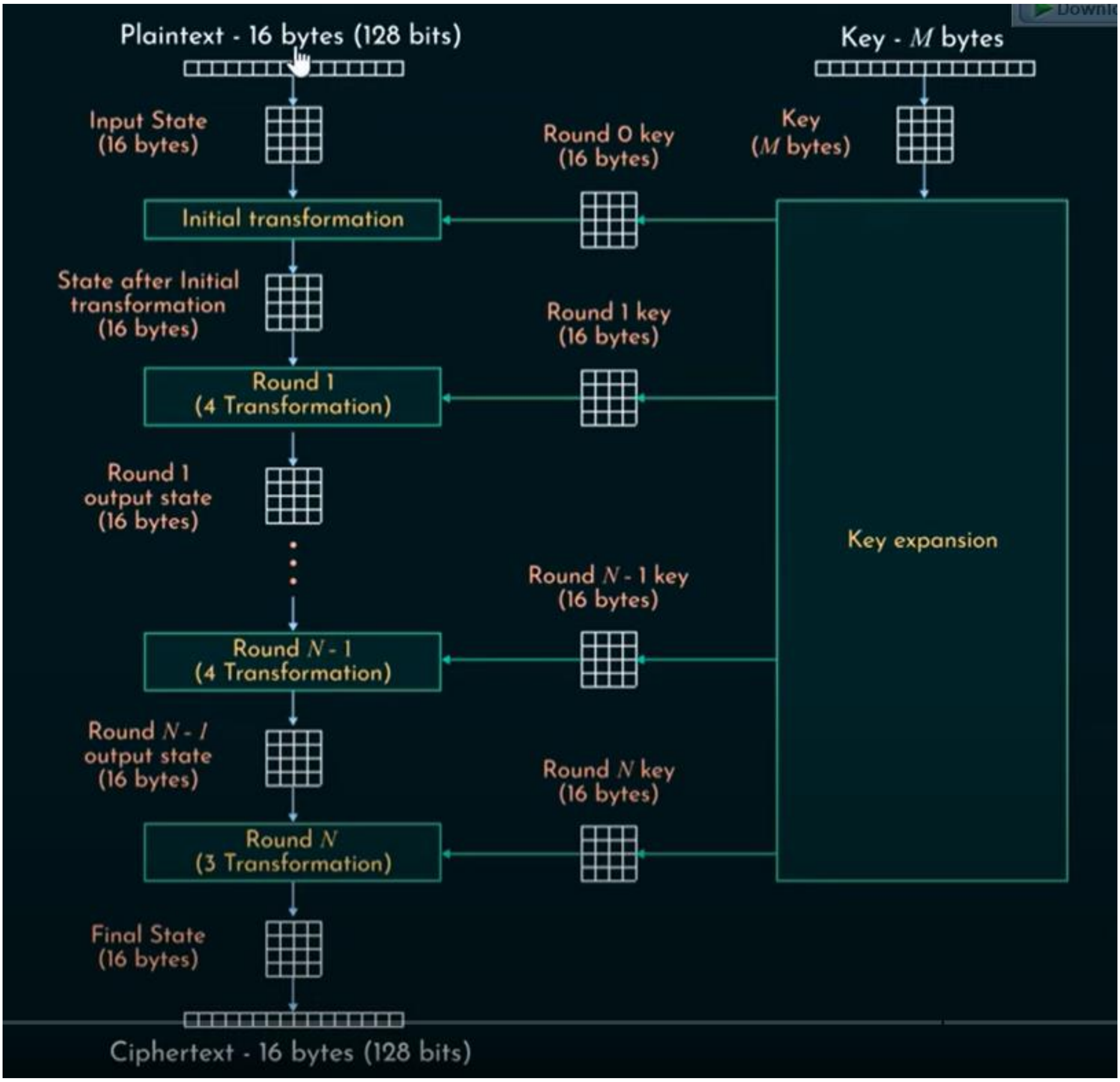
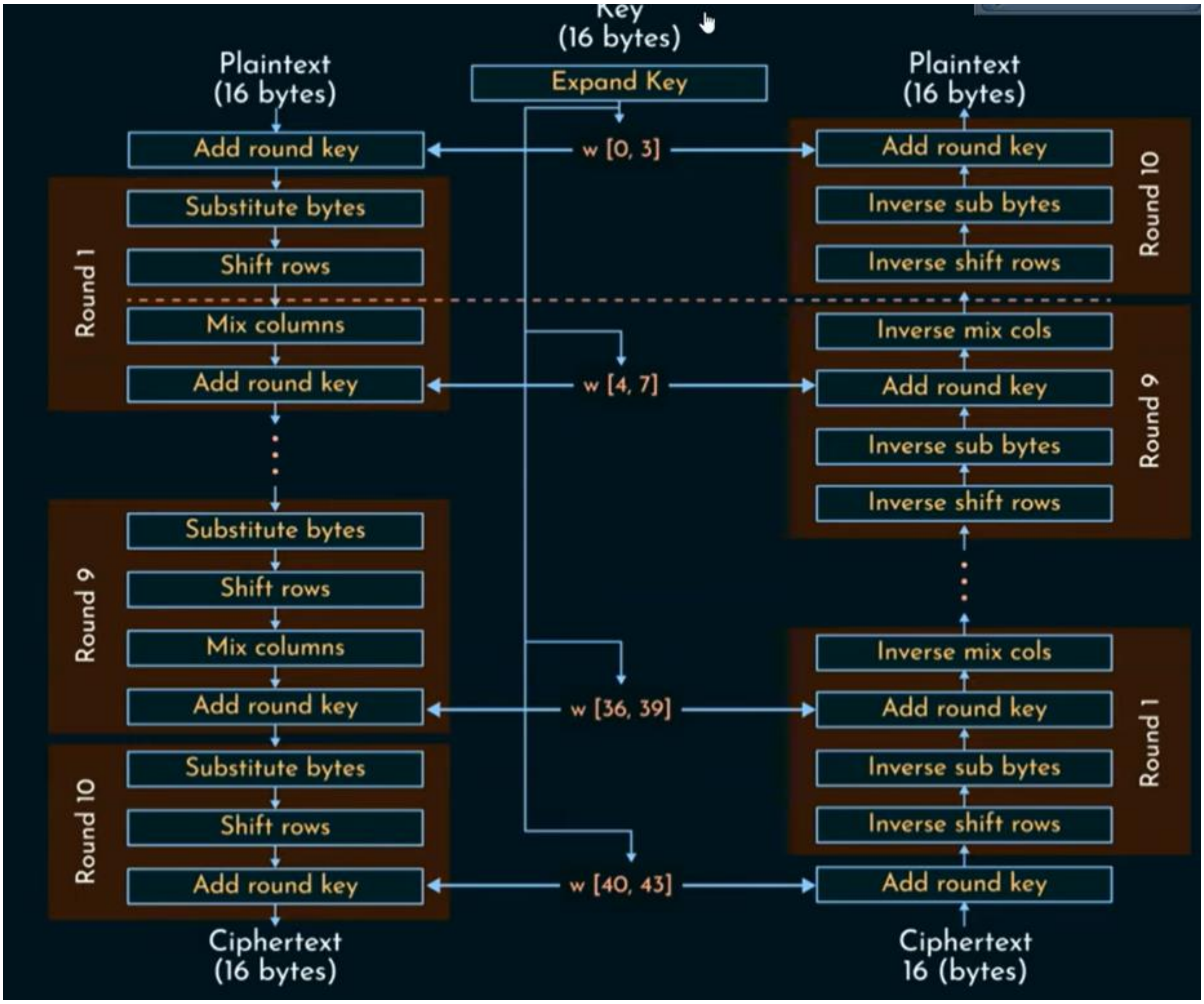


The 128 bits of the plaintext are arranged such that they form 4x4 matrix. The 4 transformations of the rounds are [substitute bytes – shift rows – mix columns – add round key]. Note that all rounds have same transformation steps except the initial transformation (a.k.a round 0) and the last round (Nth round). They number of rounds depends on the key size (illustrated in the following figure). However, the **<u>round keys</u>** have constant sizes of 128 bits.

No. of rounds	Key size (in bits)
10	128
12	192
14	256

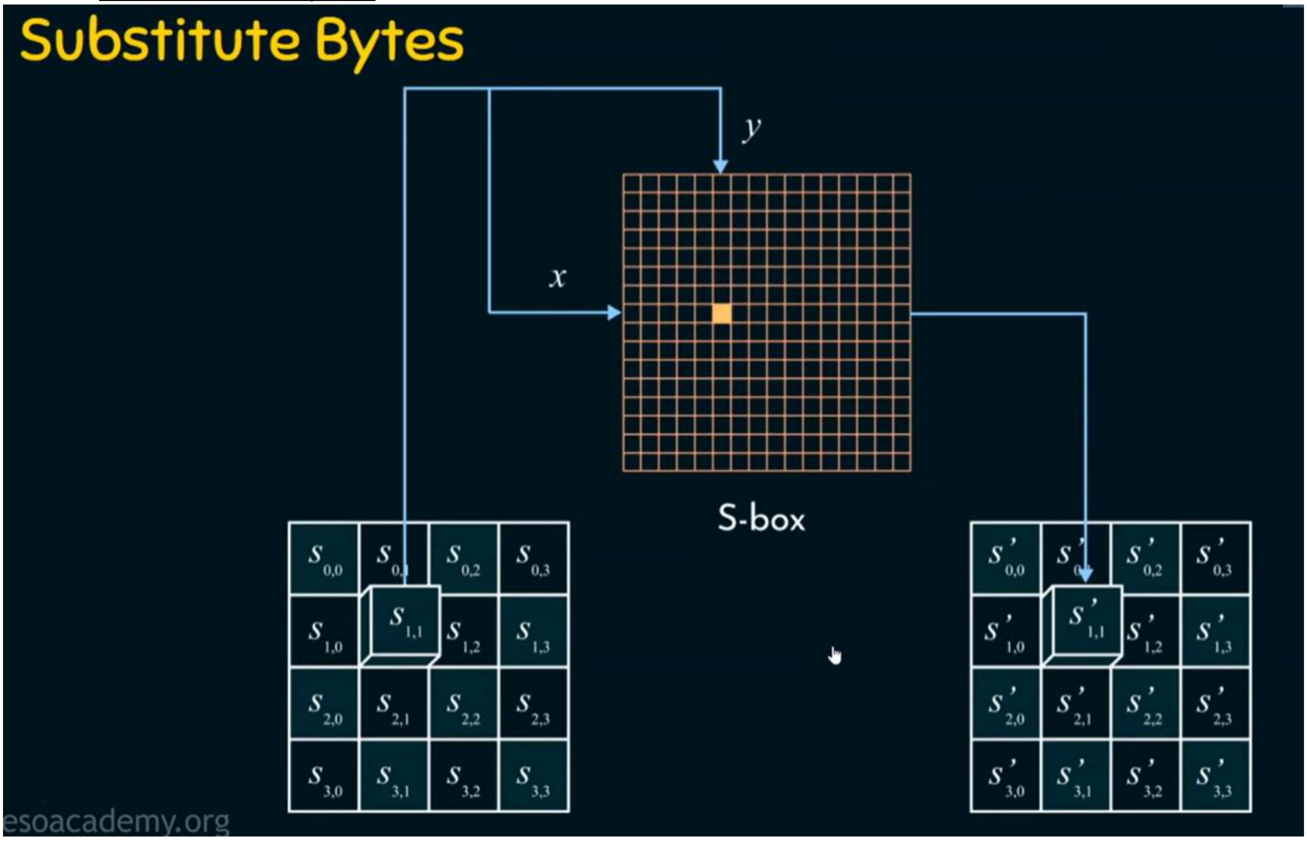
The round keys are generated using the key schedule algorithm that is a function of the main key. The transformation of rounds is explained in the following figure. Note that the initial transformation **<u>(having add round key only)</u>** and the final transformation **<u>(excluding the mix columns)</u>** are unique to make the process invertible.



The word of the round key is 32 bits, which means that for 10 rounds, the key expansion unit exported 44 words with 4 words for each round. Now let’s talk about each transformation individually.

Round	Words
Initial Transformation	w_0 w_1 w_2 w_3
Round 1	w_4 w_5 w_6 w_7
Round 2	w_8 w_9 w_{10} w_{11}
...	
Round 10	w_{40} w_{41} w_{42} w_{43}

➢ Substitute Bytes

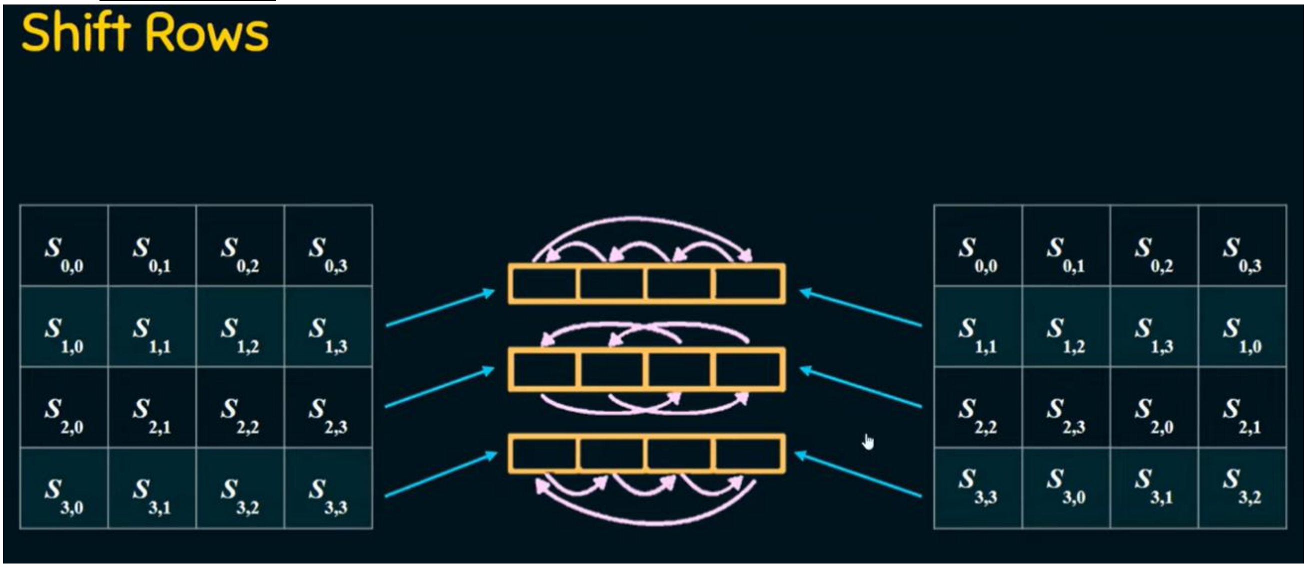


16x16 table is constructed where each element is represented by a byte that will be replaced by another value (ex. $S_{0,0} \to S_{\overline{0,0}}$) based on a look-up table shown in the following figure.

S-Box in AES

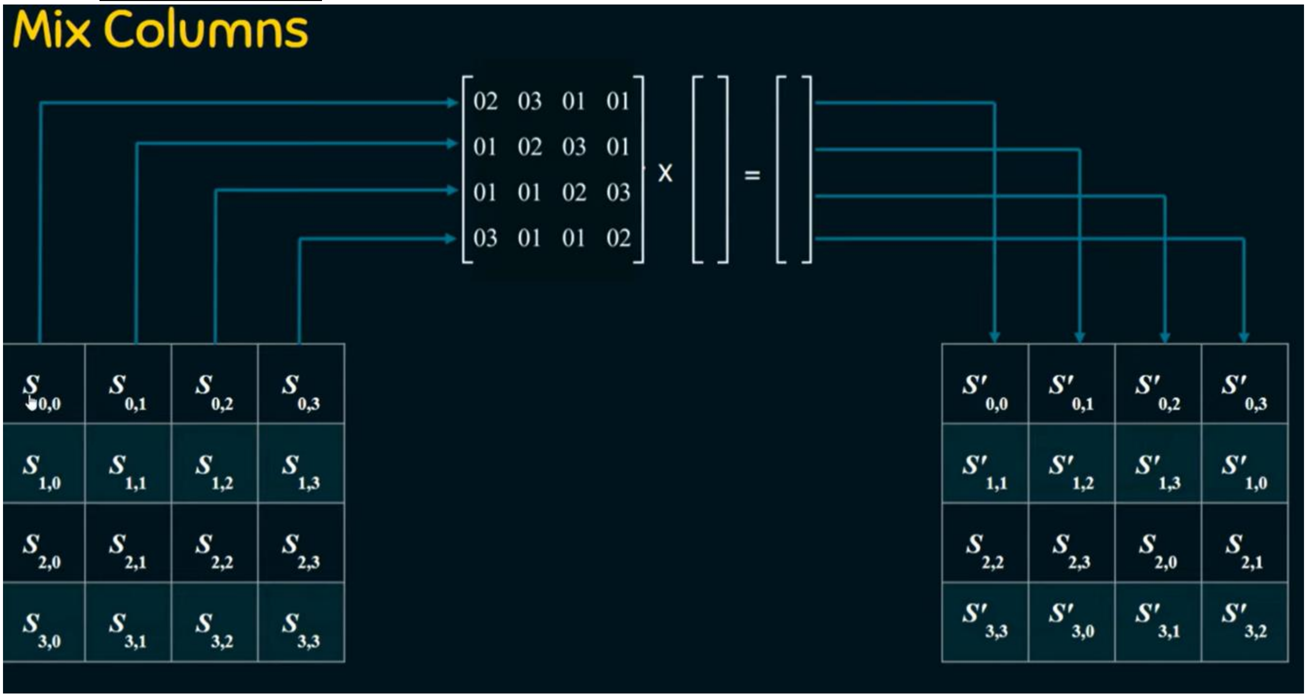
	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	$\boxed{4d}$	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

➢ Shift Rows



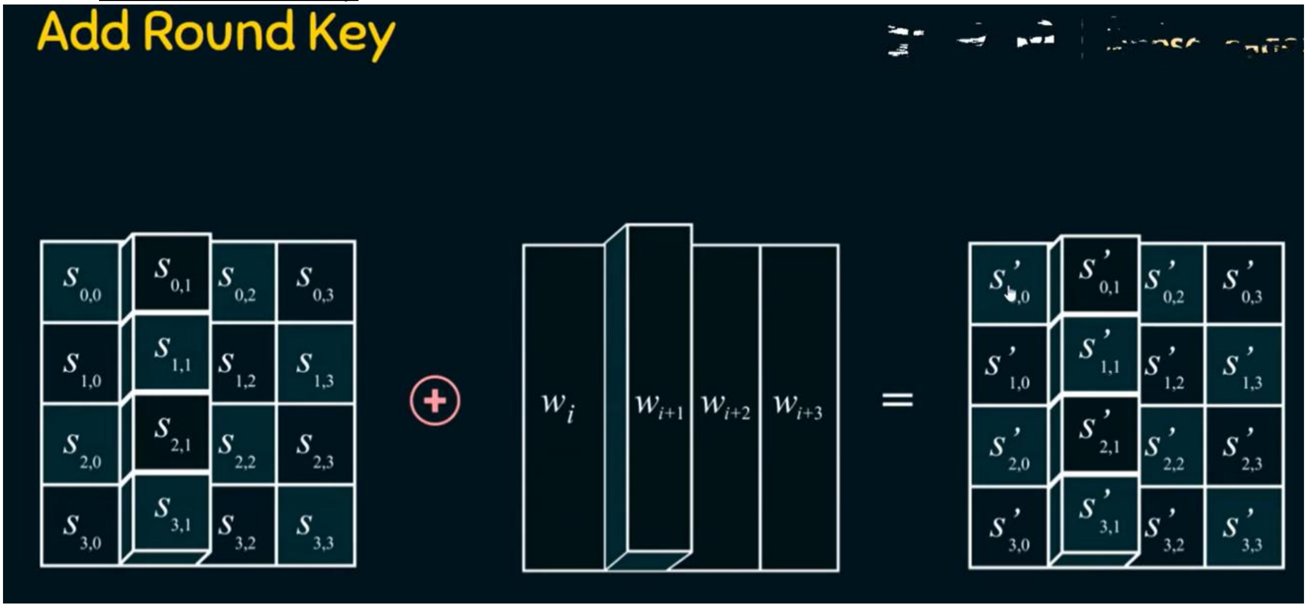
Based on the previous figure, the first row has 0 byte left circular shift, the second row has 1 byte left circular shift, the third row has 2 byte left circular shift, the fourth row has 3 byte left circular shift.

➢ Mix Columns



Matrix multiplication of the 4x4 data matrix with a pre-defined 4x4 matrix to have an output of 4x4 matrix.

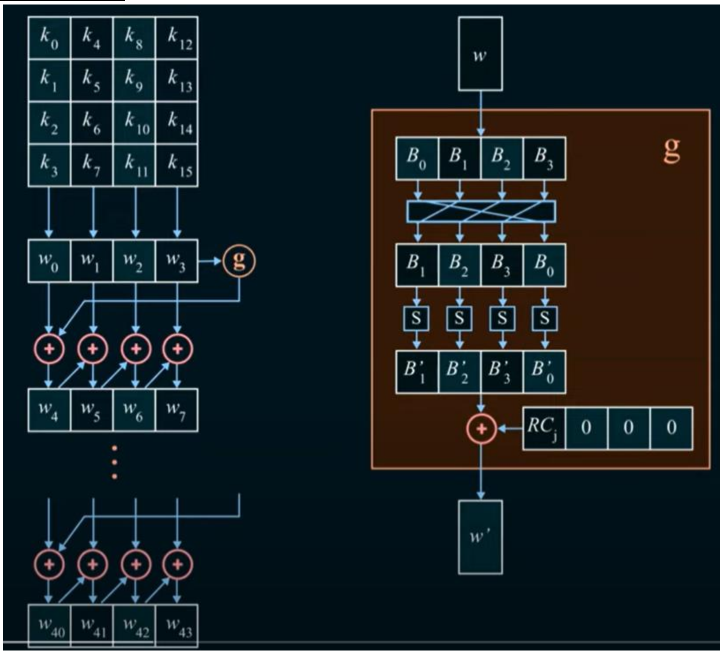
➢ Add Round Key



As shown before, the round key has 4 words where each word is 32 bits. The word is stored in columns to add (simple XOR operation) it with each row corresponding to the 4x4 matrix to have new 4x4 matrix.

Each of the four previously mentioned operations is performed once in each round with the specified order. But the question now is **<u>how to generate the round keys based on the main key</u>**?

➢ AES Key Expansion



We need 11 round keys if the key size is 128 bits or 13 if key size is 192 bits or 15 if key size is 256 bits (due to the initial transformation in the beginning of the encryption). For the initial transformation, the first four words are generated directly from the summation (or concatenation, not sure yet) of the main key columns to get four words. However, to achieve the rest of the words, we need to perform the 'G' function for the last word in the round key (for the initial round it would be W_3) then add the output of the 'G' function to the first word to get the next first word of the round key.

G Function

The word is divided into 4 bytes, where we perform 1 byte left circular shift at the beginning. Then we use the s-box (discussed earlier) to generate different bytes. Then we XOR the result with round constant (its generation is discussed briefly in the **AES definition** document) that has the following values for 128-bit key size.

Round Constant in AES–128

Round	RC	Round	RC
1	$(01\ 00\ 00\ 00)_{16}$	6	$(20\ 00\ 00\ 00)_{16}$
2	$(02\ 00\ 00\ 00)_{16}$	7	$(40\ 00\ 00\ 00)_{16}$
3	$(04\ 00\ 00\ 00)_{16}$	8	$(80\ 00\ 00\ 00)_{16}$
4	$(08\ 00\ 00\ 00)_{16}$	9	$(1B\ 00\ 00\ 00)_{16}$
5	$(10\ 00\ 00\ 00)_{16}$	10	$(36\ 00\ 00\ 00)_{16}$

Note: Initial Transformation takes $(00\ 00\ 00\ 00)_{16}$ as the RC.

Final note, visit this site to check the model correctness.