AES STRUCTURE

1. Plain text – 128 bits or 16 bytes

2. Key length
   - 16 bytes (128 bits)
   - 24 bytes (192 bits)
   - 32 bytes (256 bits)

3. As per keys algorithms are named as

   AES – 128
   AES – 192
   AES – 256

4. The input to the encryption and decryption algorithms is a single 128-bit block (4x4 square matrix of bytes).

each block is of 8 bits or 1 byte

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

→ State array

(5) State array is modified at each stage of encryption or decryption.

After final stage, State is copied to an output matrix.

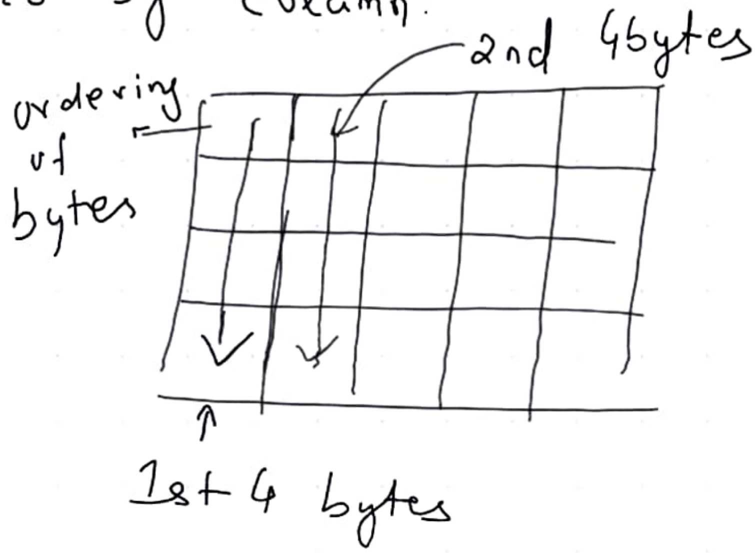
(6) Key is also depicted as a square matrix of bytes.

This key is then expanded into an array of key schedule words.

word = 4 bytes (32 bits).

(7) For AES-128 the total key schedule is 44 words.

Ordering of bytes within a matrix is by column.



(8) Similarly the first four bytes of the expanded key, which form a word, occupy the first column of the w matrix.

(9) N rounds ← (depends of the key size)

Key Size	N rounds
16 byte (128 bits)	10
24 byte (192 bits)	12
32 byte (256 bits)	14

(10) The first N-1 rounds consist of 4 distinct transformation functions:

(i) Substitution Bytes (SubBytes)

(ii) Shift Rows ($\rightleftarrows$)

(iii) Mix Columns (Matrix multiplication in GF(2^8)]

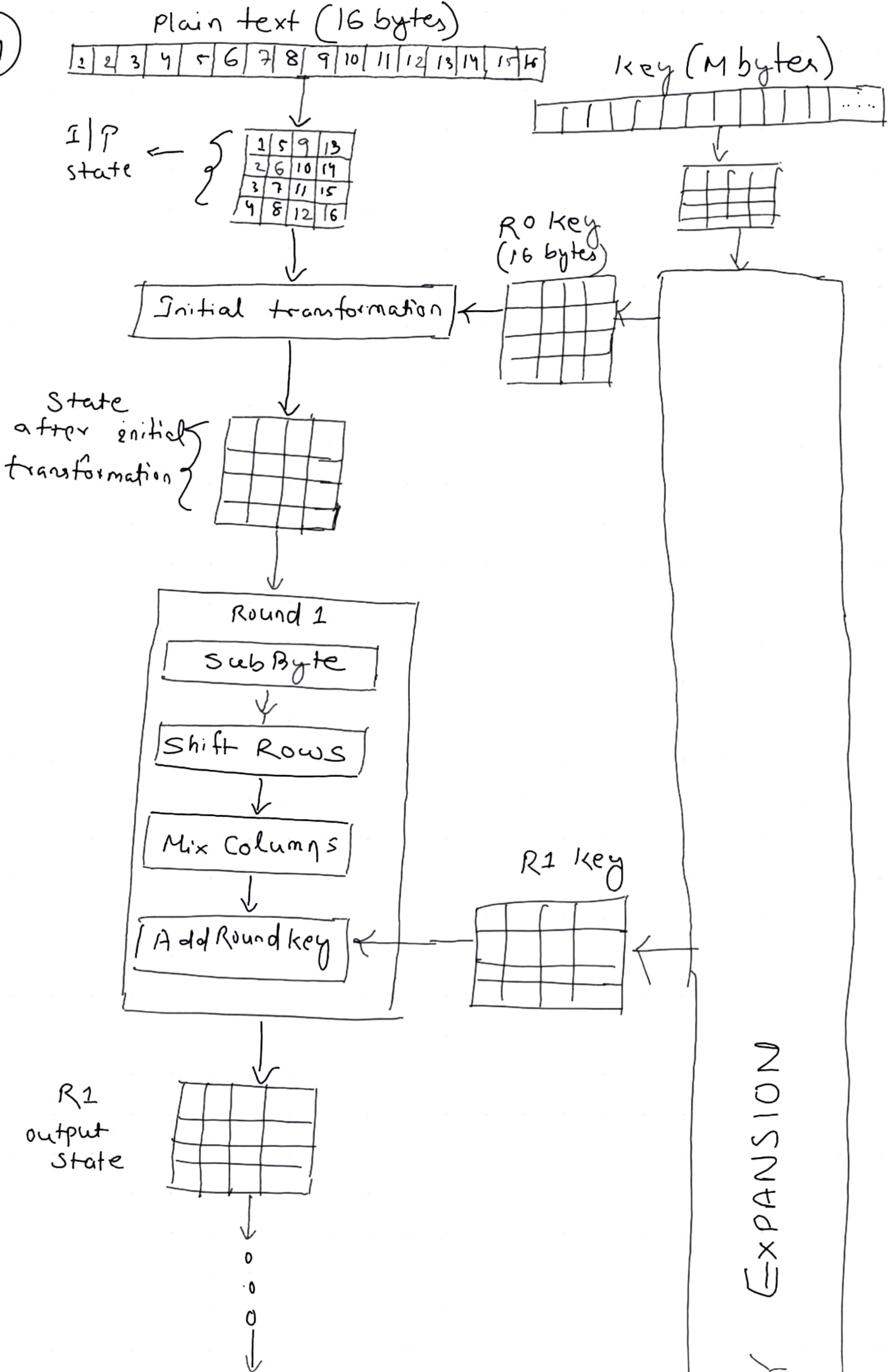
(iv) Add Round Key (bitwise XOR with part of key)

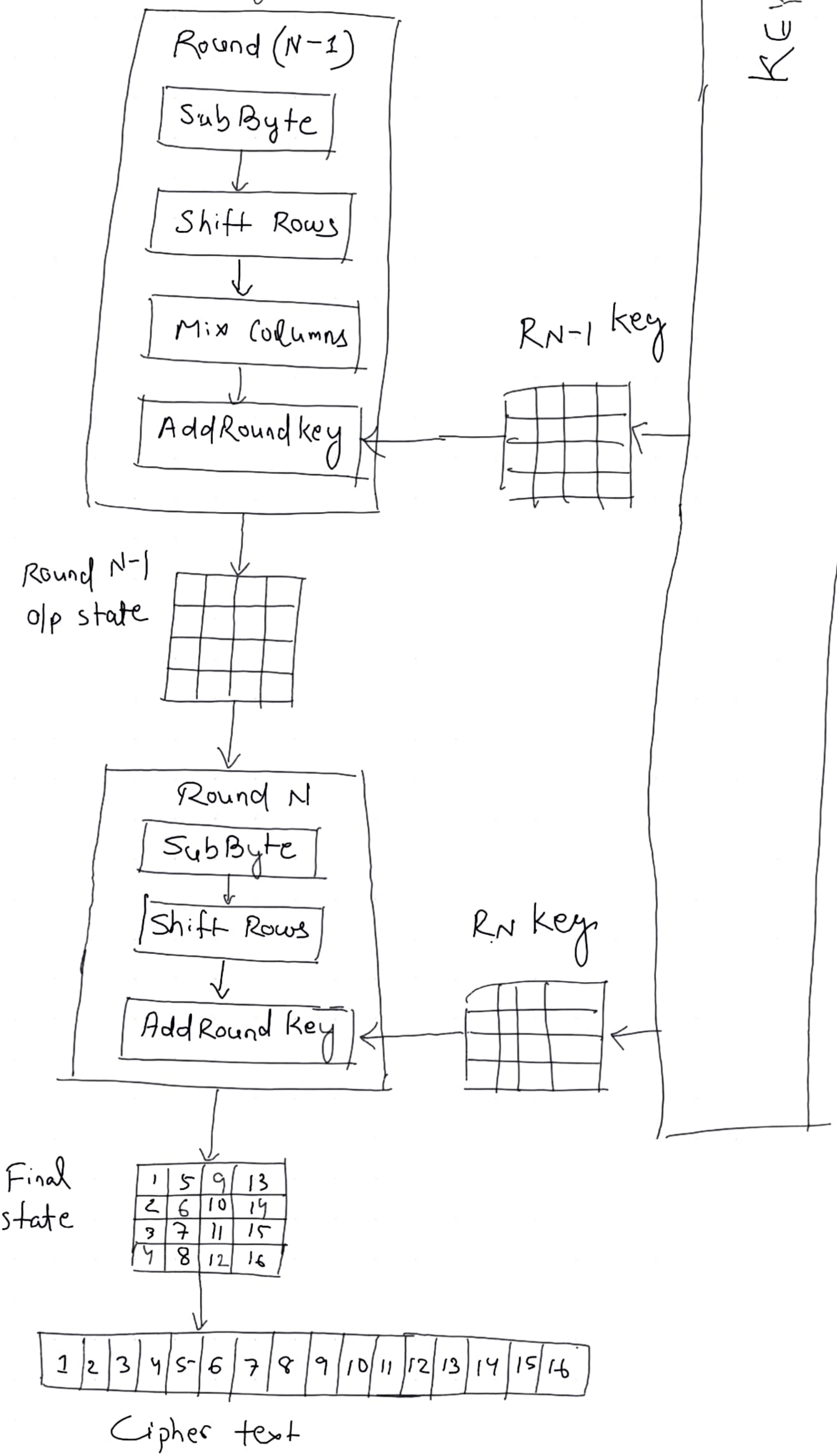
(11) Final round contains only 3 transformations and there is an initial single transformation (Add Round key) before the first round.

(Round 0)

(12) Each transformation takes one or more 4x4 matrices as input and produces a 4x4 matrix as output.

(13) Key expansion function generates N+1 round keys, each of which is a distinct 4x4 matrix. Each round key serves as one of the inputs to the AddRoundKey transformation in each round.


Plain text (16 bytes)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Key (M bytes)
I/P state
1 5 9 13
2 6 10 14
3 7 11 15
4 8 12 16
R0 Key (16 bytes)
Initial transformation
State after initial transformation
Round 1
SubByte
Shift Rows
Mix Columns
AddRoundkey
R1 key
R1 output state
EXPANSION



Round (N-1)
SubByte
Shift Rows
Mix Columns
AddRoundkey
RN-1 key
Round N-1
o/p state
Round N
SubByte
Shift Rows
AddRound key
RN key
Final
state
1 5 9 13
2 6 10 14
3 7 11 15
4 8 12 16
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Cipher text

(15) It is not a feistel Structure.

Only one half of the plain text was operated. → Right block

Here whole plain text is a single matrix and all the operations are carried out on the whole matrix.

(16) SubByte → Confusion (Byte by Byte) → 1 s-block (s-box)

Shift Rows → Diffusion

Mix Columns → Confusion in $GF(2^8)$

Add Round key → XOR (bitwise)

(17) Decryption

Each stage is easily reversible.

Substitute Byte.
Shift Rows —
Mix Columns —

} Inverse function is used in decryption algorithm

XORING Same Round key.

(18) SubByte

Byte at row y col x → $\boxed{y\,x}$

↓

$\boxed{\text{Inverse in } GF(2^8)}$

↓

$\boxed{\text{Byte to bit vector}}$

$$
\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix}
=
\begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
=
\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix}
+
\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}
$$

↓

$\boxed{\text{Bit column vector to byte}}$

↓

$\boxed{S(y\,x)}$

Inverse SubByte

Byte at row y
col x

YX

↓

Byte to bit column vector

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

↓

Bit column vector to Byte

↓

Inverse in $GF(2^8)$

↓

IS (yx)

(19) <u>Shift Rows</u>

b_1	b_5	b_9	b_{13}	
b_2	b_6	b_{10}	b_{14}	← 1bit left shift
b_3	b_7	b_{11}	b_{15}	← 2bit " "
b_4	b_8	b_{12}	b_{16}	← 3bit " "

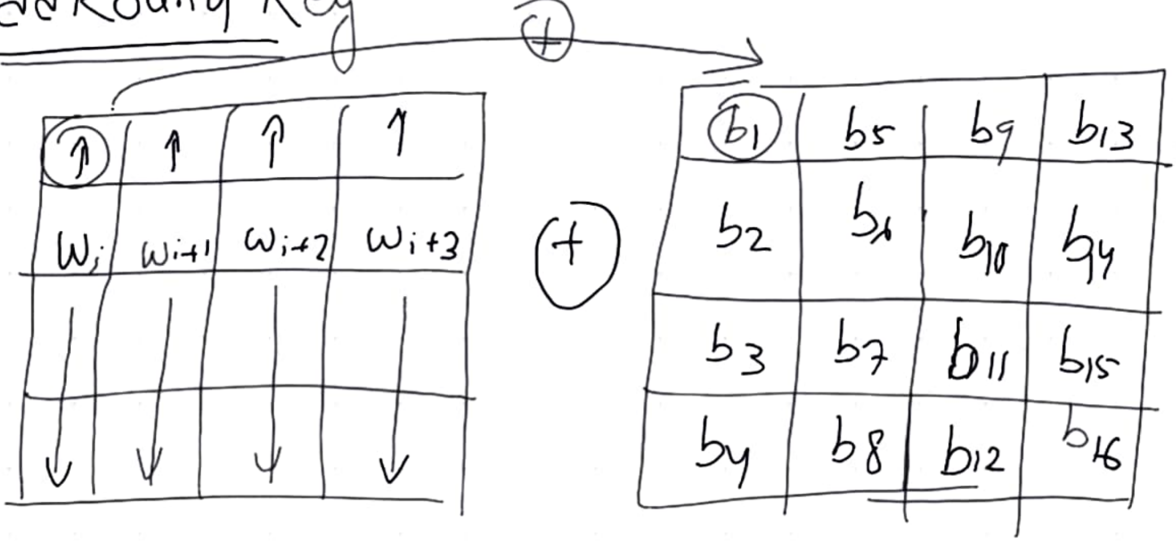
b_1	b_5	b_9	b_{13}
b_6	b_{10}	b_{14}	b_2
b_{11}	b_{15}	b_3	b_7
b_{16}	b_4	b_8	b_{12}

(20) Mix Columns

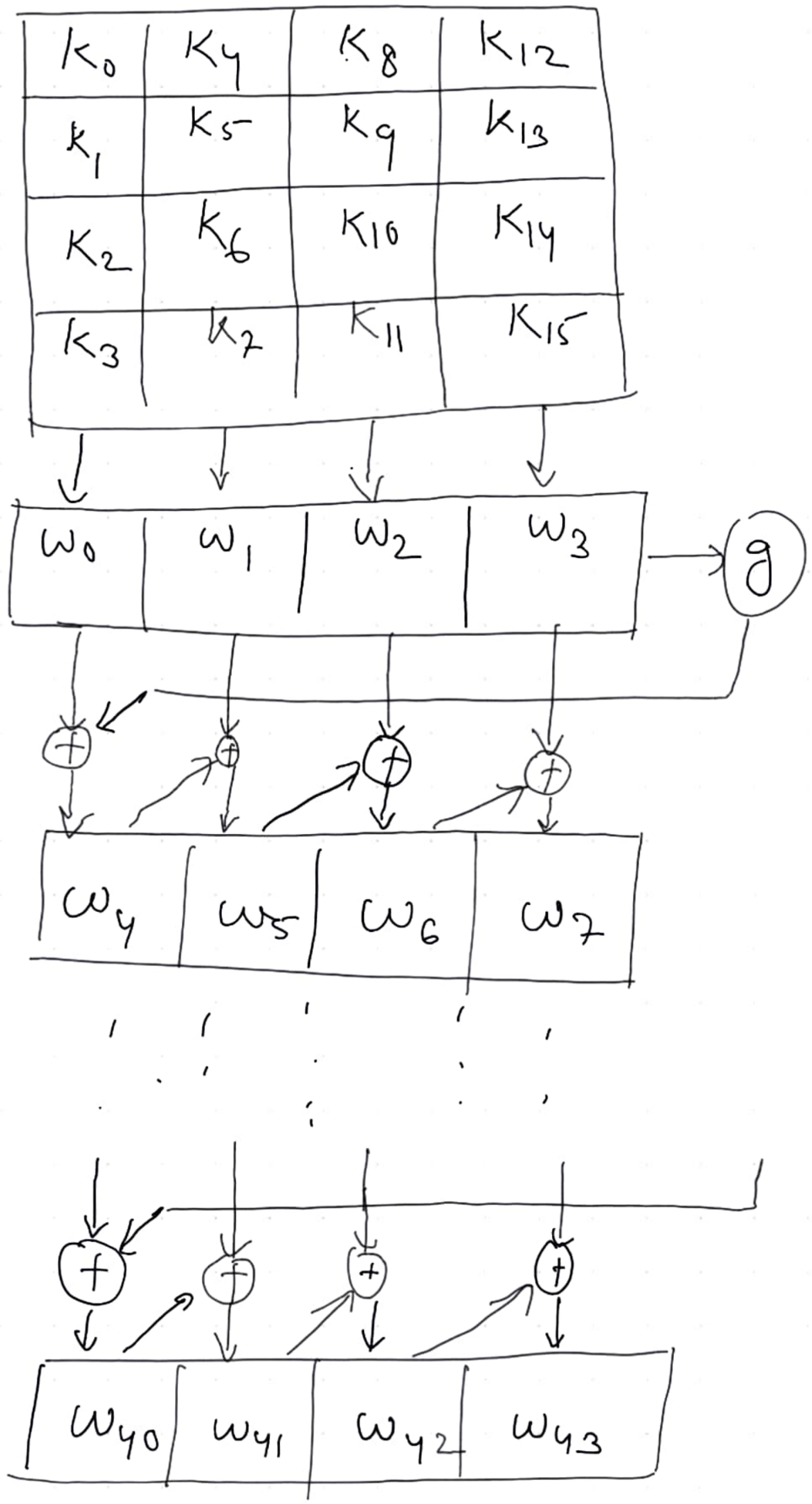
$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \\ b_4 & b_8 & b_{12} & b_{16} \end{bmatrix}$$

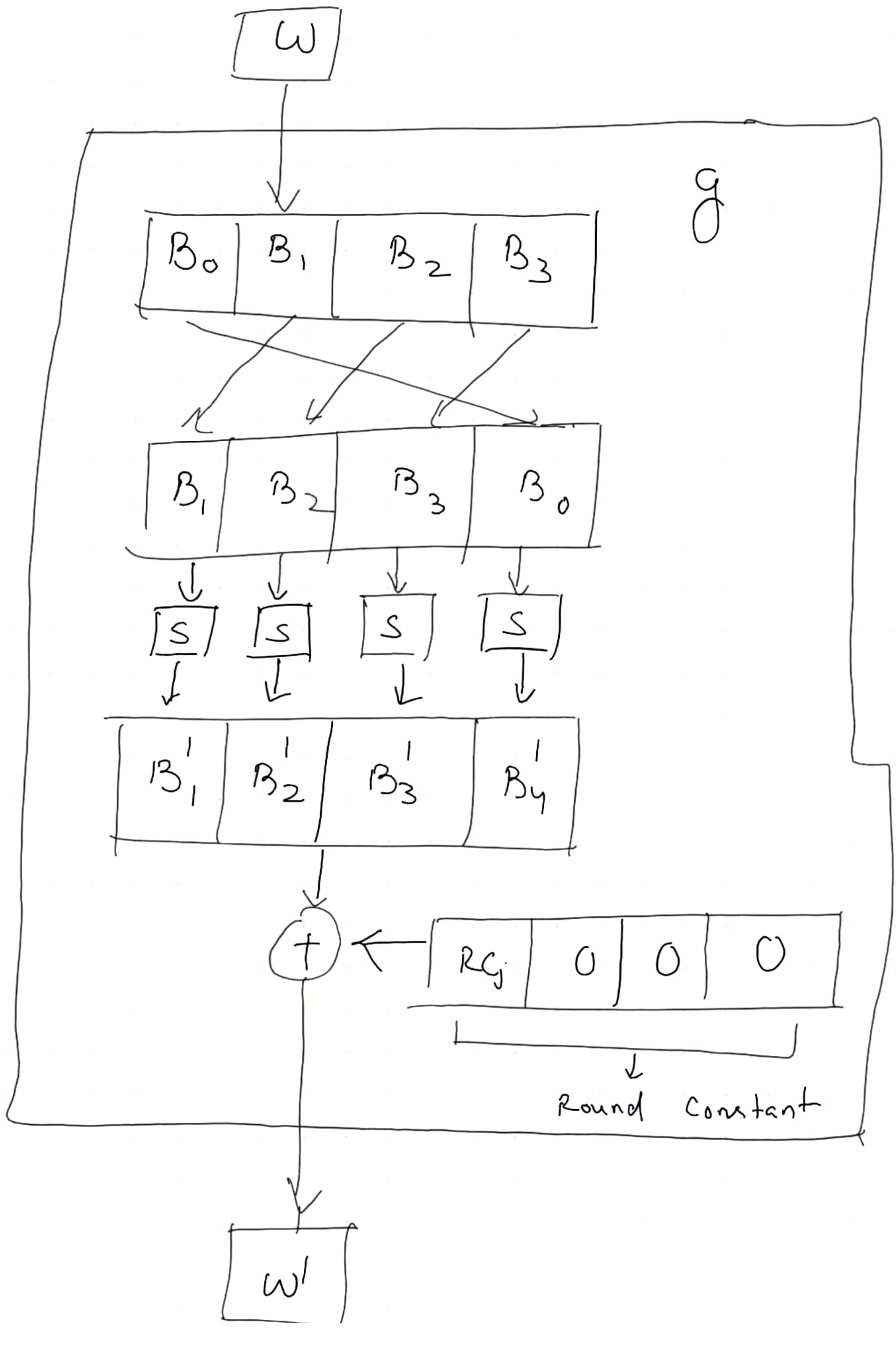
$$= \begin{bmatrix} b_1' & b_5' & b_9' & b_{13}' \\ b_2' & b_6' & b_{10}' & b_{14}' \\ b_3' & b_7' & b_{11}' & b_{15}' \\ b_4' & b_8' & b_{12}' & b_{16}' \end{bmatrix}$$

(21) AddRound Key



22 Key Expansion




W
g
B0 B1 B2 B3
B1 B2 B3 B0
S S S S
B'1 B'2 B'3 B'4
+
RCj 0 0 0
Round Constant
W'

j	1	2	3	4	5	6	7	8	9	10
RC_j	01	02	04	08	10	20	40	80	1B	36

$GF(2^8)$ plane

1000 0000 ← 80

0001 1011 ← 80 x 2 mod m(x)

1 0000 0000 ← 80 x 2

1 0001 1011 ← M(x)

0001 1011 ← 80 x 2 mod m(x)