


Module 2

IDEA: Primitive operations- Key expansions- One round, Odd round, Even Round- Inverse keys for decryption. AES: Basic Structure- Primitive operation- Inverse Cipher- Key Expansion, Rounds, Inverse Rounds. Stream Cipher –RC4.

International DATA encryption algorithm (Idea)

- The Simplified **International Data Encryption Algorithm (IDEA)** is a **symmetric key block cypher** that:
- uses a fixed-length plaintext of **16 bits** and
- encrypts them in **4 chunks of 4 bits** each
- to produce **16 bits ciphertext**.
- The length of the key used is **32 bits**.
- The key is also divided into 8 blocks of 4 bits each.

Primitive Operations

- This algorithm involves a series of 4 identical complete rounds and 1 half-round. Each complete round involves a series of 14 steps that includes operations like:
- IDEA uses three operations
 - Bit-by-bit exclusive OR \oplus
 - Addition of integers modulo 2^n 
 - Multiplication of integers modulo $2^n + 1$ \odot or \otimes

Primitive Operations

- Addition of integers modulo 2^n



Assume X, Y are 2 bit no's, then

$$X \oplus Y = \boxed{X} + Y \bmod 2^2 = X + Y \bmod 4$$

$$3 \oplus 1 = \boxed{3} + 1 \bmod 4 = 4 \bmod 4 = 0$$

$$1 \oplus 1 = \boxed{0} \oplus 0 = 0$$

\oplus Function Used in IDEA
(for operand length of 2 bits)

X		Y		$X \oplus Y$	
0	00	0	00	0	00
0	00	1	01	1	01
0	00	2	10	2	10
0	00	3	11	3	11
1	01	0	00	1	01
1	01	1	01	2	10
1	01	2	10	3	11
1	01	3	11	0	00
2	10	0	00	2	10
2	10	1	01	3	11
2	10	2	10	0	00
2	10	3	11	1	01
3	11	0	00	3	11
3	11	1	01	0	00
3	11	2	10	1	01
3	11	3	11	2	10

Primitive Operations

- Multiplication of integers modulo $2^n + 1$ \odot or \otimes

Assume X, Y are 2 bit no's, then

$$X \odot Y = X^x Y \bmod 2^2 + 1 = X^x Y \bmod 5$$

$$3 \odot 1 = 3^x 1 \bmod 5 = 3 \bmod 5 = 3$$

$$11 \odot 01 = 11$$

Primitive Operations

- Multiplication of integers modulo $2^n + 1$ \odot or \otimes

If an operand is 0 then it is assumed as 2^n

$$X \odot Y = X^x Y \bmod 2^2 + 1 = X^x Y \bmod 5$$

$$3 \odot 4 = 3^x 4 \bmod 5 = 12 \bmod 5 = 2$$

$$11 \odot 00 = 10$$

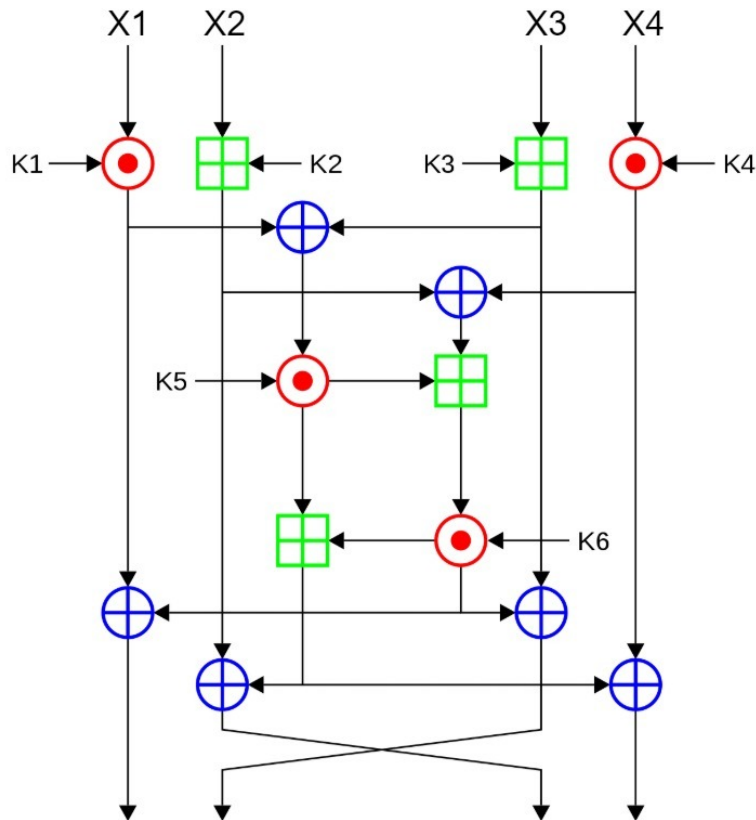
\odot Function Used in IDEA
(for operand length of 2 bits)

X		Y		$X \odot Y$	
0	00	0	00	1	01
0	00	1	01	0	00
0	00	2	10	3	11
0	00	3	11	2	10
1	01	0	00	0	00
1	01	1	01	1	01
1	01	2	10	2	10
1	01	3	11	3	11
2	10	0	00	3	11
2	10	1	01	2	10
2	10	2	10	0	00
2	10	3	11	1	01
3	11	0	00	2	10
3	11	1	01	3	11
3	11	2	10	1	01
3	11	3	11	0	00

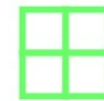
Functions Used in IDEA (for operand length of 2 bits)

X		Y		$X \oplus Y$		$X \odot Y$		$X \oplus Y$	
0	00	0	00	0	00	1	01	0	00
0	00	1	01	1	01	0	00	1	01
0	00	2	10	2	10	3	11	2	10
0	00	3	11	3	11	2	10	3	11
1	01	0	00	1	01	0	00	1	01
1	01	1	01	2	10	1	01	0	00
1	01	2	10	3	11	2	10	3	11
1	01	3	11	0	00	3	11	2	10
2	10	0	00	2	10	3	11	2	10
2	10	1	01	3	11	2	10	3	11
2	10	2	10	0	00	0	00	0	00
2	10	3	11	1	01	1	01	1	01
3	11	0	00	3	11	2	10	3	11
3	11	1	01	0	00	3	11	2	10
3	11	2	10	1	01	1	01	1	01
3	11	3	11	2	10	0	00	0	00

International Data Encryption Algorithm(IDEA)



Where,



= Modular Addition



= Modular Multiplication



= BitwiseXOR



Primitive Operations in IDEA

- Each primitive operations in IDEA maps two 16 bit quantities into a 16-bit quantity
[In DES, s-box maps 6 bit quantity into a 4 bit quantity]
- IDEA uses three operations
 - Bit-by-bit exclusive OR, \oplus
 - Addition of integers modulo 2^{16} (modulo 65536), \boxplus
 - Multiplication of integers modulo $2^{16} + 1$ (modulo 65537),

\odot or \otimes

Primitive Operations in IDEA

- The operations are reversible only if one of the inputs is known.

i.e for, $A \text{ op } B = C$

We can find A, only if B and C is known.

IDEA

- Developed by Xuejia Lai and James L
- Symmetric Block cipher: 64 bit plaintext generate 64 bit cipher
- Uses 128 bit key
- Encryption and decryption keys are related in a complex manner

- The mentioned algorithm works on **64-bit plain text** and cipher text block at one time in the algorithm.
- Then, For encryption, the **64-bit plain text is divided into four 16 bits sub-blocks of the algorithm**. In our discussion, we denote these four blocks as P1 (16 bits), P2 (16 bits), P3 (16 bits) and P4 (16 bits) as the divide of 4 blocks. Here, each of these blocks goes through 8 ROUNDS and one OUTPUT TRANSFORMATION phase at the end of the operation.
- In each of these eight rounds, some as arithmetic and logical operations are performed by this algorithm. Therefore, the eight ROUNDS are the same sequences of operations are repeated after every round in the cryptography



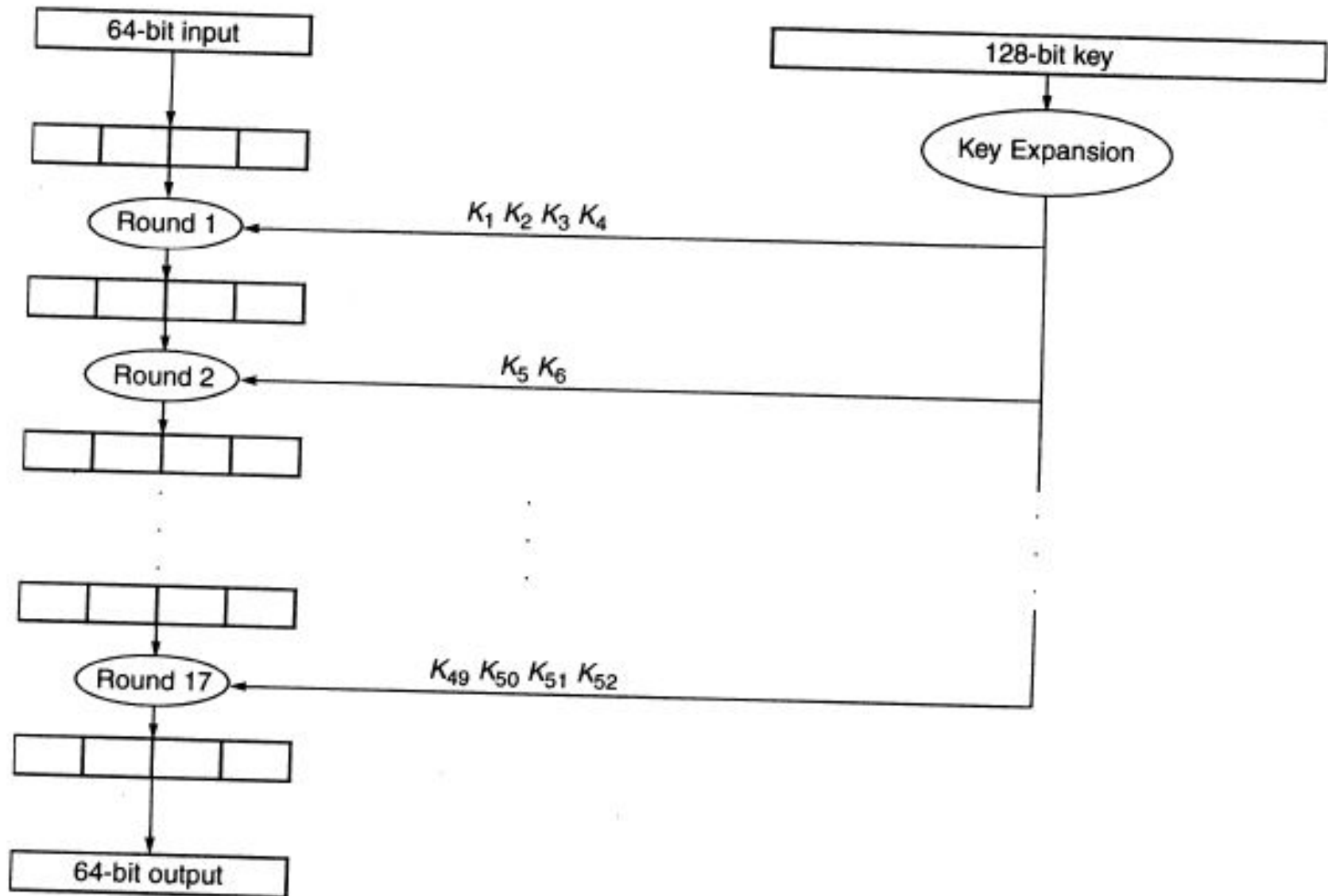


Figure 3-18. Basic Structure of IDEA

- The plaintext of 64-bit input block-divided into 4 part (16 bits each) Declare p1 to p4:
- Therefore, from plaintext p1 to p4 will be the inputs for the initial round of the algorithm in this.
- Here, are 8 such rounds.
- Then, the key is made up of 128 bits.
- In every round, 6 sub-keys will be produced by key generation.
- Each one of the sub-keys includes 16 bits of character.
- All these sub-keys will be put on the 4 input blocks p1 to p4 in the operation.
- There will be last actions include Output Transformation which usually benefits simply 4 sub-Keys at the last key generation.

Key Expansion

- The 128-bit key is expanded into 52 (16-bit) keys
- The key expansion is done differently for encryption than for decryption
- Once the 52 keys are generated the encryption and decryption is the same

Key Expansion

- The first eight sub-keys are generated directly from the 128-bit key.
 - Starting from MSB, **chopping 16 bits at a time**

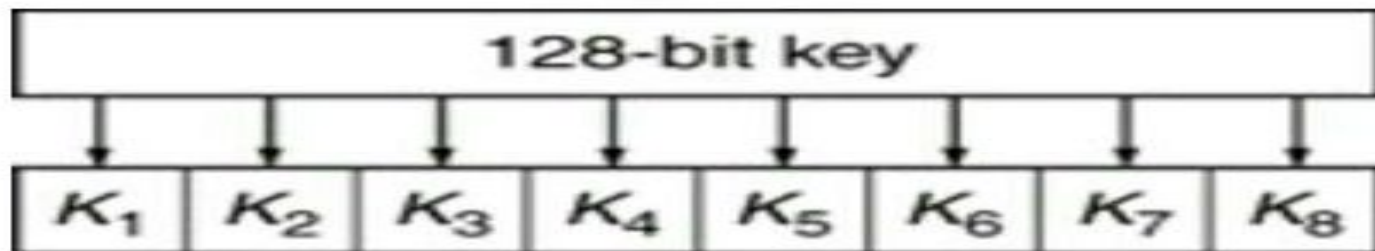


Figure 3-19. Generation of keys 1 through 8

Key Expansion

- Then circular left shift of 25 bit positions is applied to the key, and the next eight sub-keys are extracted
- This procedure is repeated until all 52 sub-keys are generated.

One Round

- Like DES, IDEA is performed in rounds
- It has 17 rounds, where odd rounds are different from even rounds
- Each round takes 64-bit input and treat it as four 16-bit quantities
- Odd round uses four of the K_i and even rounds uses two of the K_i

Odd Round

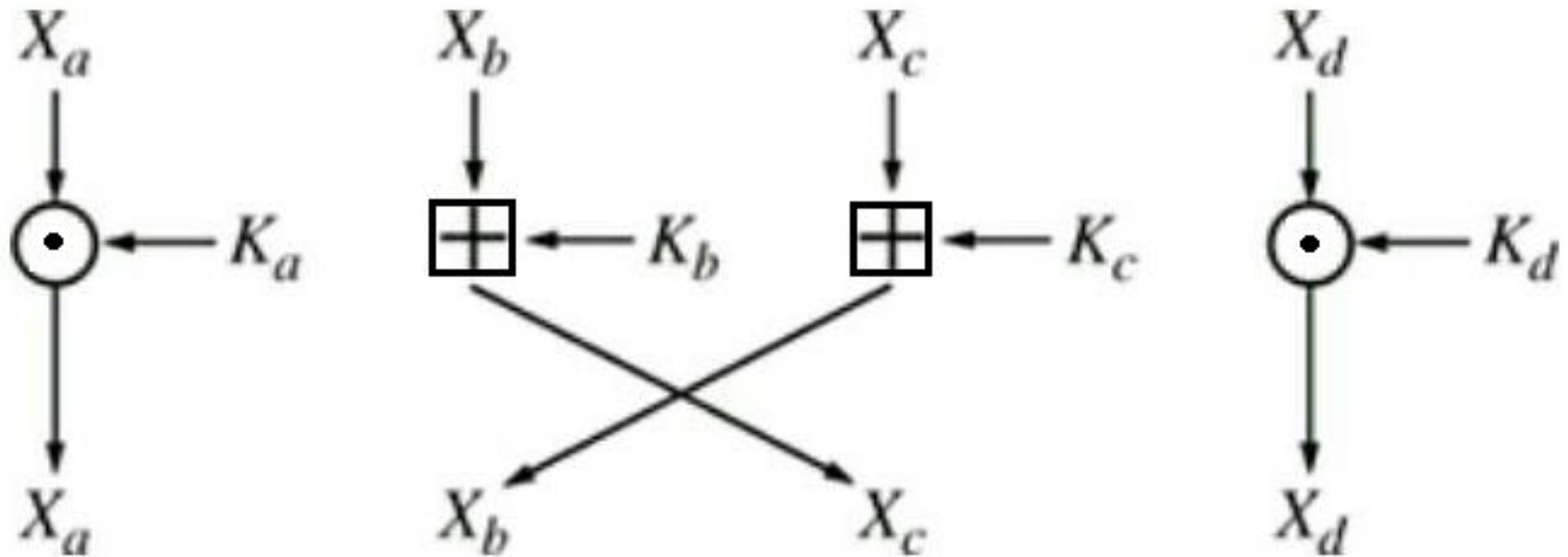
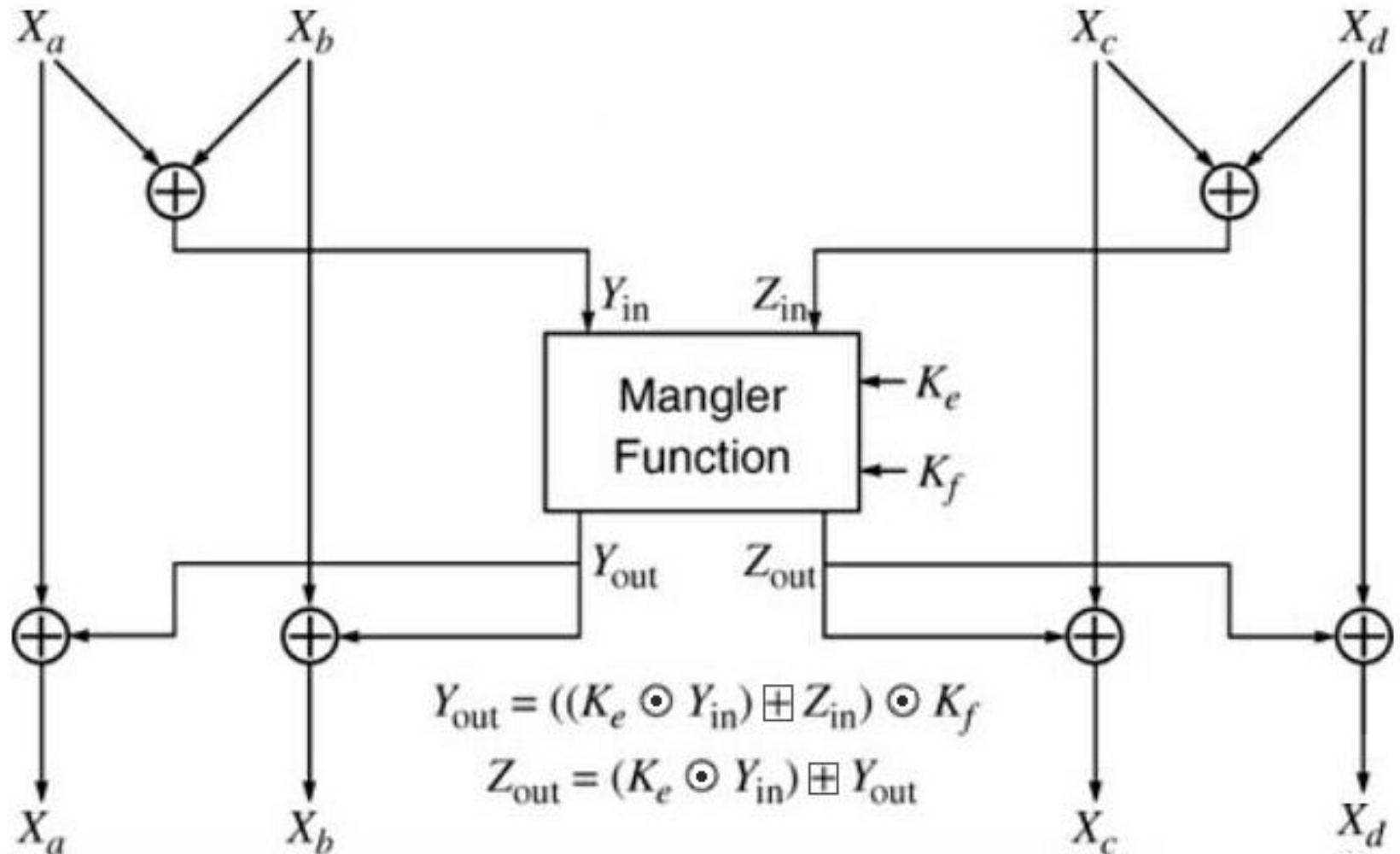


Figure 3-21. IDEA Odd Round

Even Round



Inverse Keys for Decryption

- Note that:

- $A \oplus B = C$, then $C \oplus B = A$

- $A \boxplus B = C$, then $C \boxminus B = A$

- $A \odot B = C$, then $C \odot B^{-1} = A$

Inverse Keys for Decryption

- Remember that for encryption we use 52 sub-keys, four in odd rounds and two in even rounds
- Decryption work backwards
- Hence first four sub-keys during decryption K_{1-4} will be inverses of the encryption sub-keys K_{49-52}
 - $DK_1 = EK_{49}^{-1}$, $DK_4 = EK_{52}^{-1}$
 - $DK_2 = -EK_{51}$, $DK_3 = -EK_{50}$

Inverse Keys for Decryption

- In even rounds, the keys do not have to be inverted. The same keys are used for encryption and decryption.
- i.e. $DK_5 = EK_{47}$ and $DK_6 = EK_{48}$

