Unit – III

Block Cipher –

A block cipher is a symmetric encryption method that encrypts data in fixed-size blocks, such as 64 or 128 bits, using a secret key. It contrasts with a stream cipher, which encrypts data one bit at a time. The same key is used for both encryption and decryption, and the algorithm is deterministic, meaning the same input will always produce the same output for a given key. Popular examples include AES and DES.
Key features of block ciphers

- **Deterministic:**

  For a given key and input block, the output ciphertext block is always the same.

- • **Block-based processing:**
Data is divided into fixed-size blocks before being processed by the algorithm.
- • **Symmetric key:**
The same secret key is used for both encryption and decryption.
- • **Invertible:**
The encryption process can be reversed to decrypt the ciphertext back into its original plaintext.
- • **Padding:**
If the final block of data is shorter than the block size, padding is used to fill it to the required length.
- • **Security:**
The strength of a block cipher depends on the key length, not the block size. A higher number of rounds in the algorithm also contributes to greater security.
- • **Mode of operation:**
To securely encrypt multiple blocks of data, various modes of operation are used to build upon the block cipher construction.

-

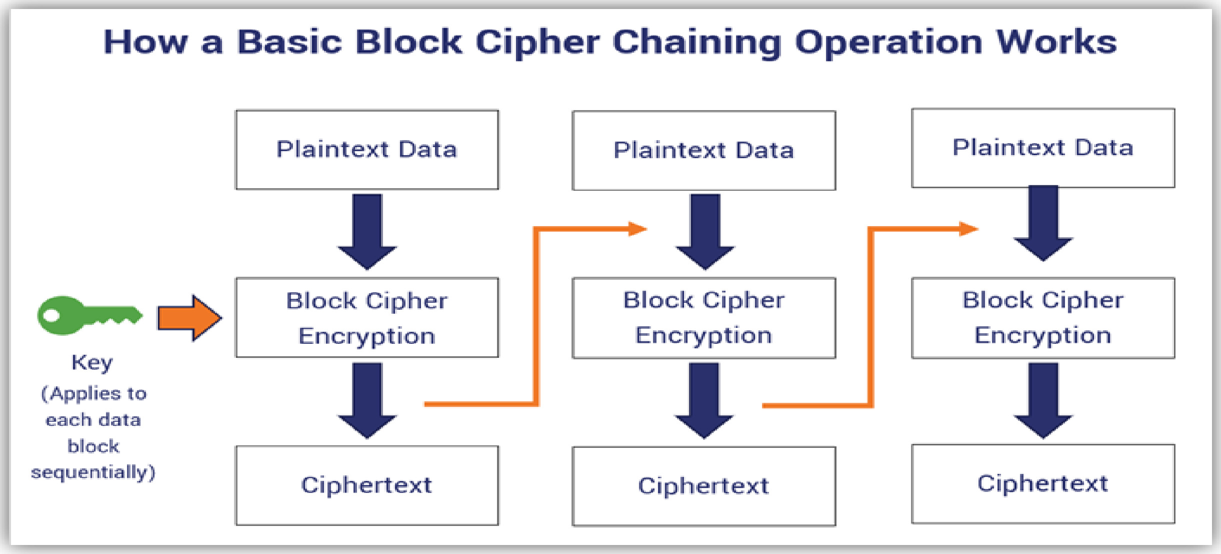
Examples of block ciphers

- **Advanced Encryption Standard (AES):**

  A modern and widely used standard that encrypts 128-bit blocks with keys of 128, 192, or 256 bits.

- • **Data Encryption Standard (DES):**
An older standard that encrypts 64-bit blocks with a 56-bit key, though it is now considered vulnerable to modern attacks.
- • **Triple DES (3DES):**

A cipher based on DES that applies the DES algorithm three times to increase security.



Block Cipher modes of Operation

Encryption algorithms are divided into two categories based on the input type: block cipher and stream cipher. A block cipher is an encryption algorithm that takes a fixed-size input (e.g., b bits) and produces a ciphertext of b bits. If the input is larger than b bits, it can be divided further. There are several modes of operation for a block cipher, each suited for different applications and uses.

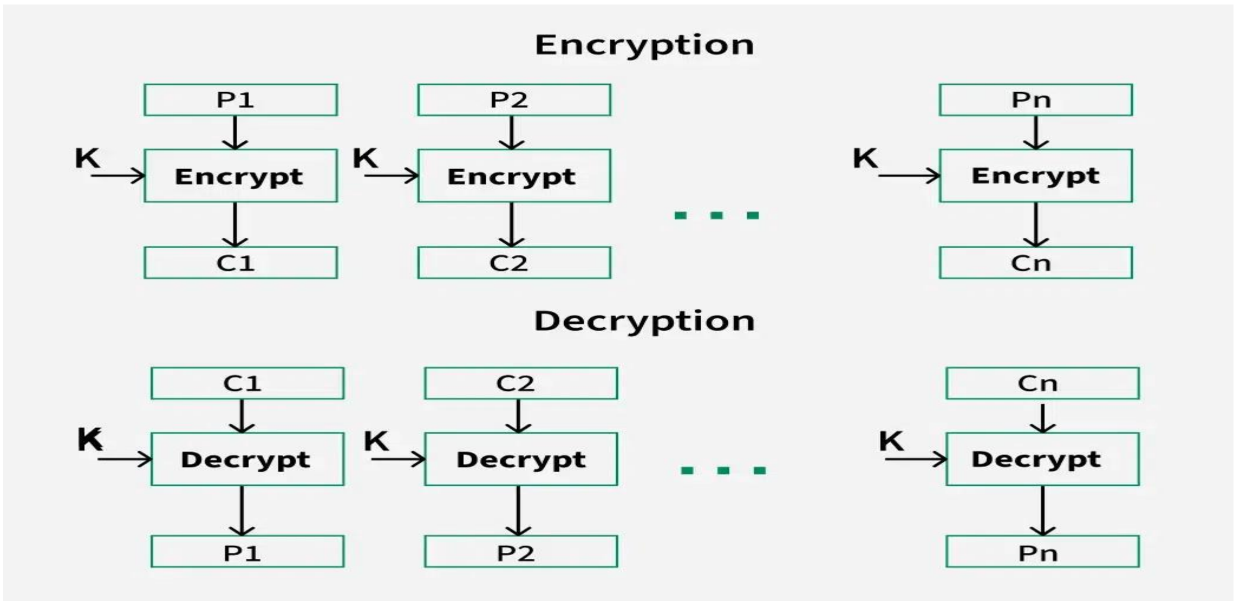
What are Block Cipher Modes of Operation?

Block Cipher Modes of Operation define how to securely encrypt and decrypt large amounts of data using a block cipher. A block cipher is an encryption algorithm that processes data in fixed-size blocks (e.g., 128 bits) rather than one bit at a time. However, to encrypt data larger than a single block, different modes of operation are used to ensure both security and efficiency. Here are a few common modes. **Here are a few common modes:**

Electronic Code Book (ECB)

The electronic codebook is the easiest block cipher mode of functioning. It is easier because of the direct encryption of each block of input plaintext and output is in the form of blocks of encrypted ciphertext. Generally, if a message is larger than *b* bits in size, it can be broken down into a bunch of blocks and the procedure is repeated.

The procedure of ECB is illustrated below:



Advantages of using ECB

Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.

Simple way of the block cipher.

Disadvantages of using ECB

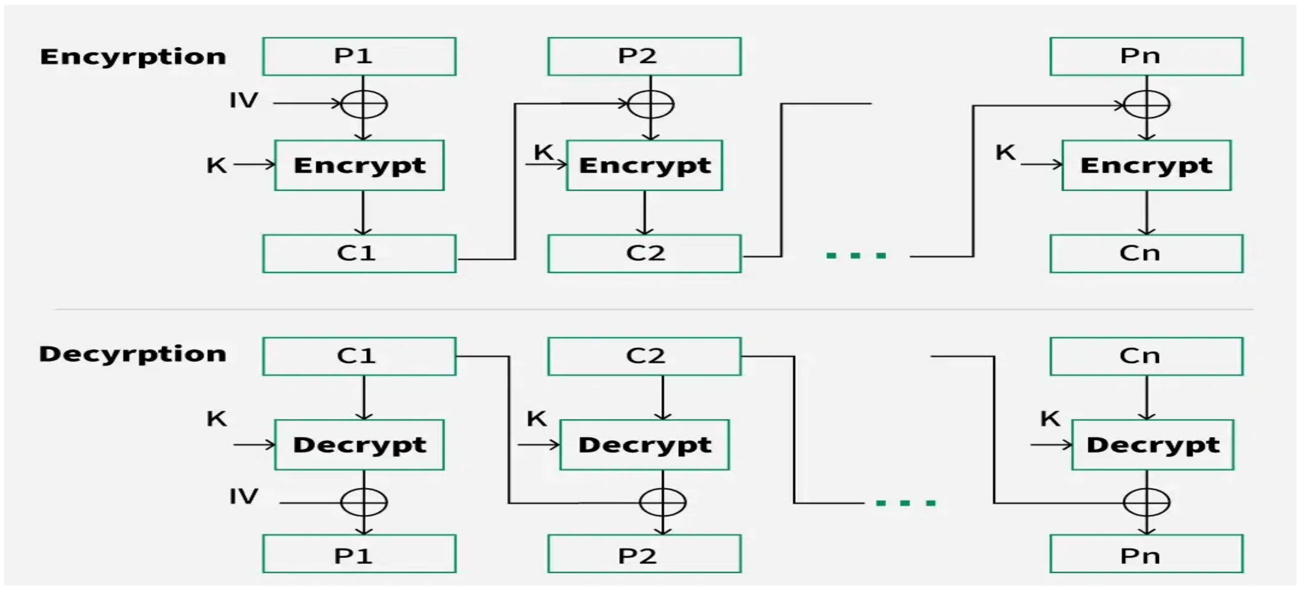
Prone to cryptanalysis since there is a direct relationship between plaintext and ciphertext.

Identical plaintext blocks produce identical ciphertext blocks, which can reveal patterns.

Cipher Block Chaining

Cipher block chaining or CBC is an advancement made on ECB since ECB compromises some security requirements. In CBC, the previous cipher block is given as input to the next encryption algorithm after XOR with the original plaintext block. In a nutshell here, a cipher block is produced by encrypting an XOR output of the previous cipher block and present plaintext block.

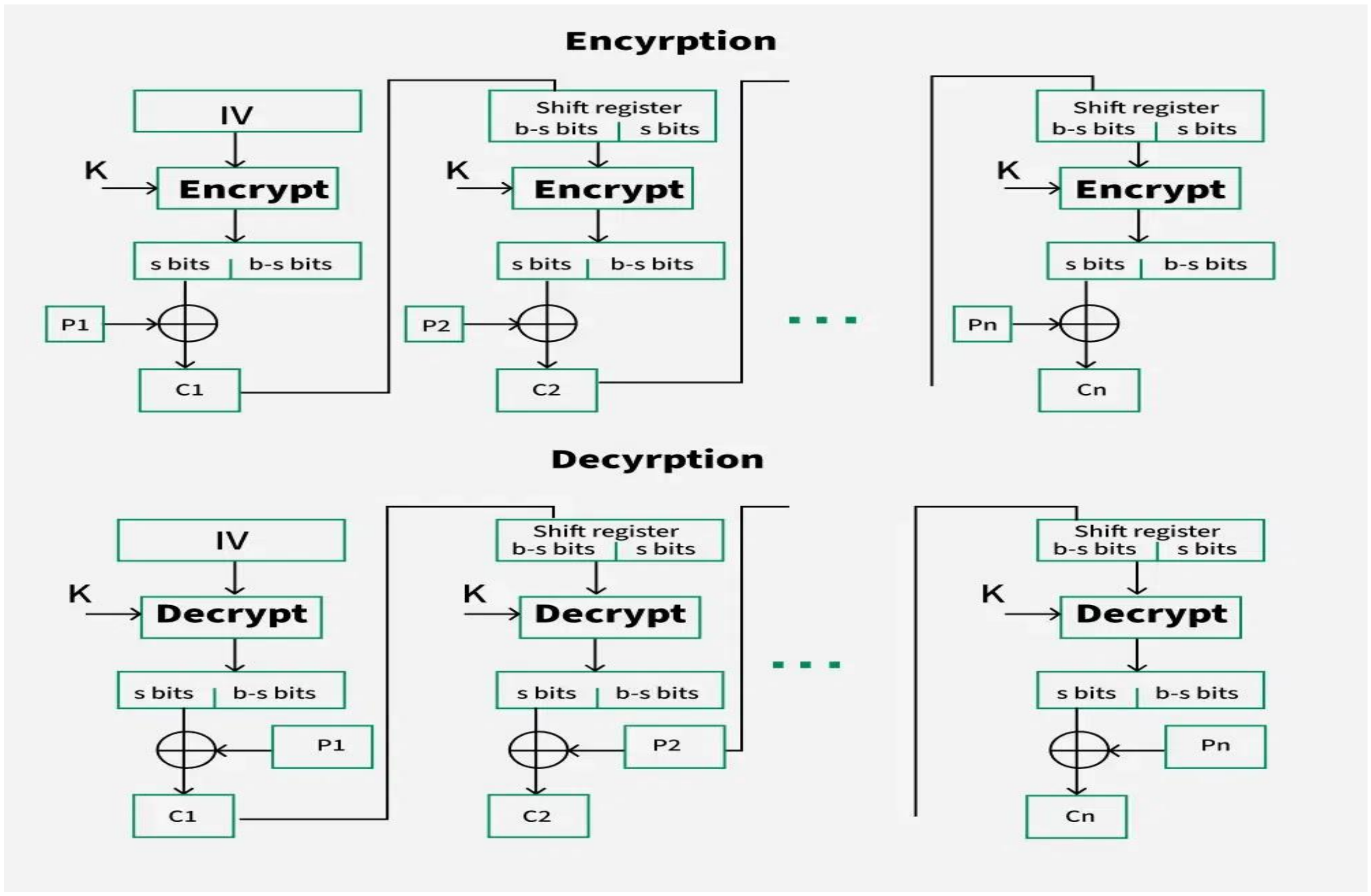
The process is illustrated here:



Cipher Feedback Mode (CFB)

In this mode the cipher is given as feedback to the next block of encryption with some new specifications: first, an initial vector IV is used for first encryption and output bits are divided as a set of s and b-s bits. The left-hand side s bits are selected along with plaintext bits to which an XOR operation is applied. The result is given as input to a shift register having b-s bits to lhs, s bits to rhs and the process continues. The encryption and decryption process for the same is shown below, both of them use encryption algorithms.

Cipher Feedback Mode


Encyrption
IV
K
Encrypt
s bits
b-s bits
P1
C1
Shift register
b-s bits
s bits
K
Encrypt
s bits
b-s bits
P2
C2
Shift register
b-s bits
s bits
K
Encrypt
s bits
b-s bits
Pn
Cn
Decyrption
IV
K
Decrypt
s bits
b-s bits
P1
C1
Shift register
b-s bits
s bits
K
Decrypt
s bits
b-s bits
P2
C2
Shift register
b-s bits
s bits
K
Decrypt
s bits
b-s bits
Pn
Cn

Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is a highly trusted **encryption algorithm** used to secure data by converting it into an unreadable format without the proper key. It is developed by the National Institute of Standards and Technology (NIST) in 2001. It is is widely used today as it is much stronger than DES and triple DES despite being harder to implement. **AES encryption** uses various **key lengths** (128, 192, or 256 bits) to provide strong protection against unauthorized access. This **data security** measure is efficient and widely implemented in securing **internet communication**, protecting **sensitive data**, and encrypting files. AES, a cornerstone of modern cryptography, is recognized globally for its ability to keep information safe from cyber threats.

- AES is a Block Cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.

That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text. AES relies on the substitution-permutation network principle, which is performed using a series of linked operations that involve replacing and shuffling the input data.

Working of The Cipher

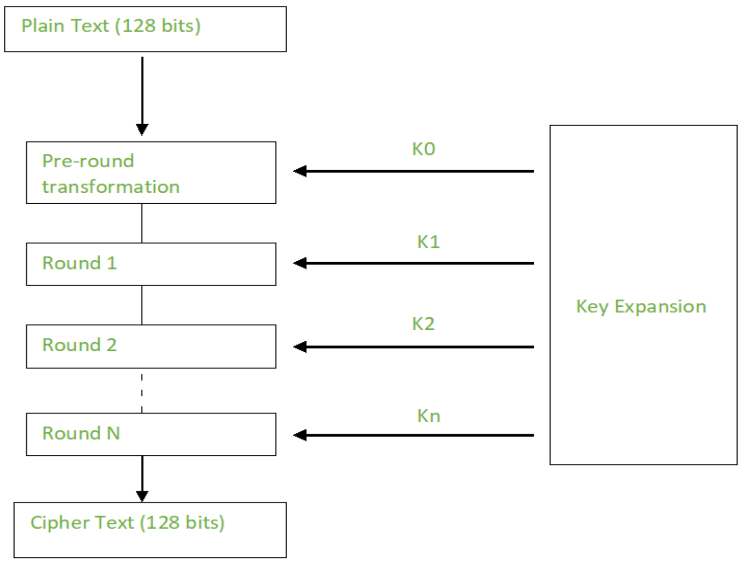
AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.

The number of rounds depends on the key length as follows :

N (Number of Rounds)	Key Size (in bits)
10	128
12	192
14	256

Creation of Round Keys

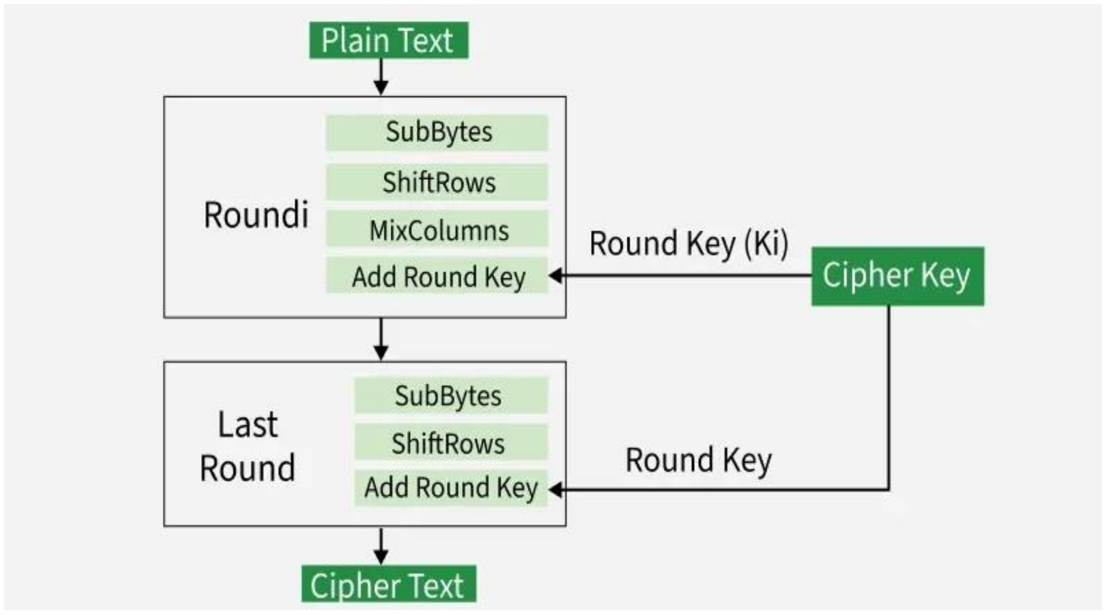
A Key Schedule algorithm calculates all the round keys from the key. So the initial key is used to create many different round keys which will be used in the corresponding round of the encryption.



Encryption

AES considers each block as a 16-byte (4 byte x 4 byte = 128) grid in a column-major arrangement.

[b0 | b4 | b8 | b12 |
| b1 | b5 | b9 | b13 |
| b2 | b6 | b10| b14 |
| b3 | b7 | b11| b15]



Each round comprises of 4 steps :

- SubBytes
- ShiftRows
- MixColumns
- Add Round Key

Step1. Sub Bytes

This step implements the substitution.

In this step, each byte is substituted by another byte. It is performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16-byte (4 x 4) matrix like before.

The next two steps implement the permutation.

Step2. Shift Rows

This step is just as it sounds. Each row is shifted a particular number of times.

- The first row is not shifted
- The second row is shifted once to the left.
- The third row is shifted twice to the left.
- The fourth row is shifted thrice to the left.

(A left circular shift is performed.)

[b0 | b1 | b2 | b3] [b0 | b1 | b2 | b3]
| b4 | b5 | b6 | b7 | -> | b5 | b6 | b7 | b4 |
| b8 | b9 | b10 | b11 | | b10 | b11 | b8 | b9 |
[b12 | b13 | b14 | b15] [b15 | b12 | b13 | b14]

Step 3: Mix Columns

This step is a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

This step is skipped in the last round.

[c0] [2 3 1 1] [b0]
| c1 | = | 1 2 3 1 | | b1 |
| c2 | | 1 1 2 3 | | b2 |
[c3] [3 1 1 2] [b3]

Step 4: Add Round Keys

- Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes are not considered as a grid but just as 128 bits of data.
- After all these rounds 128 bits of encrypted data are given back as output. This process is repeated until all the data to be encrypted undergoes this process.

Decryption

The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes. Each 128 blocks goes through the 10,12 or 14 rounds depending on the key size.

The stages of each round of decryption are as follows :

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte

The decryption process is the encryption process done in reverse so I will explain the steps with notable differences.

Inverse MixColumns

- This step is similar to the Mix Columns step in encryption but differs in the matrix used to carry out the operation.
- Mix Columns Operation each column is mixed independent of the other.
- Matrix multiplication is used. The output of this step is the matrix multiplication of the old values and a

constant matrix

[b0] = [14 11 13 9] [c0]
[b1]=[9 14 11 13] [c1]
[b2] =[13 9 14 11] [c2]
[b3]=[11 13 9 14] [c3]

Inverse SubBytes

- Inverse S-box is used as a lookup table and using which the bytes are substituted during decryption.

- Function Substitute performs a byte substitution on each byte of the input word. For this purpose, it uses an S-box.

Applications of AES

AES is widely used in many applications which require secure data storage and transmission. Some common use cases include:

- **Wireless security:** AES is used in securing wireless networks, such as Wi-Fi networks, to ensure data confidentiality and prevent unauthorized access.

- **Database Encryption:** AES can be applied to encrypt sensitive data stored in databases. This helps protect personal information, financial records, and other confidential data from unauthorized access in case of a data breach.

- **Secure communications:** AES is widely used in protocols such as internet communications, email, instant messaging, and voice/video calls. It ensures that the data remains confidential.

- **Data storage:** AES is used to encrypt sensitive data stored on hard drives, USB drives, and other storage media, protecting it from unauthorized access in case of loss or theft.

- **Virtual Private Networks (VPNs):** AES is commonly used in VPN protocols to secure the communication between a user's device and a remote server. It ensures that data sent and received through the VPN remains private and cannot be deciphered by eavesdroppers.

- **Secure Storage of Passwords:** AES encryption is commonly employed to store passwords securely. Instead of storing plaintext passwords, the encrypted version is stored. This adds an extra layer of security and protects user credentials in case of unauthorized access to the storage.

- **File and Disk Encryption:** AES is used to encrypt files and folders on computers, external storage devices, and cloud storage. It protects sensitive data stored on devices or during data transfer to prevent unauthorized access.

Simplified International Data Encryption Algorithm (IDEA)

The International Data Encryption Algorithm (IDEA) is a symmetric-key block cipher that was first introduced in 1991. It was designed to provide secure encryption for digital data and is used in a variety of applications, such as secure communications, financial transactions, and electronic voting systems. In cryptography, block ciphers are very important in the designing of many cryptographic algorithms and are widely used to encrypt the bulk of data in chunks. By chunks, it means that the cipher takes a fixed size of the plaintext in the encryption process and generates a fixed-size ciphertext using a fixed-length key. An algorithm's strength is determined by its key length.

What is IDEA?

IDEA uses a block cipher with a block size of 64 bits and a key size of 128 bits. It uses a series of mathematical operations, including modular arithmetic, bit shifting, and exclusive OR (XOR) operations, to transform the plaintext into ciphertext. The cipher is designed to be highly secure and resistant to various types of attacks, including differential and linear cryptanalysis. One of the strengths of IDEA is its efficient implementation in software and hardware. The algorithm is relatively fast and requires only a small amount of memory and processing power. This makes it a popular choice for use in embedded systems and other applications where resources are limited.

IDEA has been widely used in various encryption applications, although it has been largely replaced by newer encryption algorithms such as AES (Advanced Encryption Standard) in recent years. However, IDEA is still considered to be a highly secure and effective encryption algorithm, and it continues to be used in some legacy systems and applications.

Historical Background of the International Data Encryption Algorithm (IDEA)

Development and Significance

IDEA was first introduced in 1991 by James Massey and Xuejia Lai as a replacement for the Data Encryption Standard (DES). It was designed to address the growing concerns about the security of DES, particularly its relatively short key length.

Key Contributions and Innovations

IDEA introduced several innovative features:

- Introduction of a 128-bit key length, revolutionary for its time.

- Novel combination of three algebraic groups: XOR, addition modulo 2^16, and multiplication modulo 2^16+1.
- Pioneering resistance to then-known cryptanalytic attacks.
- Implementation efficiency in both software and hardware platforms.

Overview of Simplified International Data Encryption Algorithm (IDEA)

The Simplified IDEA presented here is a modified version of the original algorithm, designed for educational purposes. While it maintains the core principles of IDEA, it uses smaller block and key sizes to simplify understanding and implementation.

Block Cipher Structure

The Simplified **International Data Encryption Algorithm (IDEA)** is a **symmetric key** **block** cipher that:

- **Plaintext block size:** 16 bits (divided into 4-bit chunks)
- **Key length:** 32 bits
- **Output:** 16-bit ciphertext
- **Processing:** Four complete rounds plus one half-round

Subkey Generation Process

- Initial key division into eight 4-bit subkeys.
- Key schedule generation through left rotation.
- Production of 28 subkeys (24 for complete rounds, 4 for half-round).
- Systematic distribution across rounds.

Overview of Rounds and Operations

Each complete round consists of:

- 14 distinct steps using three core operations
- Systematic transformation of data blocks

- Inter-round data swapping
- Final half-round with 4 operations

Mathematical Foundations

Core Operations:

1. Bitwise XOR (^)
2. Addition modulo 2^4 (+)
3. Multiplication modulo (2^4+1) (*)

Key Mathematical Properties:

- Use of algebraic groups with complementary properties
- Exploitation of modular arithmetic for confusion
- Implementation of systematic diffusion through operation mixing

Example Implementation:

1. Initial key expansion

2. Plaintext division into blocks

3. Round function application:

- Multiplication and addition operations
- XOR combinations
- Inter-round transformations

4. Final half-round processing

5. Ciphertext generation

Applications and Use Cases

1. Secure Communications

- Virtual Private Networks (VPNs)

- Encrypted messaging systems
- Secure email protocols

2. Data Protection

- File encryption
- Database security
- Storage encryption

3. Financial Security

- Electronic payment systems
- Banking applications
- Transaction security

Limitations and Challenges

1. Fixed block size constraints
2. Key schedule complexity
3. Implementation considerations in resource-constrained environments

Security Considerations

1. Theoretical vulnerabilities to specific attack vectors
2. Key management challenges
3. Integration complexities with modern systems

Detailed Algorithm Implementation and Round Operations

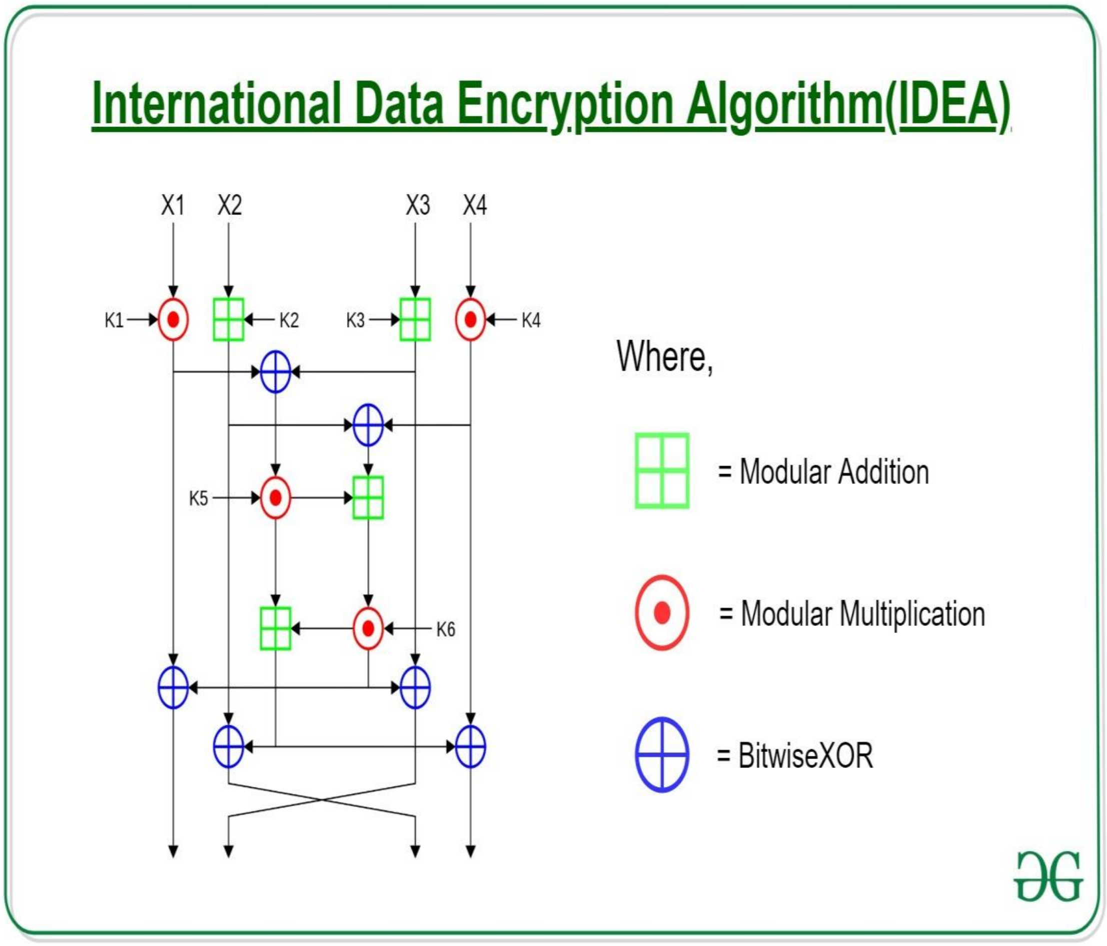
This algorithm involves a series of 4 identical complete rounds and 1 half-round. Each complete round involves a series of 14 steps that includes operations like:

- Bitwise XOR
- Addition modulo (2^4)

- Multiplication modulo $(2^4) + 1$

After 4 complete rounds, the final "half-round" consists of only the first 4 out of the 14 steps previously used in the full rounds. To perform these rounds, each binary notation must be converted to its equivalent decimal notation, perform the operation and the result obtained should be converted back to the binary representation for the final result of that particular step.

Key Schedule: 6 subkeys of 4 bits out of the 8 subkeys are used in each complete round, while 4 are used in the half-round. So, 4.5 rounds require 28 subkeys. The given key, 'K', directly gives the first 8 subkeys. By rotating the main key left by 6 bits between each group of 8, further groups of 8 subkeys are created, implying less than one rotation per round for the key (3 rotations).



	K1	**K2**	**K3**	**K4**	**K5**	**K6**
Round 1	1101	1100	0110	1111	0011	1111

	K1	**K2**	**K3**	**K4**	**K5**	**K6**
Round 2	0101	1001*****	0001	1011	1100	1111
Round 3	1101	0110	0111	0111*****	1111	0011
Round 4	1111	0101	1001	1101	1100	0110*****
Round 4.5	1111	1101	0110	0111		

* denotes a shift of bits

Notations used in the 14 steps:

Symbol	**Operation**
*	Multiplication modulo (2^4) +1
+	Addition modulo (2^4)
^	Bitwise XOR

The 16-bit plaintext can be represented as **X1 || X2 || X3 || X4**, each of size 4 bits. The 32-bit key is broken into 8 subkeys denoted as K1 || K2 || K3 || K4 || K5 || K6 || K7 || K8, again of size 4 bits each. Each round of 14 steps uses the three algebraic operation-Addition modulo (2^4), Multiplication modulo (2^4)+1 and Bitwise XOR. The steps involved are as follows:

1. X1 * K1
2. X2 + K2
3. X3 + K3
4. X4 * K4
5. Step 1 ^ Step 3
6. Step 2 ^ Step 4
7. Step 5 * K5
8. Step 6 + Step 7
9. Step 8 * K6
10. Step 7 + Step 9
11. Step 1 ^ Step 9
12. Step 3 ^ Step 9

13. Step 2 ^ Step 10

14. Step 4 ^ Step 10

The input to the next round is Step 11 || Step 13 || Step 12 || Step 14, which becomes X1 || X2 || X3 || X4. This swap between 12 and 13 takes place after each complete round, except the last complete round (4th round), where the input to the final half round is Step 11 || Step 12 || Step 13 || Step 14.

After last complete round, the half-round is as follows:

1. X1 * K1
2. X2 + K2
3. X3 + K3
4. X4 * K4

The final output is obtained by concatenating the blocks.

Example:

```
Key: 1101 1100 0110 1111 0011 1111 0101 1001
Plaintext: 1001 1100 1010 1100
Ciphertext: 1011 1011 0100 1011
```

Explanation:
The explanation is only for 1st complete round (the remaining can be implemented similarly) and the last half-round.

- **Round 1:**
  - From the plaintext: **X1 - 1001, X2 - 1100, X3 - 1010, X4 - 1100**
  - From the table above: **K1 - 1101, K2 - 1100, K3 - 0110, K4 - 1111, K5 - 0011, K6 - 1111**

Triple DES (3DES)

This article discusses various aspects of the Cryptography section. In this article, we are going to look at What is Triple DES. We will discuss its working and features. We will further dive into how the encryption process works in Triple DES. Then we will have a look at the advantages of Triple DES.

What is Triple DES?

Triple DES is an encryption algorithm based on the original Data Encryption Standard (DES). It is a symmetric encryption algorithm that uses multiple rounds of the Data Encryption Standard (DES) to improve security. It is also known as Triple DES because it uses the Data Encryption Standard (DES) cypher which takes three times to encrypt its data. It is essentially a block cypher used to encrypt data in 64-bit blocks. Security-wise, it outperforms the original Data Encryption Standard (DES). However, Triple DES is less efficient and slower than the Advanced Encryption Standard (AES).

Features of Triple DES

- It utilizes a triple layer of encryption which means it utilizes three different keys to encrypt the plaintext three times.
- It supports variable key sizes which range from 128 bits to 192 bits.
- It basically involves the usage of a symmetric key encryption system, which states that the same key is used for both encryption and decryption.
- It is a block cypher encryption algorithm that works with 64-bit blocks of plaintext at a time.
- It is suitable for legacy systems that require secure encryption.

Encryption Process

The Encryption process of Triple DES involves the following steps:-

Key Generation

This is the first step of the Encryption process of Triple DES. In this step, three unique keys are generated using a key derivation algorithm.

Initial Permutation

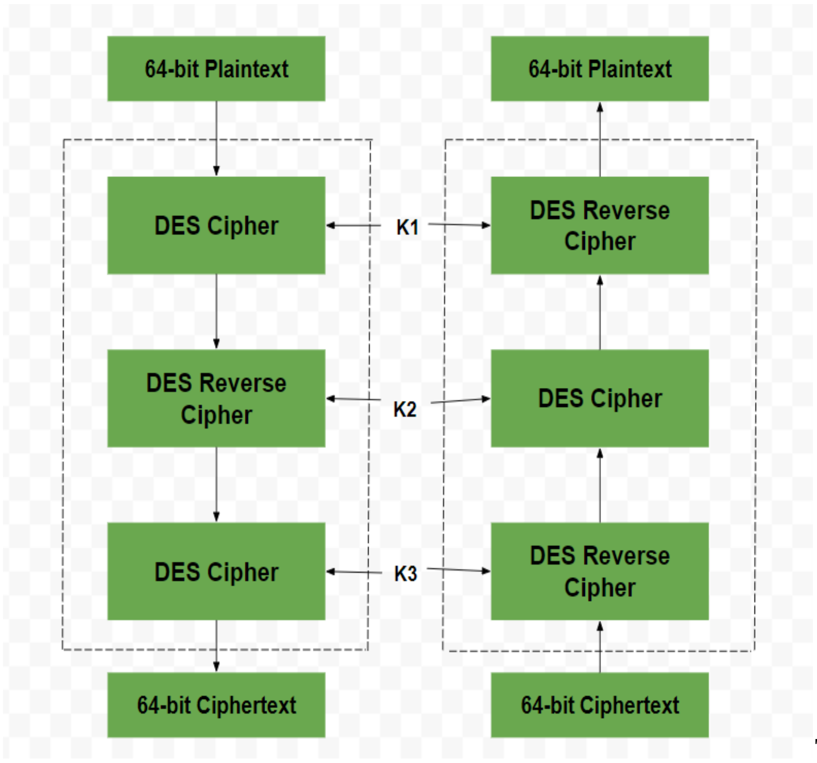
This step comes after the process of Key Generation. It involves the rearrangement of the bits of the plaintext according to a predefined permutation table.

Three Rounds of Encryption

This is regarded as the most important round of the encryption process of Triple DES. It consists of multiple rounds typically 48 rounds in total. In this step, the plaintext is processed three times and get encrypted, each time we take use of a different key, to create three layers of encryption.

Final Permutation

It completes the Triple DES encryption process. In this step, the resulting ciphertext block undergoes a final permutation (FP) operation, which is the inverse of the initial permutation. It returns the bits of the ciphertext block to their original order.



Triple DES

Advantages of Triple DES

- It provides three layered encryption technique which provides enhanced security features.
- It offers backward compatibility with Data Encryption Standard which means it can use legacy system that DES uses.
- It supports variable key sizes, which led to enhanced security.

- It is widely used encryption algorithm and is used with many encryption standards and protocols.

Applications of Triple DES

- **Financial Transactions:** Triple DES is widely used in financial transactions because it secures the transaction that takes place like online banking, credit card payment, etc.

- **Data Protection:** Triple DES is often used to protect sensitive data which are stored on computers, servers, and other electronic devices. It is used in various fields such as healthcare, government sector, etc.

- **Virtual Private Networks:** Triple DES is used to secure the communication process between remote locations. It is done by securing the virtual private networks.

- **Authentication and Digital Signatures:** Triple DES can be used in combination with cryptographic hash functions for generating digital signatures and verifying the authenticity of digital documents and message