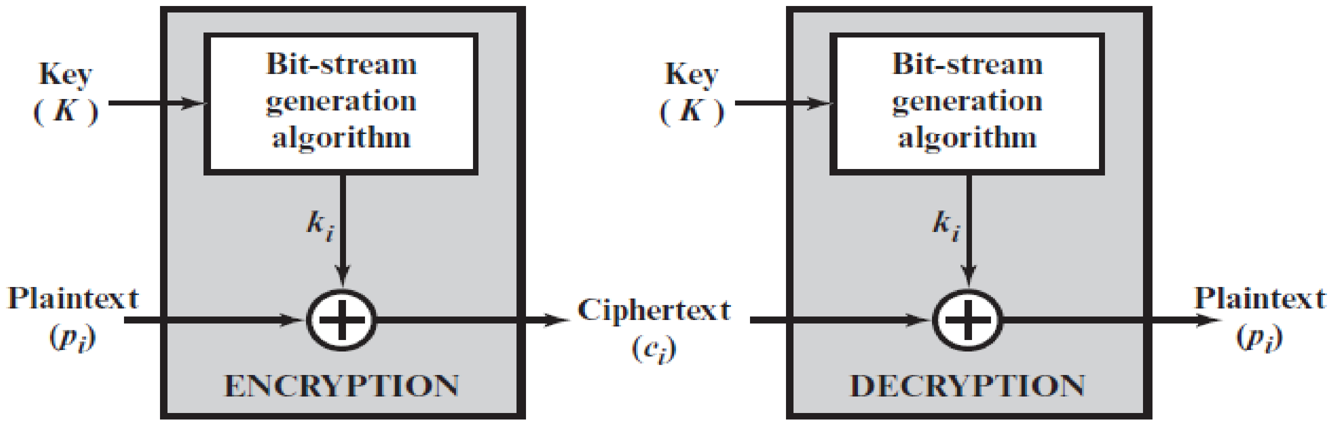
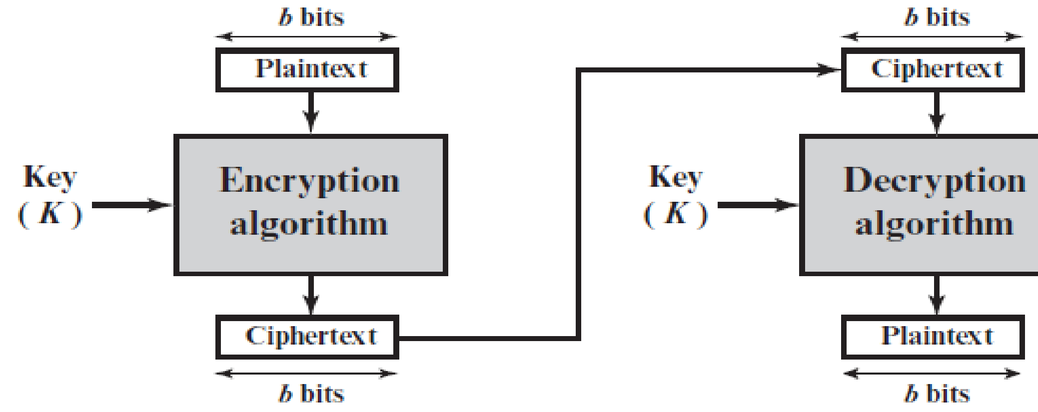
Stream Cipher and Block Cipher



(a) Stream cipher using algorithmic bit-stream generator



(b) Block cipher

Differences between Stream and Block Ciphers

Stream	Block
One Bit or Byte is processed (encrypted or decrypted) at a time	Fixed size block is processed at a time (64 bits, 128 bits, etc.)
Usually Variable Key size	A given algorithm will have a fixed key size
Padding is not required	Most probably requires padding
Processing is faster	Processing is slower
Error Propagation is limited	Error propagation could be significant and can produce a completely different output
Examples:- RC4, Rabbit, ChaCha, etc.	Examples:- AES, DES, Triple DES, Serpent, etc.
Used for real-time data stream encryptions and decryptions, where minimum latency is expected.	Used for applications which can handle large data chunks.

MOTIVATION FOR FEISTEL CIPHER STRUCTURE

n-bit to n-bit Block Substitution

- Block Cipher (n-bit PT block ➡ n-bit CT block, and vice-versa)
- Number of PT blocks possible = 2^n.
- Decryption is possible when each PT block can produce a unique CT block (Reversible or Non-Singular Transformation).

- Reversible set of Transformations when n = 2:-

PT Block	CT Block
00	11
01	10
10	00
11	01

n-bit to n-bit Block Substitution (Contd..)

- Irreversible set of Transformations when n = 2:-

PT Block	CT Block
00	11
01	10
10	01
11	01

- Number of Reversible sets of transformations = $2^n!$

4-bit to 4-bit Block Substitution Example

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

n-bit to n-bit Block Substitutions (Contd..)

- We noticed with n=2, and n=4, that the corresponding tables define straightforward mappings between PT blocks and CT blocks.
- Feistel called these types of mappings as an ideal block cipher.
- Increasing the block size makes the block cipher more resistant to cryptanalysis.
- The mappings can be defined by a key whose length is $n*2^n$ bits.
- However, when n is huge, Key-Management becomes cumbersome.

BLOCK CIPHER DESIGN PRINCIPLES

Principles of designing a Block Cipher

1) Confusion
2) Diffusion
3) Avalanche Effect
4) Feistel Structure
5) Round Functions
6) Key Size and Block Size
7) Bit Independence Criterion (BIC)
8) Key Schedule Algorithm

Confusion

- Term was introduced by Claude Shannon.
- The property makes the relationship between the CT and the key as complex as possible.
- Achieved through Complex Key-Schedule and Substitution.
- Each bit of the CT depends on several bits of the key.

Diffusion

- The term was introduced by Claude Shannon.
- The property makes the relationship between the CT and PT complex.
- Achieved through mixing operations and Permutation.
- One bit change in the PT will result in significant change in the bits of CT.

Avalanche Effect

- Property which ensures that a small change in any of the Inputs results in a significant and unpredictable change in the Output.
- Achieved through Confusion and Diffusion.

Feistel Structure

- Used to achieve Confusion and Diffusion in an organized manner.
- Involves division of each block, operations through multiple rounds, and swapping operations.

Round Functions

- Internal functions used for each round of a Cipher.
- The functions are complex and foundations for Avalanche Effect.
- Involves Substitutions, Permutations, and Mixing Operations.

Key Size and Block Size

- Plays a crucial role in security.
- Larger the key, more is the Cipher resistant to BFA.
- Larger the PT block, more is the Cipher resistant to Statistical and Pattern attack.

BIC

- More associated with cryptographic hash functions and PRNGs.
- However, applicable to Block Cipher design as well.
- Provides statistically random output.
- Provides negligible or 0 predictable relationships between the current and the neighboring bits.

Key Schedule Algorithm

- Round keys are derived from a Master Key.
- Crucial Component in many block ciphers like DES, AES, etc.
- A good algorithm should generate unique and random round keys.
- Generated using different operations like Substitutions, Permutations, Mixing, Bit shifting operations, etc.

FEISTEL CIPHER

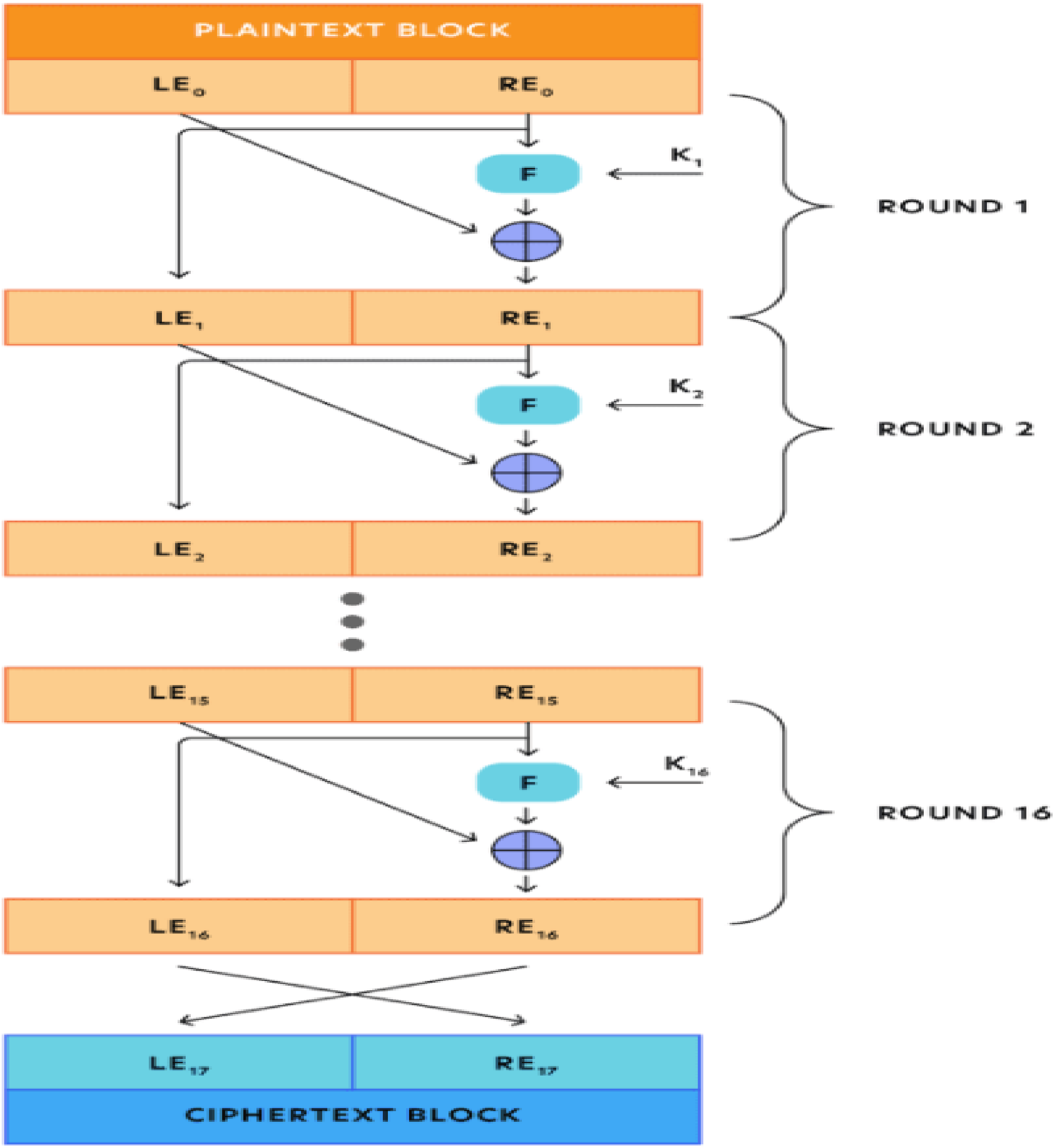
Feistel Cipher

- Key Length = k bits
- Number of sets of Transformations = 2^k
- Alternates Substitution and Permutation operations.
- Can have any number of rounds with same structure.

Design Features to be considered in Feistel Cipher

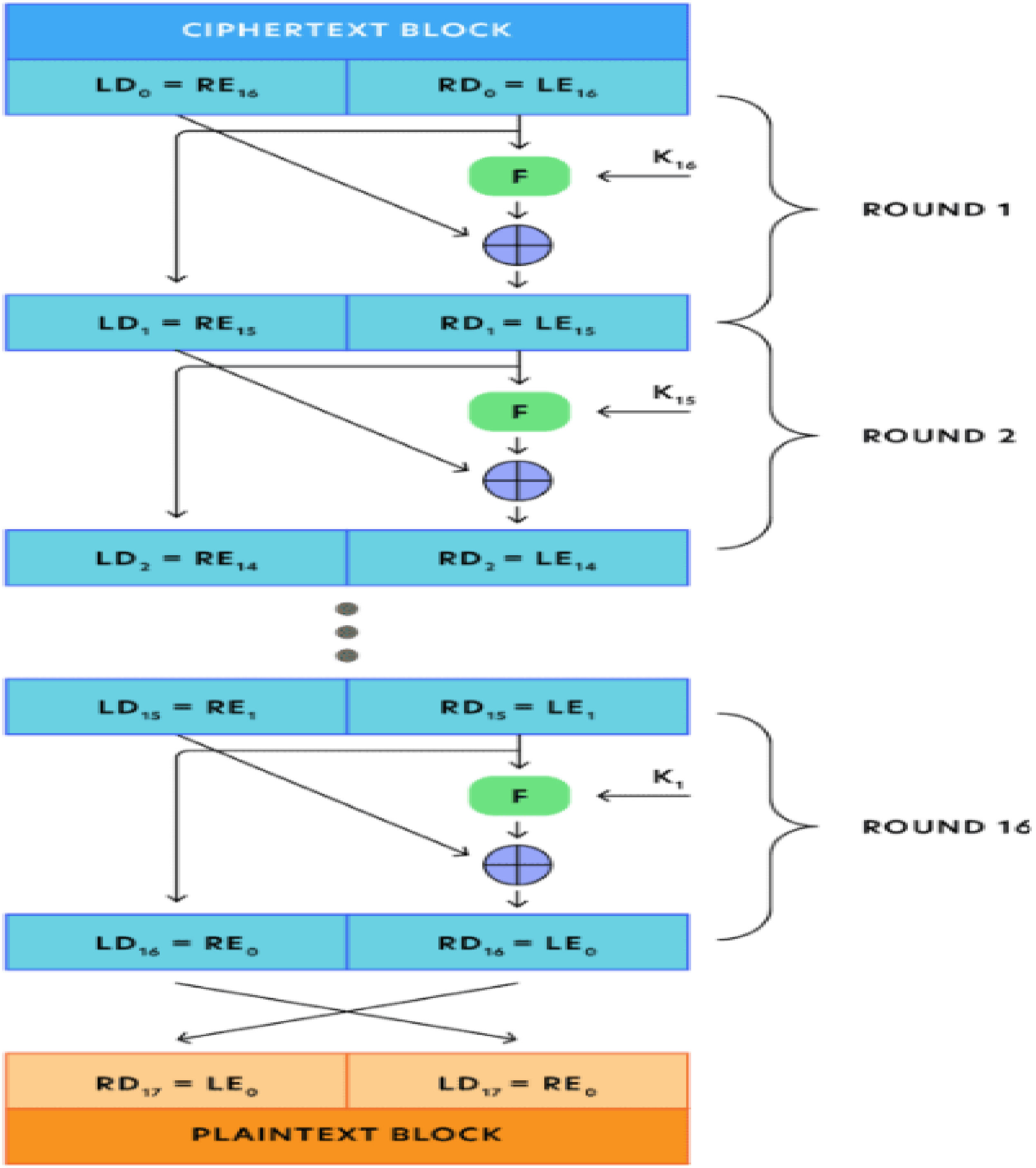
- Block size
- Ease of Analysis
- Key Size
- Sub-Keys generation Function
- Number of Rounds
- Round Function
- Fast software encryption/decryption

Encryption



PLAINTEXT BLOCK
LE_0
RE_0
F
K_1
ROUND 1
LE_1
RE_1
F
K_2
ROUND 2
LE_2
RE_2
LE_{15}
RE_{15}
F
K_{16}
ROUND 16
LE_{16}
RE_{16}
LE_{17}
RE_{17}
CIPHERTEXT BLOCK

Decryption



CIPHERTEXT BLOCK
$LD_0 = RE_{16}$
$RD_0 = LE_{16}$
F
K_{16}
ROUND 1
$LD_1 = RE_{15}$
$RD_1 = LE_{15}$
F
K_{15}
ROUND 2
$LD_2 = RE_{14}$
$RD_2 = LE_{14}$
$LD_{15} = RE_1$
$RD_{15} = LE_1$
F
K_1
ROUND 16
$LD_{16} = RE_0$
$RD_{16} = LE_0$
$RD_{17} = LE_0$
$LD_{17} = RE_0$
PLAINTEXT BLOCK

Observations in Encryption and Decryption

- Assume that n is the number of rounds.
- During Encryption, $LE_i = RE_{i-1}$
- During Encryption, $RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i)$
- During Decryption, $LD_i = RD_{i-1}$
- During Decryption, $RD_i = LD_{i-1} \oplus F(RD_{i-1}, K_{n-i+1})$
- $LD_i = RE_{n-i}$
- $RD_i = LE_{n-i}$

Generation of Round Keys from Master Key

- The round keys (sub-keys) K_i are derived from the master key K (The process is called Key-Expansion)

- Each Round uses a unique round-key.

- The Key-Expansion uses operations like Permutations, Substitutions, and other transformations.

Example 1

- Assume that the PT = 0x3C; K_1 = 0xF, K_2 = 0xA. F(x,y) = Bitwise Logical AND of x and y. What are the outputs of the first 2 rounds of Feistel Cipher encryption?

Solution:-

- PT = 0x3C
- $LE_0 = (0011)_2$
- $RE_0 = (1100)_2$

Example 1 (Contd..)

Round 1:-

- $LE_1 = (1100)_2$
- $RE_1 = LE_0 \oplus F(RE_0, K_1)$
- $RE_1 = (0011)_2 \oplus F((1100)_2,(1111)_2)$
- $RE_1 = (0011)_2 \oplus (1100)_2$
- $RE_1 = (1111)_2$
- *Therefore, Round 1 Output = 0xCF*

Example 1 (Contd..)

Round 2:-

- $LE_2 = (1111)_2$
- $RE_2 = LE_1 \oplus F(RE_1, K_2)$
- $RE_2 = (1100)_2 \oplus F((1111)_2,(1010)_2)$
- $RE_2 = (1100)_2 \oplus (1010)_2$
- $RE_2 = (0110)_2$
- *Therefore, Round 2 Output = 0xF6.*

Example 2

- Assume that Feistel Cipher uses 16 rounds. The output of the 14^{th} Round is 0x8D; K_{15} = 0x7, K_{16} = 0xC. F(x,y) = Logical OR of x and y. Calculate CT.

Solution:-

- $LE_{14} = (1000)_2$
- $RE_{14} = (1101)_2$

Example 2 (Contd..)

Round 15:-

- $LE_{15} = RE_{14} = (1101)_2$
- $RE_{15} = LE_{14} \oplus F(RE_{14}, K_{15})$
- $RE_{15} = (1000)_2 \oplus F((1101)_2, (0111)_2)$
- $RE_{15} = (1000)_2 \oplus (1111)_2$
- $RE_{15} = (0111)_2$

Example 2 (Contd..)

Round 16:-

- $LE_{16} = RE_{15} = (0111)_2$
- $RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$
- $RE_{16} = (1101)_2 \oplus F((0111)_2, (1100)_2)$
- $RE_{16} = (1101)_2 \oplus (1111)_2$
- $RE_{16} = (0010)_2$
- Output of Round 16 = 0x72

- *Therefore, CT = 0x27*

Example 3

- Assume that Feistel Cipher uses 16 rounds. The CT is 0xAB, K_{16} = 0x7. The Round Function F(x,y) = bitwise XOR (1-bit right rotation of x, 1-bit right rotation of y). What's the output of the first round during decryption?

Solution:-

- $LD_0 = (1010)_2$
- $RD_0 = (1011)_2$

Example 3 (Contd..)

- $LD_1 = RD_0 = (1011)_2$
- $RD_1 = LD_0 \oplus F(RD_0, K_{16})$
- $RD_1 = (1010)_2 \oplus F((1011)_2, (0111)_2)$
- $RD_1 = (1010)_2 \oplus ((1101)_2 \oplus (1011)_2)$
- $RD_1 = (1010)_2 \oplus (0110)_2$
- $RD_1 = (1100)_2$

- *Therefore, Output of First Round of decryption is 0xBC.*

Example 4

- Assume that Feistel Cipher uses 16 rounds. The output of 15th Round of decryption is 0xABCD, K_1 = 0xE9. The Round Function F(x,y) = bitwise XOR (bitwise NOT of x, bitwise NOT of y). What's the deciphered text?

Solution:-

- LD_{15} = 0xAB = $(10101011)_2$
- RD_{15} = 0xCD = $(11001101)_2$
- $K_1 = (11101001)_2$

Example 4 (Contd..)

- $LD_{16} = RD_{15} = (11001101)_2$
- $RD_{16} = LD_{15} \oplus F(RD_{15}, K_1)$
- $RD_{16} = (10101011)_2 \oplus F[(11001101)_2, (11101001)_2]$
- $RD_{16} = (10101011)_2 \oplus [(00110010)_2 \oplus (00010110)_2]$
- $RD_{16} = (10101011)_2 \oplus (00100100)_2$
- $RD_{16} = (10001111)_2 = 0x8F$

- *Therefore, the Deciphered text is 0x8FCD*

S-BOXES AND P-BOXES

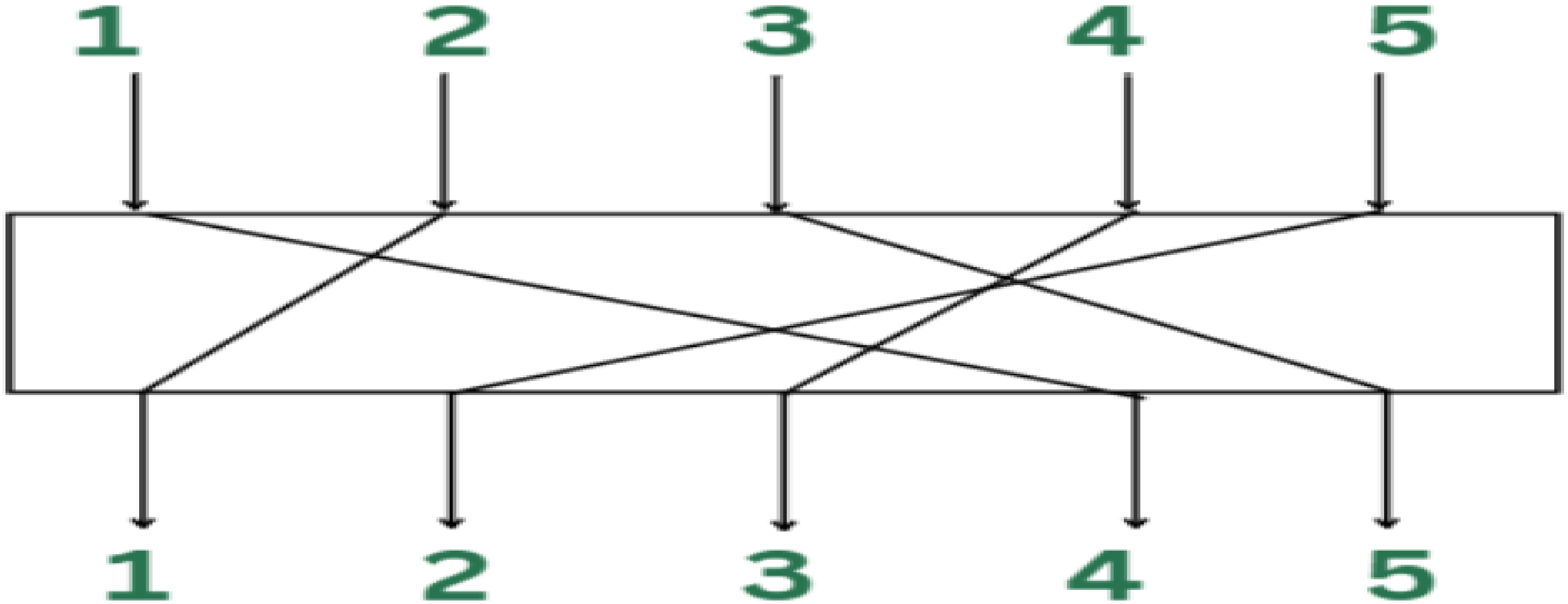
S-Boxes

- Uses Non-Linear Transformation to generate Outputs from Inputs.
- Mapping between inputs and corresponding outputs are defined by a table or a matrix.
- It should be such that its not easily invertible by an attacker.
- Two main categories of S-Boxes:- Static S-Box and Dynamic S-Box

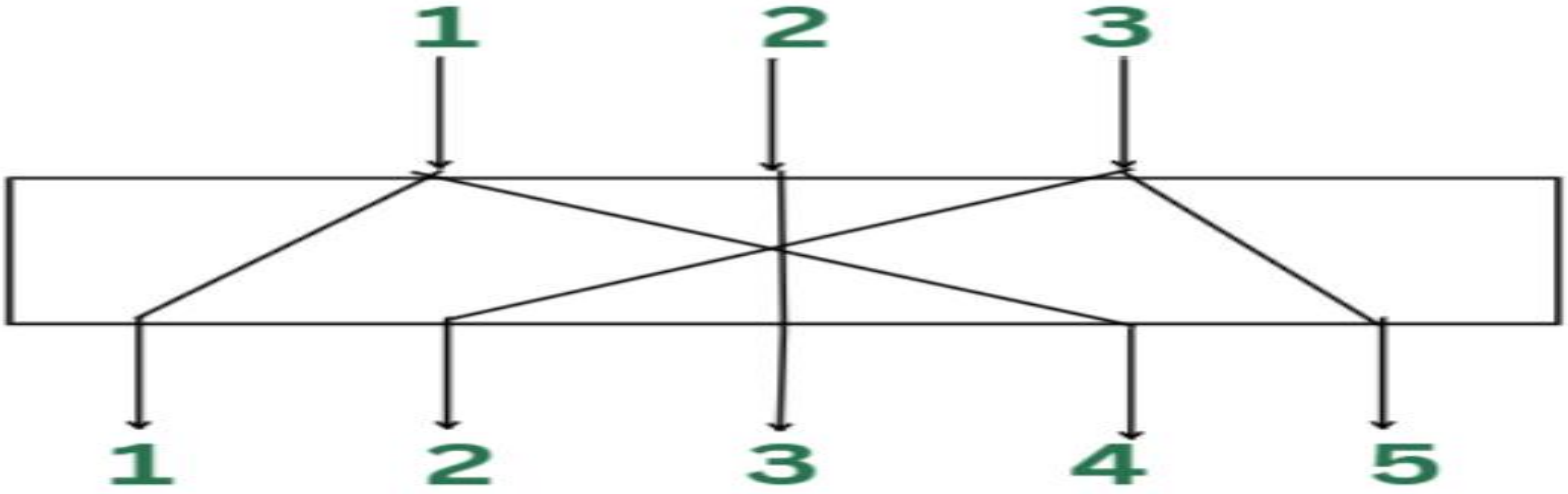
P-Boxes

- The primary goal is to increase diffusion for a cipher, by permuting the input bits.
- Permutation of the bits makes the cryptanalysis more challenging.
- Types of P-Boxes: Straight P-Box, Compression P-Box, and Expansion P-Box.

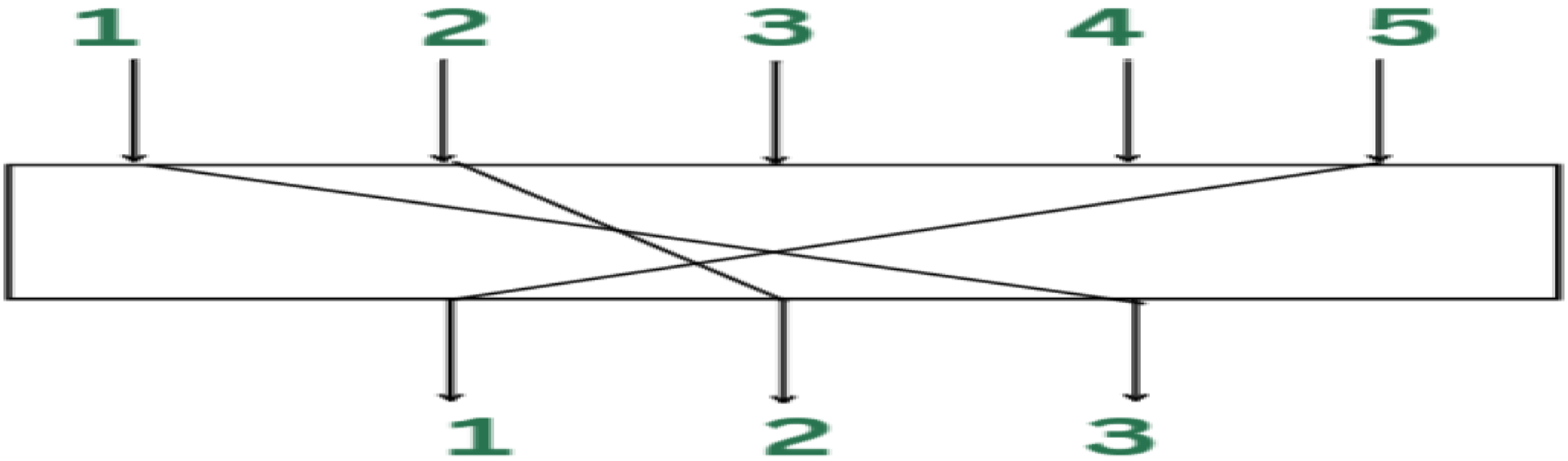
Straight P-Box



Expansion P-Box



Compression P-Box

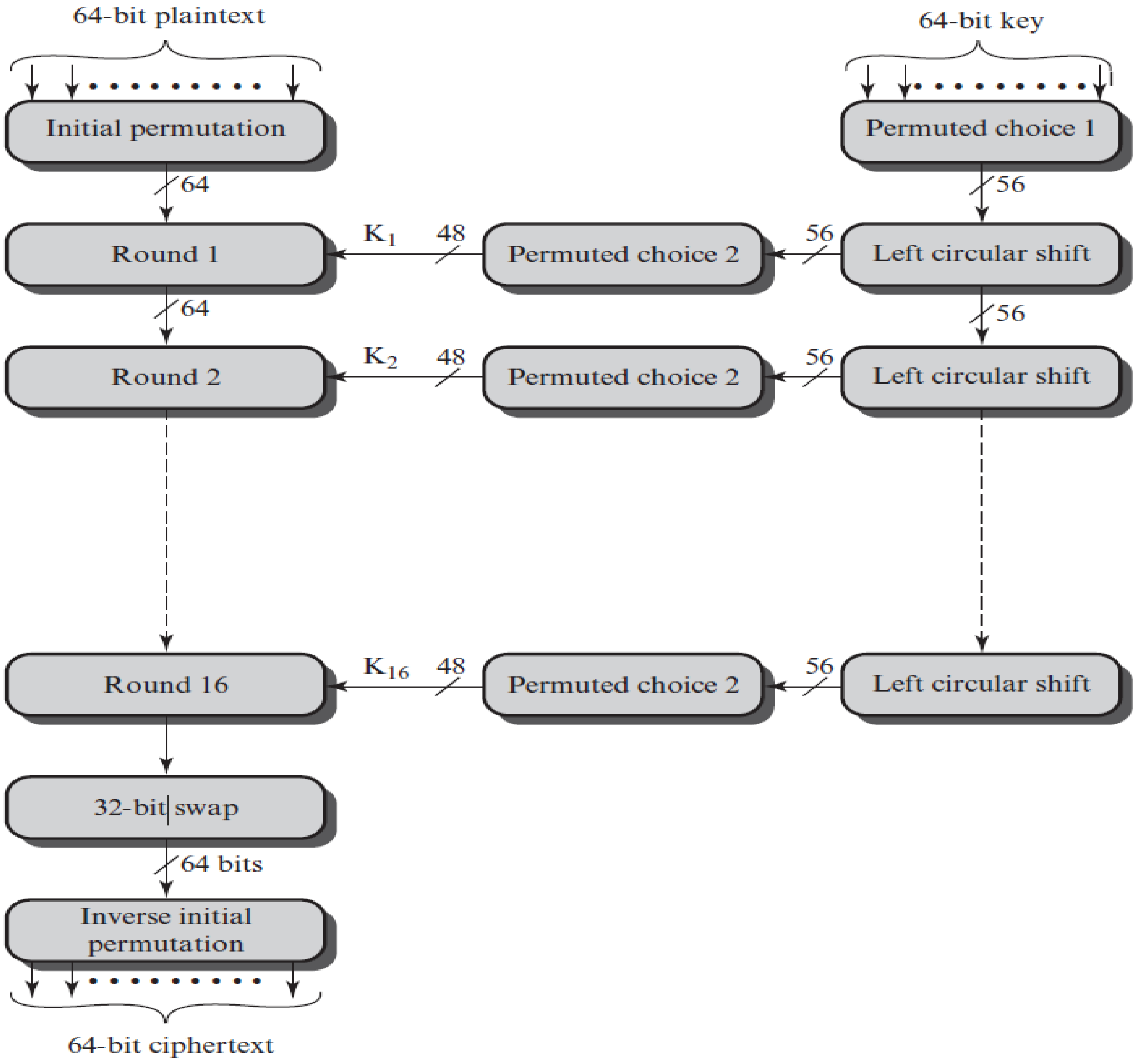


Data Encryption Standard (DES)

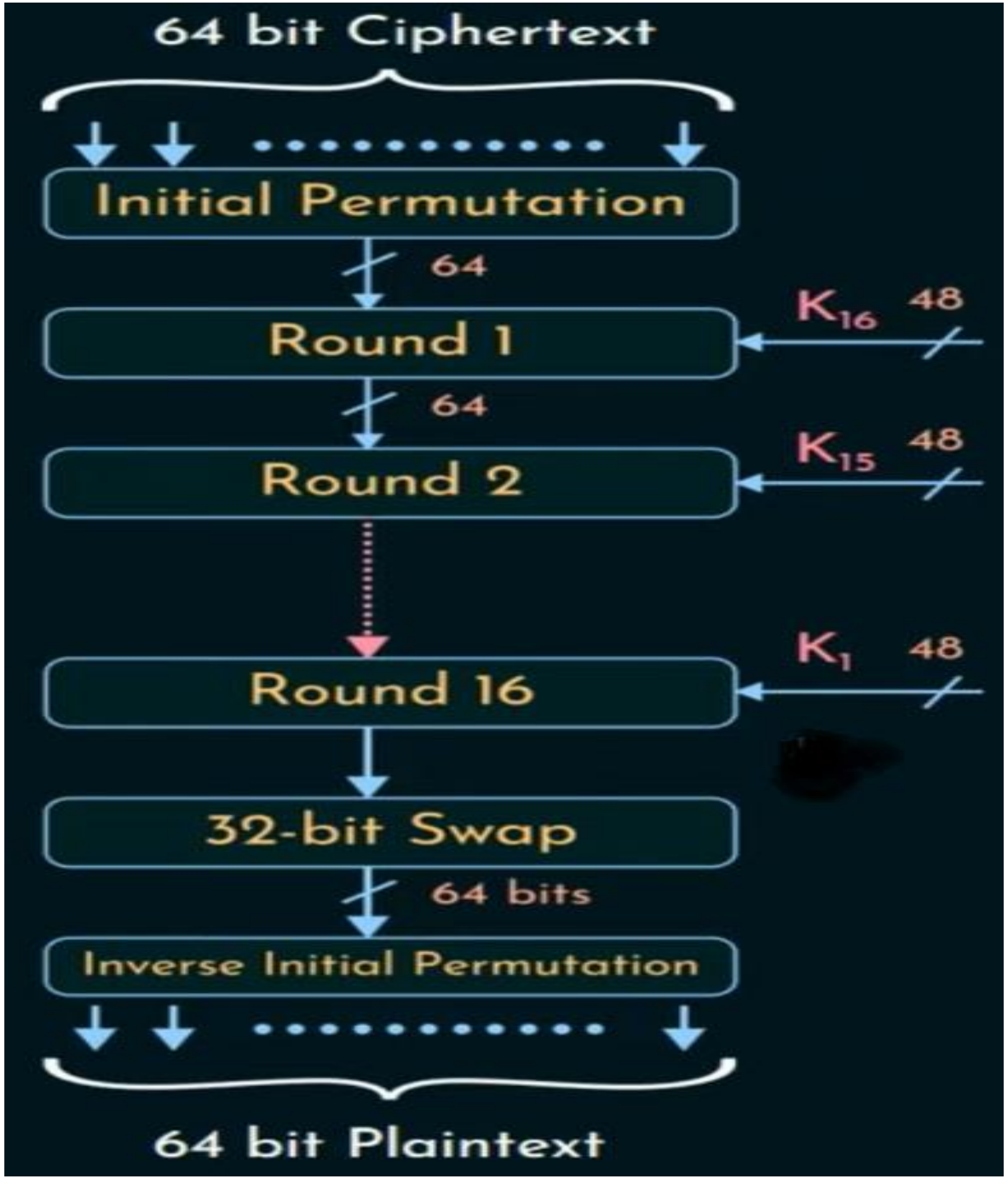
DES

- Developed by IBM.
- Adopted as a federal standard by NIST in 1977.
- Input = 64-bit block
- Output = 64-bit block
- Original Key Length = 64 bits
- Effective Key Length = 56 bits
- Round Key = 48 bits
- Consists of 16 rounds.
- Each Round consists of different operations like Substitution, Permutation, Key-Mixing, and Expansion.

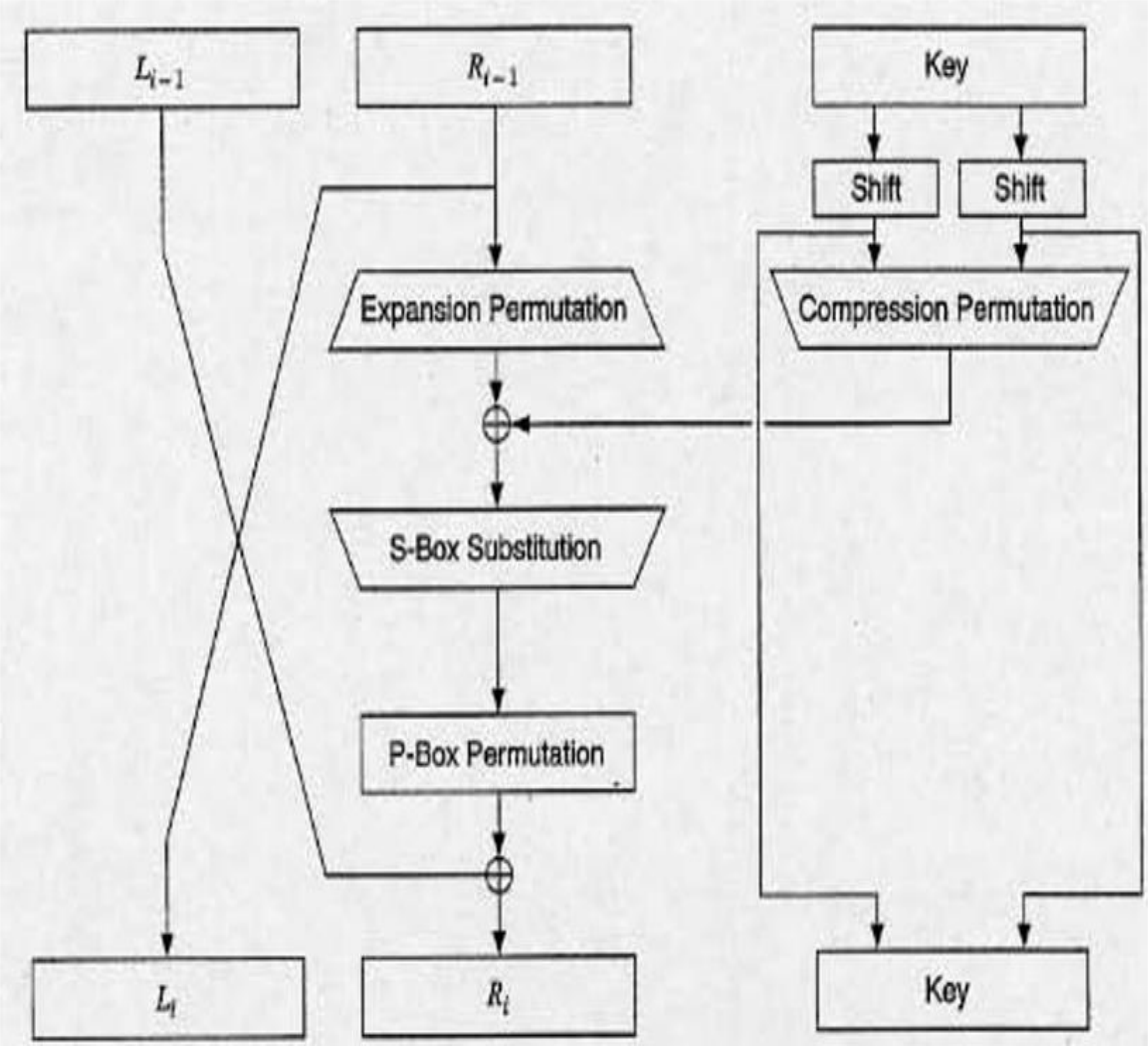
DES Encryption



DES Decryption



One Round of DES



Example 1

- PT 1 = 0x46868bd449786458
- Key 1 = 0x144573e006467894
- CT 1 = 0xae8180eb706729d3

- Key 2 = 0x144573e016467894
- CT 2 = 0xa14a01e6c590db61

Example 2

- PT 1 = 0xfedcba9876543210
- Key 1 = 0x0123456789abcdef
- CT 1 = 0x12c626af058b433b

- PT 2 = 0xfedcba9876543211
- CT 2 = 0x7b129948ca8d29d6

Strength of DES

- Number of possible keys = $7.2057 * 10^{16}$.
- Maximum time required for a PC to execute a successful DES decryption at 10^9 decryptions/second = $(7.2057 * 10^{16})/(10^9/\text{second}) = 7.2057*10^7$ seconds ≈ 2 years and 3 months.
- Cryptanalysis is possible by exploiting the characteristics of DES.
- DES is moderately resistant to a successful timing attack.

Advanced Encryption Standard (AES)

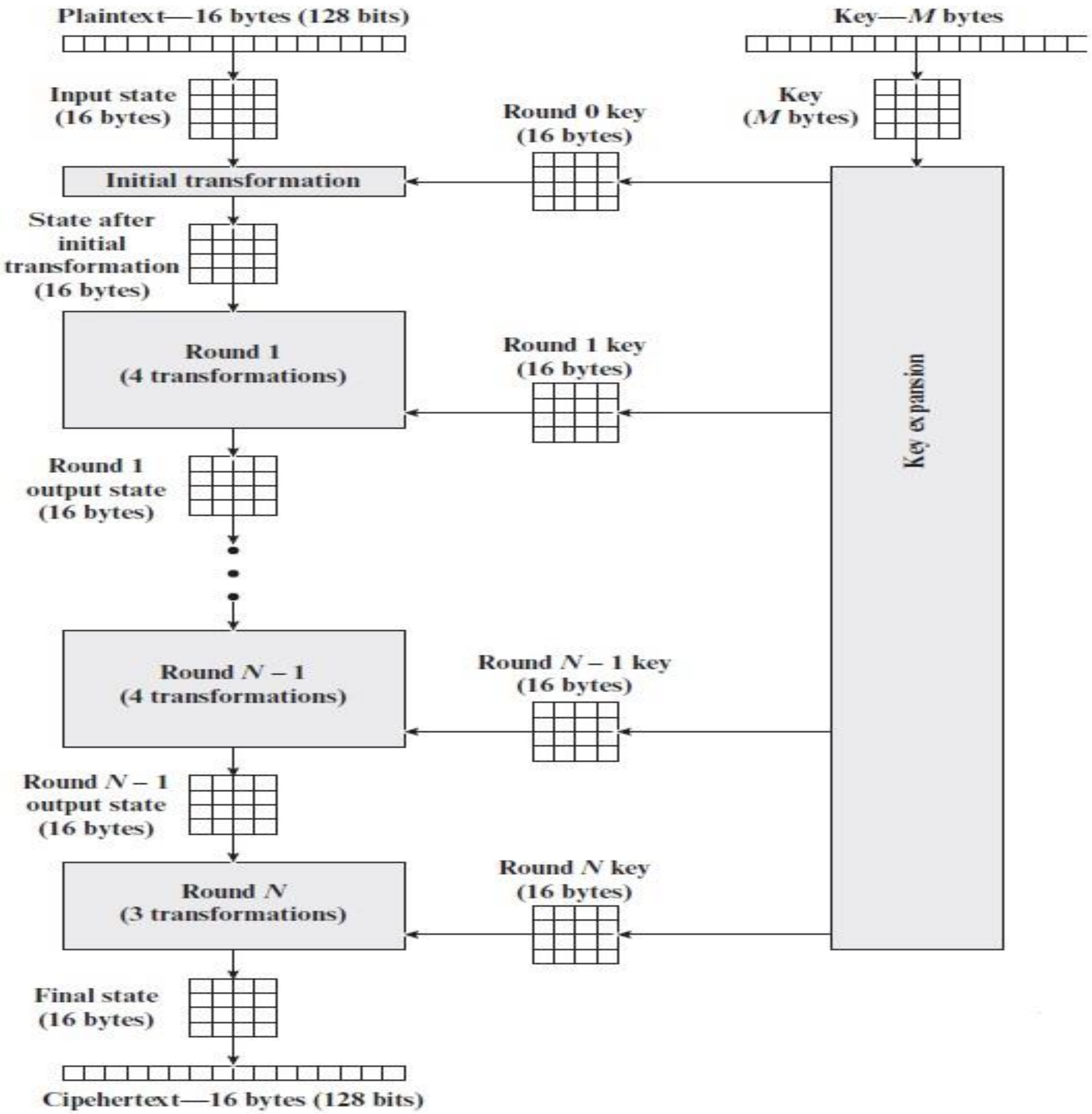
Overview of AES

- Published by NIST in 2001.
- Input Block = 128 bits
- Output Block = 128 bits
- Variants of AES:- AES-128, AES-192, and AES-256.
- AES-128 (10 rounds, 128 bits key)
- AES-192 (12 rounds, 192 bits key)
- AES-256 (14 rounds, 256 bits key)
- Round Key size = 128 bits

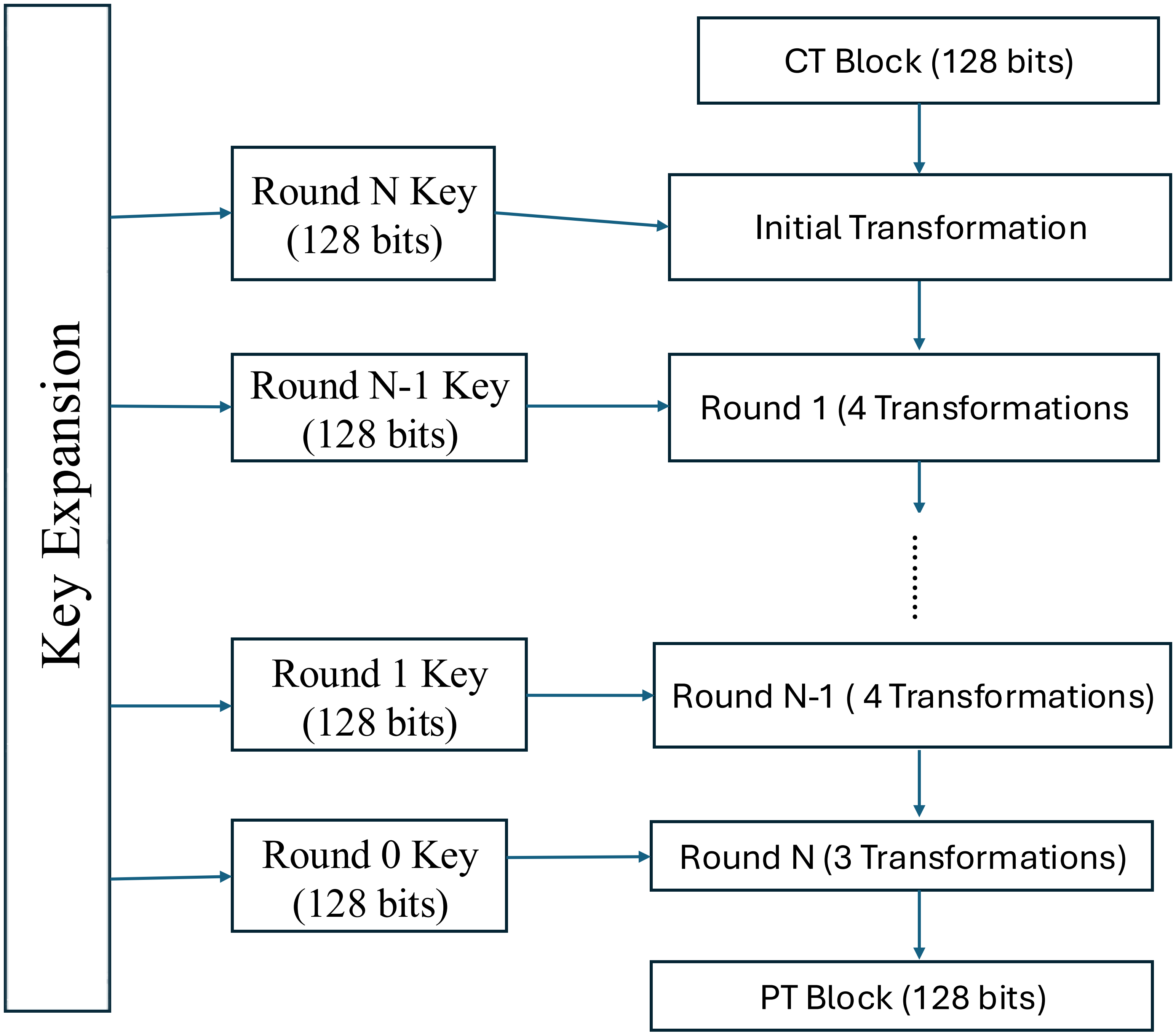
Galois Field for AES

- All the operations are performed in $GF(2^8)$.
- The irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$.
- In $GF(2^n)$, any polynomial can be represented as a n-bit value.
- For example, the binary value corresponding to the polynomial $x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x^1 + 1$ in $GF(2^8) = (11111111)_2 = 0xFF$.
- $x^6 + x^5 + x^3 + x^2 + x = 0x6E$.

General Structure of AES Encryption



General Structure of AES Decryption



General Structure of AES

- Input State Array:-

$$\begin{bmatrix} byte_0 & byte_4 & byte_8 & byte_{12} \\ byte_1 & byte_5 & byte_9 & byte_{13} \\ byte_2 & byte_6 & byte_{10} & byte_{14} \\ byte_3 & byte_7 & byte_{11} & byte_{15} \end{bmatrix}$$

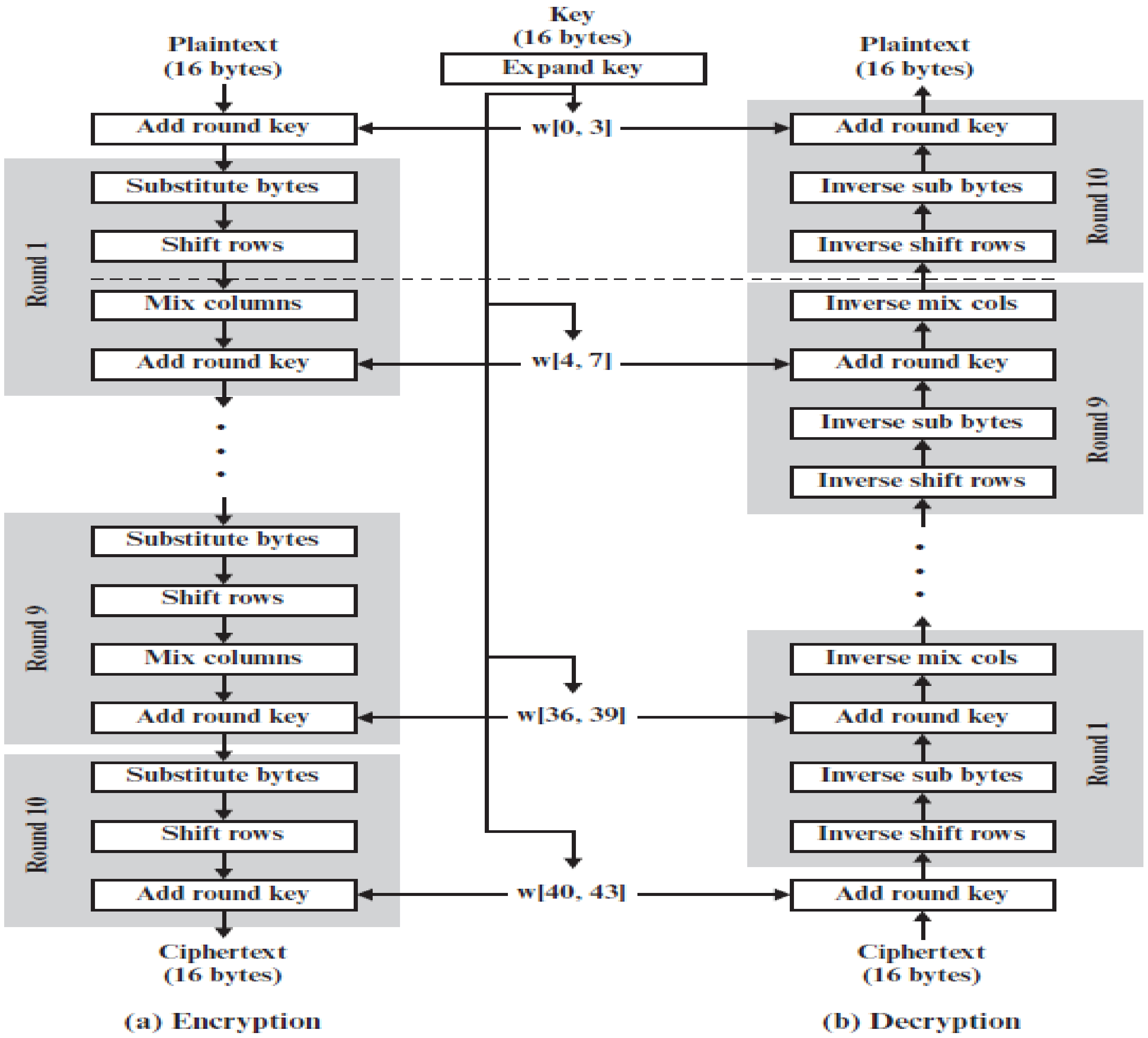
- Output State Array:-

out_0	out_4	out_8	out_{12}
out_1	out_5	out_9	out_{13}
out_2	out_6	out_{10}	out_{14}
out_3	out_7	out_{11}	out_{15}

AES Parameters

Key Size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext Block Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14
Round Key Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded Key Size (words/bytes)	44/176	52/208	60/240

Structure of AES-128



S Box

		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

Steps to Construct S Box

1) Initialize the S Box row-wise for 16 rows and 16 columns (Row and Column Indices are 0 to F), in ascending order.

2) Map each of the 256 values in Output of Step 1 with its multiplicative inverse over GF(2^8).

3) Convert each element of Output of Step 2 into its binary equivalent ($b_7b_6b_5b_4b_3b_2b_1b_0$).

4) Now calculate ($b'_7b'_6b'_5b'_4b'_3b'_2b'_1b'_0$) for each element of Output of Step 3 using the affine transformation:-

Steps to Construct S Box (Contd..)

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Steps to Construct S Box (Contd..)

5) Convert each element of Output of Step 4 into its Hexadecimal Equivalent.

Inverse S Box

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	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

Steps to Construct Inverse S Box

1) Initialize the Inverse S Box row-wise for 16 rows and 16 columns (Row and Column Indices are 0 to F), in ascending order.

2) Convert each element of Output of Step 1 into its binary equivalent ($b_7b_6b_5b_4b_3b_2b_1b_0$).

3) Now calculate ($b'_7b'_6b'_5b'_4b'_3b'_2b'_1b'_0$) for each element of Output of Step 2 using the affine transformation:-

Steps to Construct Inverse S Box (Contd..)

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Steps to Construct Inverse S Box (Contd..)

4) Calculate the Multiplicative Inverse of each element of Output of Step 3, over GF(2^8).

5) Convert each element of Output of Step 4 into its 8-bit binary equivalent, and eventually into Hexadecimal equivalent.

Proof for Inverse Affine Transformation

- The Affine Transformation in S Box construction is $B' = X*B \oplus C$.
- The Affine Transformation in Inverse S Box construction is $B = Y*B' \oplus D$.
- Now we have to prove that LHS = RHS for Inverse S Box construction
- $RHS = Y*B' \oplus D$
- $RHS = Y*(X*B \oplus C) \oplus D$
- $RHS = Y*X*B \oplus Y*C \oplus D$
- RHS =

Proof for Inverse Affine Transformation (Contd..)

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

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} =$$

Proof for Inverse Affine Transformation (Contd..)

- RHS =

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

- Therefore, RHS = B

Numerical 1

- If Input to S Box is 0x1D, what's the corresponding output?

Solution:-

Step 1:-

$0x1D = (00011101)_2 = x^4 + x^3 + x^2 + 1$

Step 2:-

Iteration 1:-

$a(x) = x^8 + x^4 + x^3 + x + 1$; $b(x) = x^4 + x^3 + x^2 + 1$;

$q(x) = x^4 + x^3 + x + 1$; $r(x) = x^2$;

$v1(x) = 0$; $v2(x) = 1$; $v(x) = x^4 + x^3 + x + 1$

Numerical 1 (Contd..)

Iteration 2:-

$a(x) = x^4 + x^3 + x^2 + 1$; $b(x) = x^2$; $q(x) = x^2 + x + 1$; $r(x) = 1$;

$v1(x) = 1$; $v2(x) = x^4 + x^3 + x + 1$; $v(x) = x^6$

Iteration 3:-

$a(x) = x^2$; $b(x) = 1$; $q(x) = x^2$; $r(x) = 0$;

$v1(x) = x^4 + x^3 + x + 1$; $v2(x) = x^6$; $v(x) = 0$;

Iteration 4:-

$a(x) = 1$; $b(x) = 0$; $v1(x) = x^6$; $v2(x) = 0$;

Numerical 1 (Contd..)

$MI(x^4 + x^3 + x^2 + 1) \bmod (x^8 + x^4 + x^3 + x + 1) = x^6$

Step 3:-

$(b_7b_6b_5b_4b_3b_2b_1b_0) = (01000000)_2$

Step 4:-

Numerical 1 (Contd..)

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Numerical 1 (Contd..)

- $= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$

- Output = *0xA4*

Numerical 2

- If Input to S Box is 0x7D, what's the corresponding output?

Solution:-

Step 1:-

$0x7D = (01111101)_2 = (x^6+x^5+x^4+x^3+x^2+1)$

Step 2:-

Iteration 1:-

$a(x) = x^8 + x^4 + x^3 + x + 1$; $b(x) = x^6 + x^5 + x^4 + x^3 + x^2 + 1$; $q(x) = x^2 + x$;

$r(x) = x^4 + x^2 + 1$; $v1(x) = 0$; $v2(x) = 1$; $v(x) = x^2 + x$

Numerical 2 (Contd..)

Iteration 2:-

$a(x) = x^6 + x^5 + x^4 + x^3 + x^2 + 1$; $b(x) = x^4 + x^2 + 1$; $q(x) = x^2 + x$; $r(x) = x + 1$;
$v1(x) = 1$; $v2(x) = x^2 + x$; $v(x) = x^4 + x^2 + 1$

Iteration 3:-

$a(x) = x^4 + x^2 + 1$; $b(x) = x + 1$; $q(x) = x^3 + x^2$; $r(x) = 1$;
$v1(x) = x^2 + x$; $v2(x) = x^4 + x^2 + 1$; $v(x) = x^7 + x^6 + x^5 + x^4 + x^3 + x$

Iteration 4:-

$a(x) = x + 1$; $b(x) = 1$; $q(x) = x + 1$; $r(x) = 0$;
$v1(x) = x^4 + x^2 + 1$; $v2(x) = x^7 + x^6 + x^5 + x^4 + x^3 + x$; $v(x) = 0$

Numerical 2 (Contd..)

Iteration 5:-

$a(x) = 1$; $b(x) = 0$; $v1(x) = x^7 + x^6 + x^5 + x^4 + x^3 + x$; $v2(x) = 0$

$MI(x^6 + x^5 + x^4 + x^3 + x^2 + 1) \bmod (x^8 + x^4 + x^3 + x + 1) = x^7 + x^6 + x^5 + x^4 + x^3 + x$

Step 3:-

$(b_7b_6b_5b_4b_3b_2b_1b_0) = (11111010)_2$

Step 4:-

Numerical 2 (Contd..)

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Numerical 2 (Contd..)

$$= \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

- Output = *0xFF*

Numerical 3

- If Input to Inverse S Box is 0xA4, what's the corresponding output?

<u>**Solution:-**</u>

<u>*Step 1:-*</u>

$(b_7b_6b_5b_4b_3b_2b_1b_0) = (10100100)_2$

<u>*Step 2:-*</u>

Numerical 3 (Contd..)

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} =$$

Numerical 3 (Contd..)

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- $(01000000)_2 = x^6$

Numerical 3 (Contd..)

Step 3:-

Iteration 1:-

$a(x) = x^8 + x^4 + x^3 + x + 1$; $b(x) = x^6$; $q(x) = x2$; $r(x) = x^4 + x^3 + x + 1$;

$v1(x) = 0$; $v2(x) = 1$; $v(x) = x^2$

Iteration 2:-

$a(x) = x^6$; $b(x) = x^4 + x^3 + x + 1$; $q(x) = x^2 + x + 1$; $r(x) = 1$;

$v1(x) = 1$; $v2(x) = x^2$; $v(x) = x^4 + x^3 + x^2 + 1$

Numerical 3 (Contd..)

Iteration 3:-

$a(x) = x^4 + x^3 + x + 1$; $b(x) = 1$; $q(x) = x^4 + x^3 + x + 1$; $r(x) = 0$;
$v1(x) = x^2$; $v2(x) = x^4 + x^3 + x^2 + 1$; $v(x) = 0$

Iteration 4:-

$a(x) = 1$; $b(x) = 0$; $v1(x) = x^4 + x^3 + x^2 + 1$

Step 4:-

- Output = $(00011101)_2$ = *0x1D*

Numerical 4

- If Input to Inverse S Box is 0x55, what's the corresponding output?

Solution:-

Step 1:-

$(b_7b_6b_5b_4b_3b_2b_1b_0) = (01010101)_2$

Step 2:-

Numerical 4 (Contd..)

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} =$$

Numerical 4 (Contd..)

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

- $(01010000)_2 = x^6 + x^4$

Numerical 4 (Contd..)

Step 3:-

Iteration 1:-

$a(x) = x^8 + x^4 + x^3 + x + 1$; $b(x) = x^6 + x^4$; $q(x) = x2 + 1$; $r(x) = x^3 + x + 1$;
$v1(x) = 0$; $v2(x) = 1$; $v(x) = x^2 + 1$

Iteration 2:-

$a(x) = x^6 + x^4$; $b(x) = x^3 + x + 1$; $q(x) = x^3 + 1$; $r(x) = x + 1$;
$v1(x) = 1$; $v2(x) = x^2 + 1$; $v(x) = x^5 + x^3 + x^2$

Iteration 3:-

Numerical 4 (Contd..)

$a(x) = x^3 + x + 1$; $b(x) = x + 1$; $q(x) = x^2 + x$; $r(x) = 1$;

$v1(x) = x^2 + 1$; $v2(x) = x^5 + x^3 + x^2$; $v(x) = x^7 + x^6 + x^5 + x^3 + x^2 + 1$

Iteration 4:-

$a(x) = x + 1$; $b(x) = 1$; $q(x) = x + 1$; $r(x) = 0$;

$v1(x) = x^5 + x^3 + x^2$; $v2(x) = x^7 + x^6 + x^5 + x^3 + x^2 + 1$; $v(x) = 0$

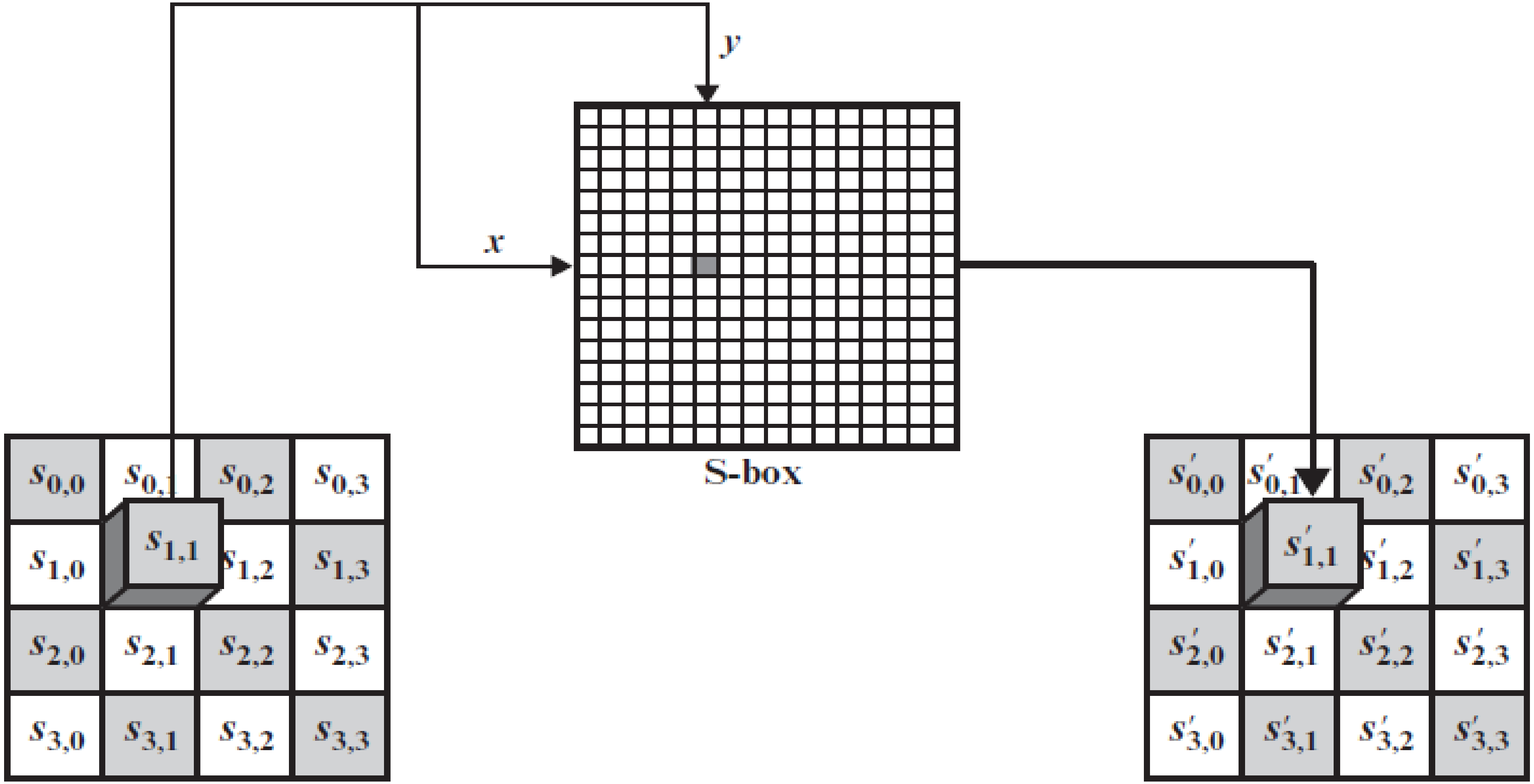
Iteration 5:-

$a(x) = 1$; $b(x) = 0$; $v1(x) = x^7 + x^6 + x^5 + x^3 + x^2 + 1$

Step 4:-

Output = $(11101101)_2$ = *0xED*

Substitute Bytes Operation



Substitute Bytes Operation (Example 1)

0x00	0x01	0x02	0x03
0x10	0x11	0x12	0x13
0x20	0x21	0x22	0x23
0x30	0x31	0x32	0x33

→

0x63	0x7C	0x77	0x7B
0xCA	0x82	0xC9	0x7D
0xB7	0xFD	0x93	0x26
0x04	0xC7	0x23	0xC3

Inverse Substitute Bytes Operation

- Operation is similar to Substitute Bytes Operation, but here Inverse S box is used instead.
- Example:-

0x00	0x01	0x02	0x03
0x10	0x11	0x12	0x13
0x20	0x21	0x22	0x23
0x30	0x31	0x32	0x33

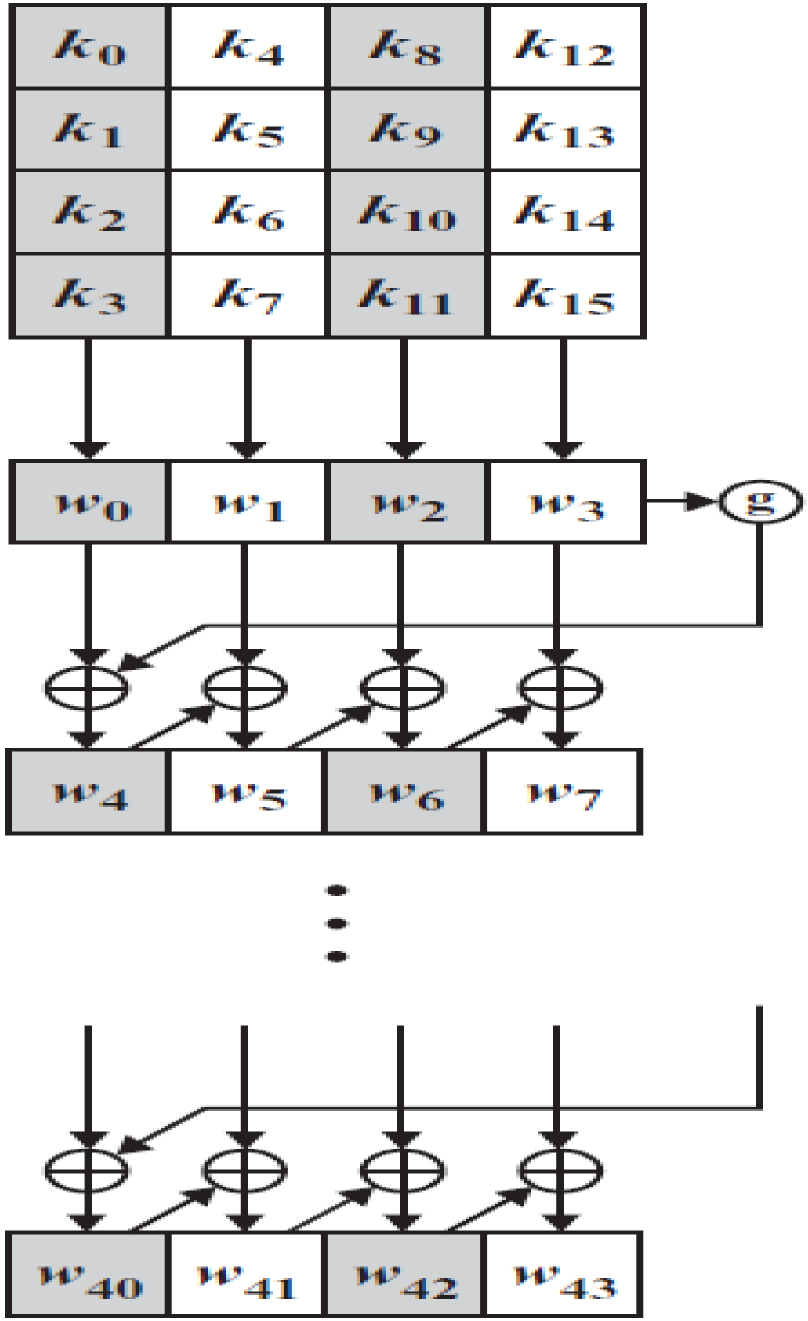
$\longrightarrow$

0x52	0x09	0x6A	0xD5
0x7C	0xE3	0x39	0x82
0x54	0x7B	0x94	0x32
0x08	0x2E	0xA1	0x66

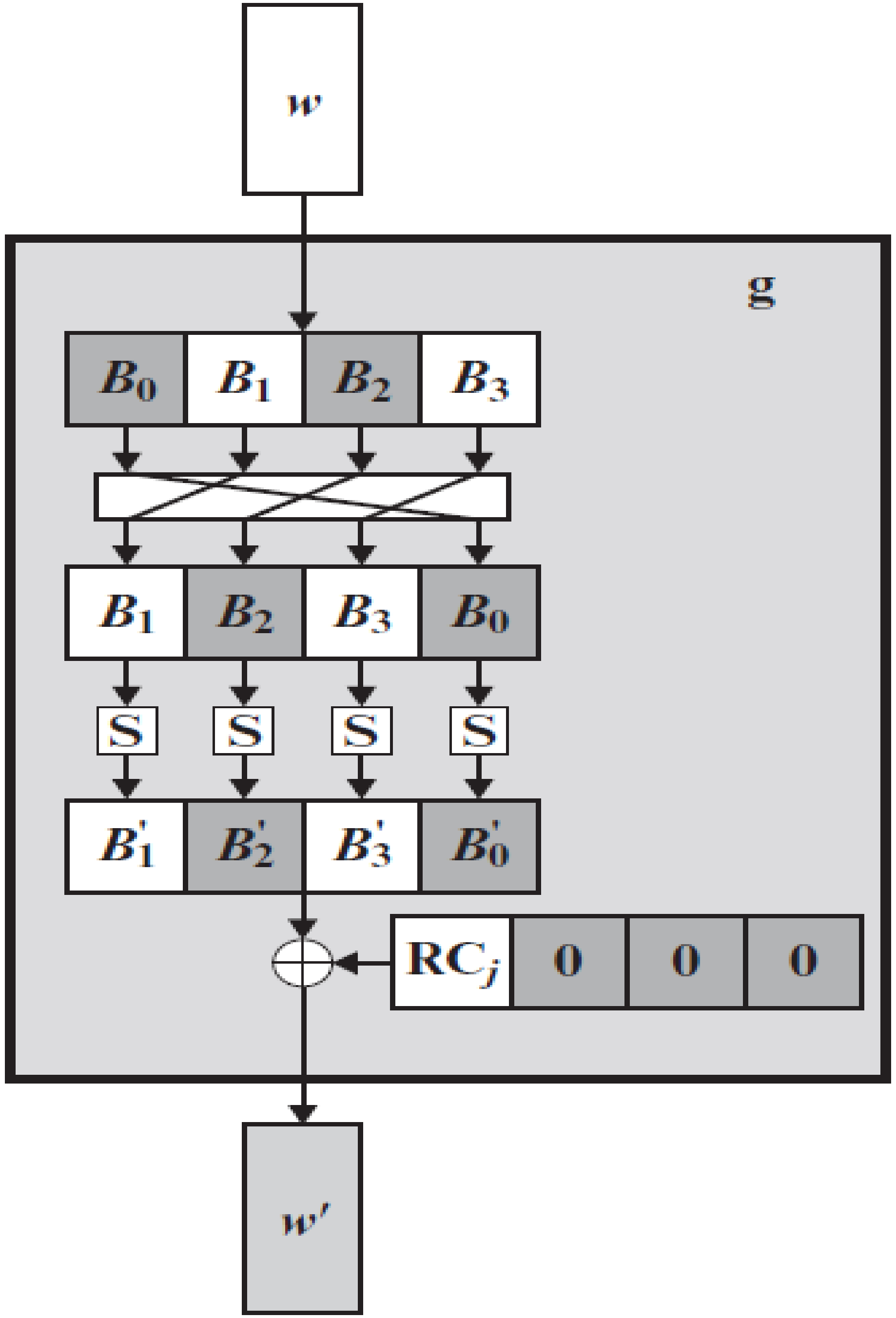
Rationale of Substitute Bytes Operation

- Rijndael developers targeted to provide low correlation between input and output bits.
- Non-Linearity of S Box is provided by Multiplicative Inverse calculations.
- Invertible property of S Box.
- S Box has non-homomorphic nature.
- Enhances Confusion

Key Expansion in AES-128



g-Function for Key Expansion in AES-128



Round Constant for Key Expansion in AES-128

- Rcon[j] = (RC[j],0,0,0)

- RC[j] = 2*RC[j-1] over GF(2^8)

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

Observations on Key Expansion in AES-128

- $w_{4*j} = w_{4*(j-1)} \oplus g(w_{4*j-1})$

- $w_{4*j+1} = w_{4*j-3} \oplus w_{4*j}$

- $w_{4*j+2} = w_{4*j-2} \oplus w_{4*j+1}$

- $w_{4*j+3} = w_{4*j-1} \oplus w_{4*j+2}$

Numerical 5

- If Key = 0x0F1571C947D9E8590CB7ADD6AF7F6798, calculate w_4 during Round Keys generation in AES-128?

Solution:-

Step 1:-

$w_0 = 0$ x 0F 15 71 C9
$w_1 = 0$ x 47 D9 E8 59
$w_2 = 0$ x 0C B7 AD D6
$w_3 = 0$ x AF 7F 67 98

Numerical 5 (Contd..)

Step 2:-

$x = \text{RotWord}(w_3) = 0 \text{ x } 7F\ 67\ 98\ AF$

$y = \text{SubWord}(x) = 0 \text{ x } D2\ 85\ 46\ 79$

$\text{Rcon}(1) = 0 \text{ x } 01\ 00\ 00\ 00$

$z = y \oplus \text{Rcon}(1) = (0 \text{ x } D2\ 85\ 46\ 79) \oplus (0 \text{ x } 01\ 00\ 00\ 00)$

$z = 0 \text{ x } D3\ 85\ 46\ 79$

Step 3:-

$w_4 = w_0 \oplus z = (0 \text{ x } 0F\ 15\ 71\ C9) \oplus (0 \text{ x } D3\ 85\ 46\ 79)$

$w_4 = 0\ x\ DC\ 90\ 37\ B0$

Numerical 6

- During Key Expansion in AES-128, the output array of Round 6 is as given in the array below. Generate the Output array of Round 7 for key expansion.

71	8C	83	CF
C7	29	E5	A5
4C	74	EF	A9
C2	EF	52	EF

Numerical 6 (Contd..)

Solution:-

71	8C	83	CF
C7	29	E5	A5
4C	74	EF	A9
C2	EF	52	EF

w_{24}	w_{25}	w_{26}	w_{27}

Numerical 6 (Contd..)

- $x = \text{RotWord}(w_{27}) = 0 \text{ x A5 A9 EF CF}$
- $y = \text{SubWord}(x) = 0 \text{ x 06 D3 DF 8A}$
- $\text{Rcon}(7) = 0 \text{ x 40 00 00 00}$
- $z = y \oplus \text{Rcon}(7) = 0 \text{ x 46 D3 DF 8A}$

- $w_{28} = w_{24} \oplus z = 0 \text{ x 37 14 93 48}$
- $w_{29} = w_{28} \oplus w_{25} = 0 \text{ x BB 3D E7 A7}$
- $w_{30} = w_{29} \oplus w_{26} = 0 \text{ x 38 D8 08 F5}$
- $w_{31} = w_{30} \oplus w_{27} = 0 \text{ x F7 7D A1 1A}$

Numerical 6 (Contd..)

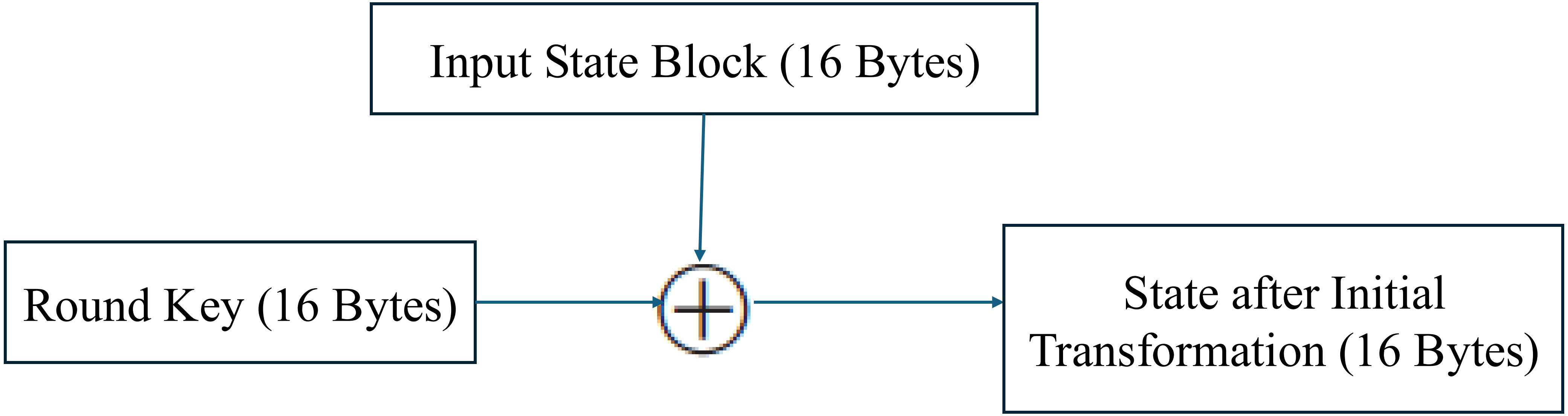
- Output array for 7th Round:-

37	BB	38	F7
14	3D	D8	7D
93	E7	08	A1
48	F7	A5	4A

Rationale of Key Expansion in AES

- Knowledge of a part of the cipher key or round key does not enable calculation of many other round-key bits.
- An invertible transformation
- Speed on a wide range of processors.
- Usage of round constants to eliminate symmetries.
- Impact of cipher key differences on the round keys.
- Enough nonlinearity to prohibit the full determination of round key differences from cipher key differences only.
- Simplicity of description.

Add Round Key Transformation in AES-128



- Rationale:- Simple operation which affects every bit of the state.

Numerical 7

- PT = 0x0123456789ABCDEFFEDCBA9876543210
- Key = 0x0F1571C947D9E8590CB7ADD6AF7F6798
- State Array after Initial Transformation in AES-128 = ?

Solution:-

- State Array after Initial Transformation = PT $\oplus$ Key =

Numerical 7 (Contd..)

01	89	FE	76
23	AB	DC	54
45	CD	BA	32
67	EF	98	10

$\oplus$

0F	47	0C	AF
15	D9	B7	7F
71	E8	AD	67
C9	59	D6	98

=

0E	CE	F2	D9
36	72	6B	2B
34	25	17	55
AE	B6	4E	88

Shift Rows Transformation in AES-128

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

$\rightarrow$

x11	x12	x13	x14
x22	x23	x24	x21
x33	x34	x31	x32
x44	x41	x42	x43

Inverse Shift Rows Transformation in AES-128

x11	x12	x13	x14
x21	x22	x23	x24
x31	x32	x33	x34
x41	x42	x43	x44

→

x11	x12	x13	x14
x24	x21	x22	x23
x33	x34	x31	x32
x42	x43	x44	x41

Numerical 8

- When the array (as shown below) is the input to Shift Rows Transformation in AES-128, what's the output array just after the transformation?

0x4A	0x7F	0x6B	0xBF
0x21	0x40	0x3A	0x3C
0x8D	0x18	0xC7	0xC9
0xB8	0x14	0xD2	0x22

Solution:-

0x4A	0x7F	0x6B	0xBF
0x40	0x3A	0x3C	0x21
0xC7	0xC9	0x8D	0x18
0x22	0xB8	0x14	0xD2

Numerical 9

- When the array (as shown below) is the input to Inverse Shift Rows Transformation in AES-128, what's the output array just after the transformation?

0x40	0xF4	0x1F	0xF2
0x6F	0x48	0x2D	0x72
0x65	0x4D	0x37	0xB7
0x2F	0x63	0x3C	0x94

Solution:-

0x40	0xF4	0x1F	0xF2
0x72	0x6F	0x48	0x2D
0x37	0xB7	0x65	0x4D
0x63	0x3C	0x94	0x2F

Rationale of Shift Rows Transformation

- Enhances Diffusion
- Scatters Bytes across various parts of the output.

Mix Columns Transformation in AES-128

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

Mix Columns Transformation in AES-128 (Contd..)

$$s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j}$$

$$s'_{1,j} = s_{0,j} \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j}$$

$$s'_{2,j} = s_{0,j} \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j})$$

$$s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j})$$

- All the operations are performed in $GF(2^8)$.

Inverse Mix Columns Transformation in AES-128

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Proof that Mix Column Operation is Invertible

- We are supposed to prove that:-
- Y =

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{0,3} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

Proof that Mix Column Operation is Invertible

Assume that X =

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

Proof that Mix Column Operation is Invertible (Contd..)

- $X_{r1c1} = (0x0E) * (0x02) \oplus (0x0B) \oplus (0x0D) \oplus (0x09)*(0x03)$
- $X_{r1c1} = (x^3 + x^2 + x)*(x) \oplus (x^3 + x + 1) \oplus (x^3 + x^2 + 1) \oplus (x^3 + 1)*(x+1)$
- $X_{r1c1} = (11100)_2 \oplus (1011)_2 \oplus (1101)_2 \oplus (11011)_2 = 0x01$

- $X_{r2c1} = (0x09) * (0x02) \oplus (0x0E) \oplus (0x0B\} \oplus (0x0D)*(03)$
- $X_{r2c1} = (x^3 + 1)*(x) \oplus (x^3 + x^2 + x) \oplus (x^3 + x + 1) \oplus (x^3 + x^2 + 1)\ (x+1)$
- $X_{r2c1} = (10010)_2 \oplus (1110)_2 \oplus (01011)_2 \oplus (10111)_2 = 0x00$

- $X_{r3c1} = (0x0D)*(0x02) \oplus (0x09) \oplus (0x0E) \oplus (0x0B)*(0x03)$
- $X_{r3c1} = (x^3 + x^2 + 1)*(x) \oplus (x^3 + 1) \oplus (x^3 + x^2 + x) \oplus (x^3 + x + 1)*(x+1)$
- $X_{r3c1} = (11010)_2 \oplus (1001)_2 \oplus (1110)_2 \oplus (11101)_2 = 0x00$

Proof that Mix Column Operation is Invertible (Contd..)

- $X_{r4c1} = (0x0B)*(0x02) \oplus (0x0D) \oplus (0x09) \oplus (0x0E)*(0x03)$
- $X_{r4c1} = (x^3 + x + 1)*(x) \oplus (x^3 + x^2 + 1) \oplus (x^3 + 1) \oplus (x^3 + x^2 + x)*(x+1)$
- $X_{r4c1} = (10110)_2 \oplus (1101)_2 \oplus (1001)_2 \oplus (10010)_2 = 0x00$

- Similarly, we can obtain other values of Y.
- Y =

$$\begin{bmatrix} 0x01 & 0x00 & 0x00 & 0x00 \\ 0x00 & 0x01 & 0x00 & 0x00 \\ 0x00 & 0x00 & 0x01 & 0x00 \\ 0x00 & 0x00 & 0x00 & 0x01 \end{bmatrix} * \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{0,3} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

Rationale of Mix Column Transformation

- Though a Linear Transformation, the operations enhance the overall security.
- Enhances Diffusion

Numerical 10

- If the input to Mix Column Transformation in AES-128 is as shown in the array below, then what's the output of the transformation in the 1st row 1st column?

0x87	0xF2	0x4D	0x97
0x6E	0x4C	0x90	0xEC
0x46	0xE7	0x4A	0xC3
0xA6	0x8C	0xD8	0x95

Solution:-

- $m(x) = x^8 + x^4 + x^3 + x + 1$
- Output = (0x02) * (0x87) ⊕ (0x03) * (0x6E) ⊕ (0x46) ⊕ (0xA6)

Numerical 10 (Contd..)

- $(0x02) * (0x87) = x * (x^7 + x^2 + x + 1) \bmod m(x)$
- $(0x02) * (0x87) = (x^8 + x^3 + x^2 + x) \bmod m(x)$
- $(0x02) * (0x87) = (x^4 + x^2 + 1) = (00010101)_2$

- $(0x03) * (0x6E) = (x + 1)\,(x^6 + x^5 + x^3 + x^2 + x) \bmod m(x)$
- $(0x03) * (0x6E) = (x^7 + x^5 + x^4 + x) = (10110010)_2$

- Output $= (00010101)_2 \oplus (10110010)_2 \oplus (01000110)_2 \oplus (10100110)_2$
- Output $= (01000111)_2 = \mathit{0x47}$

Numerical 11

- If the input to Mix Column Transformation in AES-128 is as shown in the array below, then what's the output of the transformation in the 4^{th} row 4^{th} column?

0x87	0xF2	0x4D	0x97
0x6E	0x4C	0x90	0xEC
0x46	0xE7	0x4A	0xC3
0xA6	0x8C	0xD8	0x95

Solution:-

- $m(x) = x^8 + x^4 + x^3 + x + 1$
- Output = $(0x03)*(0x97) \oplus (0xEC) \oplus (0xC3) \oplus (0x02) * (0x95)$

Numerical 11 (Contd..)

- $(0x03) * (0x97) = (x + 1) * (x^7 + x^4 + x^2 + x + 1) \bmod m(x)$
- $(0x03) * (0x97) = (x^8 + x^7 + x^5 + x^4 + x^3 + 1) \bmod m(x)$
- $(0x03) * (0x97) = (x^7 + x^5 + x) = (10100010)_2$

- $(0x02) * (0x95) = x * (x^7 + x^4 + x^2 + 1) \bmod m(x)$
- $(0x02) * (0x95) = (x^8 + x^5 + x^3 + x) \bmod m(x)$
- $(0x02) * (0x95) = (x^5 + x^4 + 1) = (00110001)_2$

- Output $= (10100010)_2 \oplus (11101100)_2 \oplus (11000011)_2 \oplus (00110001)_2$
- Output $= (10111100)_2 = \mathit{0xBC}$

Numerical 12

- If the input to Inverse Mix Column Transformation in AES-128 is as shown in the array below, then what's the output of the transformation in the 1st row 1st column?

0x47	0x40	0xA3	0x4C
0x37	0xD4	0x70	0x9F
0x94	0xE4	0x3A	0x42
0xED	0xA5	0xA6	0xBC

Solution:-

- $m(x) = x^8 + x^4 + x^3 + x + 1$

Numerical 12 (Contd..)

- Output = (0x0E) * (0x47) $\oplus$ (0x0B) * (0x37) $\oplus$ (0x0D) * (0x94) $\oplus$ (0x09) * (0xED)

- (0x0E) * (0x47) = $(x^3 + x^2 + x)(x^6 + x^2 + x + 1)$ mod m(x)
- (0x0E) * (0x47) = $(x^9 + x^8 + x^7 + x^5 + x^3 + x)$ mod m(x)
- (0x0E) * (0x47) = $(x^7 + x^2 + x + 1) = (10000111)_2$

- (0x0B) * (0x37) = $(x^3 + x + 1) * (x^5 + x^4 + x^2 + x + 1)$ mod m(x)
- (0x0B) * (0x37) = $(x^8 + x^7 + x^6 + x^5 + 1)$ mod m(x)
- (0x0B) * (0x37) = $(x^7 + x^6 + x^5 + x^4 + x^3 + x)$ mod m(x) = $(11111010)_2$

Numerical 12 (Contd..)

- (0x0D) * (0x94) = $(x^3 + x^2 + 1)(x^7 + x^4 + x^2) \bmod m(x)$
- (0x0D) * (0x94) = $(x^{10} + x^9 + x^6 + x^5 + x^2) \bmod m(x)$
- (0x0D) * (0x94) = $(x^5 + x^4 + x^3 + x^2 + x) = (00111110)_2$

- (0x09) * (0xED) = $(x^3 + 1)(x^7 + x^6 + x^5 + x^3 + x^2 + 1) \bmod m(x)$
- (0x09) * (0xED) = $(x^{10} + x^9 + x^8 + x^2 + 1) \bmod m(x)$
- (0x09) * (0xED) = $(x^7 + x^6 + x^2) = (11000100)_2$

- Output = $(10000111)_2 \oplus (11111010)_2 \oplus (00111110)_2 \oplus (11000100)_2 =$
- Output = $(10000111)_2$ = *0x87*

Numerical 13

- If the input to Inverse Mix Column Transformation in AES-128 is as shown in the array below, then what's the output of the transformation in the 1st row 1st column?

0xB9	0x94	0x57	0x75
0xE4	0x8E	0x16	0x51
0x47	0x20	0x9A	0x3F
0xC5	0xD6	0xF5	0x3B

Solution:-

- $m(x) = x^8 + x^4 + x^3 + x + 1$

Numerical 13 (Contd..)

- Output = (0x0E) * (0xB9) ⊕ (0x0B) * (0xE4) ⊕ (0x0D) * (0x47) ⊕ (0x09) * (0xC5)

- (0x0E) * (0xB9) = $(x^3 + x^2 + x)(x^7 + x^5 + x^4 + x^3 + 1) \bmod m(x)$
- (0x0E) * (0xB9) = $(x^{10} + x^9 + x^6 + x^4 + x^3 + x^2 + x) \bmod m(x)$
- (0x0E) * (0xB9) = $x^2 = (000000100)_2$

- (0x0B) * (0xE4) = $(x^3 + x + 1) * (x^7 + x^6 + x^5 + x^2) \bmod m(x)$
- (0x0B) * (0xE4) = $(x^{10} + x^9 + x^3 + x^2) \bmod m(x)$
- (0x0B) * (0xE4) = $(x^6 + x^4 + x^2 + x) \bmod m(x) = (01010110)_2$

Numerical 13 (Contd..)

- $(0x0D) * (0x47) = (x^3 + x^2 + 1)(x^6 + x^2 + x + 1) \bmod m(x)$
- $(0x0D) * (0x47) = (x^9 + x^8 + x^6 + x^5 + x + 1) \bmod m(x)$
- $(0x0D) * (0x47) = (x^6 + x^3 + x^2 + x) = (01001110)_2$

- $(0x09) * (0xC5) = (x^3 + 1)(x^7 + x^6 + x^2 + 1) \bmod m(x)$
- $(0x09) * (0xC5) = (x^{10} + x^9 + x^7 + x^6 + x^5 + x^3 + x^2 + 1) \bmod m(x)$
- $(0x09) * (0xC5) = (x^7 + x^5 + x^4 + x^2 + x + 1) = (10110111)_2$

- Output $= (00000100)_2 \oplus (01010110)_2 \oplus (01001110)_2 \oplus (10110111)_2 =$
- Output $= (10101011)_2 = \mathit{0xAB}$

EQUIVALENT INVERSE CIPHER IN AES

Equivalent Inverse Cipher in AES

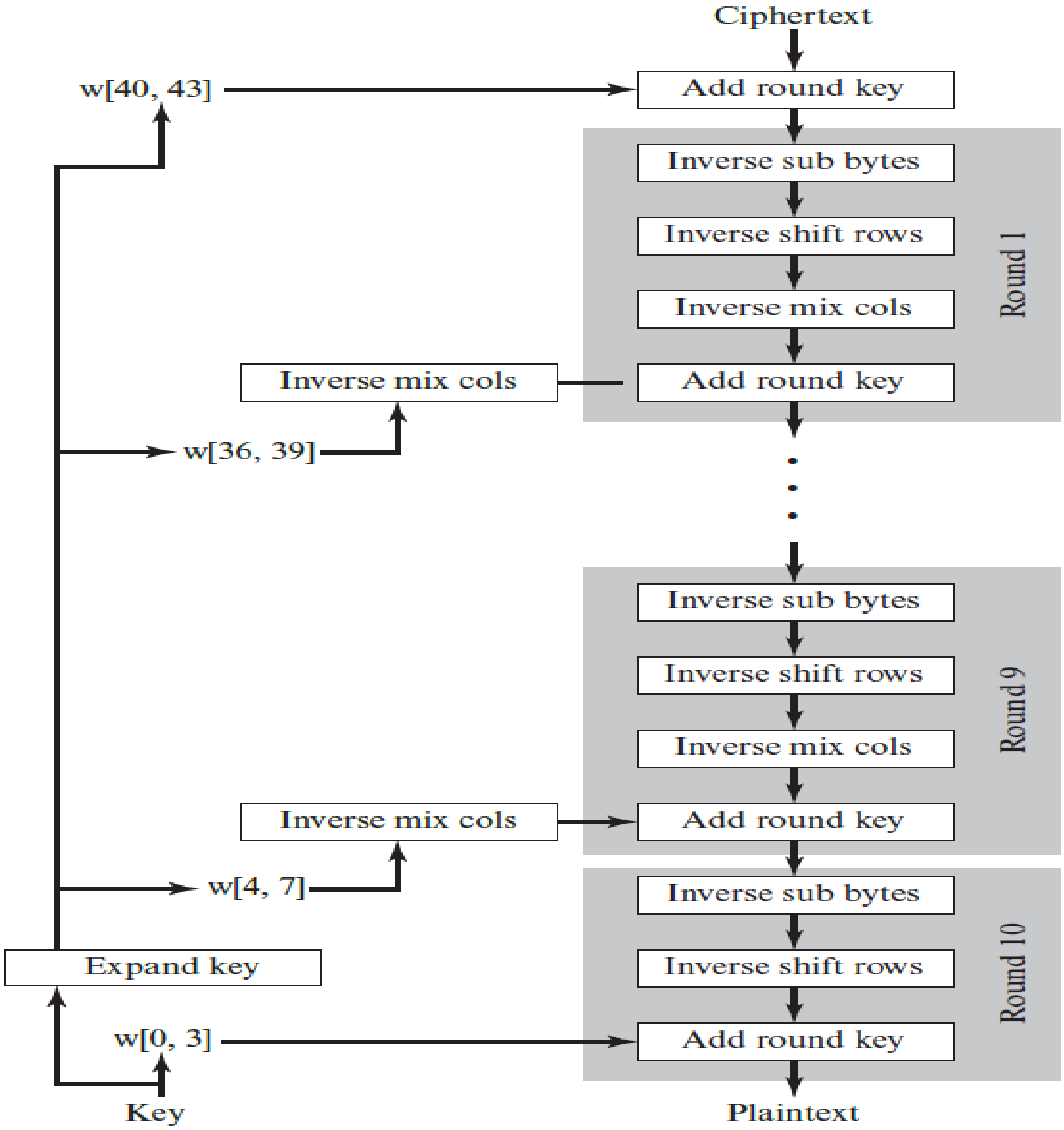
- When AES encryption and decryption algorithms are used, 2 separate software or firmware modules are required for the applications.

- It’s necessary to make 2 separate changes to make the decryption algorithm align with the encryption algorithm.

- The Inverse Shift Rows and Inverse Sub Bytes can be interchanged.

- The Add Round Key and Inverse Mix Columns can be interchanged.

Equivalent Inverse Cipher in AES (Contd..)



Interchanging Inverse Shift Rows and Inverse Sub Bytes

- Inverse Shift Rows transformation affects the sequence of Bytes in State array, without altering the Bytes contents.

- Inverse Sub Bytes affects the Bytes contents in State array, without altering the sequence of Bytes.

- Inverse Shift Rows [Inverse Sub Bytes (S_i)] = Inverse Sub Bytes [Inverse Shift Rows (S_i)]

Interchanging Inverse Shift Rows and Inverse Sub Bytes {ISB(ISR(S_i) Example}

0x00	0x01	0x02	0x03
0x10	0x11	0x12	0x13
0x20	0x21	0x22	0x23
0x30	0x31	0x32	0x33

→

0x00	0x01	0x02	0x03
0x13	0x10	0x11	0x12
0x22	0x23	0x20	0x21
0x31	0x32	0x33	0x30

↓

0x52	0x09	0x6A	0xD5
0x82	0x7C	0xE3	0x39
0x94	0x32	0x54	0x7B
0x2E	0xA1	0x66	0x08

Interchanging Inverse Shift Rows and Inverse Sub Bytes {ISR(ISB(S_i) Example}

0x00	0x01	0x02	0x03
0x10	0x11	0x12	0x13
0x20	0x21	0x22	0x23
0x30	0x31	0x32	0x33

0x52	0x09	0x6A	0xD5
0x7C	0xE3	0x39	0x82
0x54	0x7B	0x94	0x32
0x08	0x2E	0xA1	0x66

0x52	0x09	0x6A	0xD5
0x82	0x7C	0xE3	0x39
0x94	0x32	0x54	0x7B
0x2E	0xA1	0x66	0x08

Interchanging Add Round Key and Inverse Mix Columns

- The transformations do not alter the sequence of Bytes.
- The transformations are linear with respect to column input.
- Inverse Mix Columns ($S_i \oplus R_i$) = Inverse Mix Columns (S_i) $\oplus$ Inverse Mix Columns(R_i)

Interchanging Add Round Key and Inverse Mix Columns (Contd..)

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} y_0 \oplus k_0 \\ y_1 \oplus k_1 \\ y_2 \oplus k_2 \\ y_3 \oplus k_3 \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \oplus \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \end{bmatrix}$$

i.e.

$$\begin{aligned} &[\{0E\} \cdot (y_0 \oplus k_0)] \oplus [\{0B\} \cdot (y_1 \oplus k_1)] \oplus [\{0D\} \cdot (y_2 \oplus k_2)] \oplus [\{09\} \cdot (y_3 \oplus k_3) \\ &= [\{0E\} \cdot y_0] \oplus [\{0B\} \cdot y_1] \oplus [\{0D\} \cdot y_2] \oplus [\{09\} \cdot y_3] \oplus \\ &\quad [\{0E\} \cdot k_0] \oplus [\{0B\} \cdot k_1] \oplus [\{0D\} \cdot k_2] \oplus [\{09\} \cdot k_3] \end{aligned}$$