Cryptography, Network & Software Security

Jaspreet Singh

Project Engineer
E-Security
CDAC, Hyderabad

Overview

- ➔ Symmetric and Asymmetric Key Cryptography
- ➔ Diffie-Hellman Key Exchange & Man-in-the-Middle Attack
- ➔ Using OpenSSL for Setting up CA [Lab]
- ➔ Stream Ciphers & Block Ciphers, RSA
- ➔ Introduction to Hash Functions, MAC, HMAC

Symmetric Key Cryptography

Definition: Uses the same key for both encryption and decryption.

Examples:

- **AES (Advanced Encryption Standard):** Widely used, strong encryption standard.
- **DES (Data Encryption Standard):** Older, less secure, replaced by AES.

Pros:

- Fast and efficient for large data.
- Simple implementation.

Cons:

- Key distribution problem.
- Less secure for communication over insecure channels.

Asymmetric Key Cryptography

- **Definition:** Uses a pair of keys – a public key for encryption and a private key for decryption.

- **Examples:**
  - **RSA (Rivest-Shamir-Adleman):** Popular for secure data transmission.
  - **ECC (Elliptic Curve Cryptography):** Provides similar security with smaller key sizes.
- **Pros:**
  - Solves the key distribution problem.
  - More secure for communication over insecure channels.
- **Cons:**
  - Slower than symmetric key cryptography.
  - Computationally intensive.

Symmetric vs Asymmetric Key Cryptography

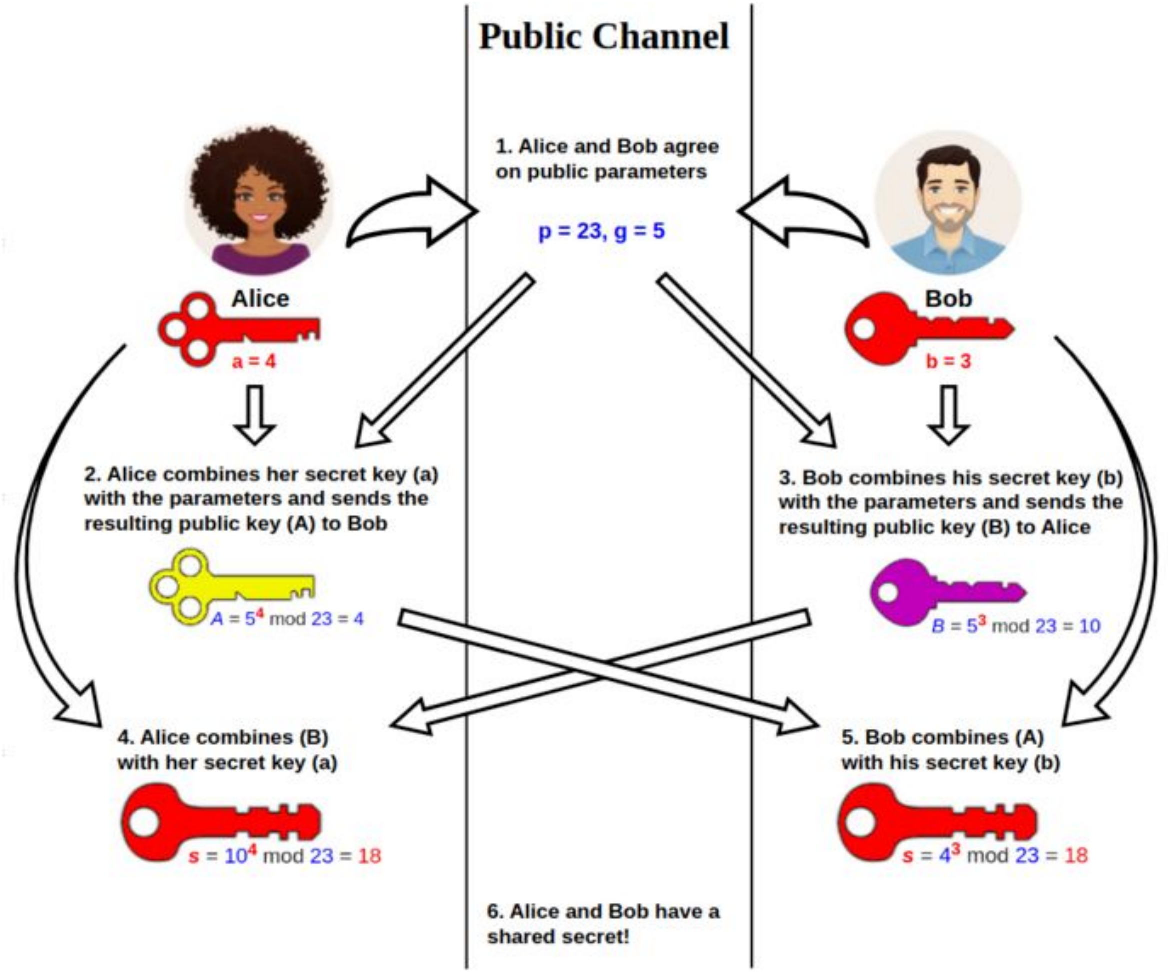
Feature	Symmetric Key Cryptography	Asymmetric Key Cryptography
Key Used	Single key	Pair of keys (public and private)
Speed	Faster	Slower
Security	Secure if key is managed properly	More secure for public channels
Key Distribution	Difficult	Easier
Examples	AES, DES	RSA, ECC
Use Cases	Bulk data encryption	Secure key exchange, digital signatures

Diffie-Hellman Key Exchange

- **Introduction:** A method for securely exchanging cryptographic keys over a public channel.

- **How It Works:**
  1. Both parties agree on a large prime number and a base (public values).
  2. Each party selects a private key and computes a public value.
  3. Public values are exchanged.
  4. Each party computes the shared secret using their private key and the other party's public value.

- **Applications:** Establishing a shared secret key for symmetric encryption.

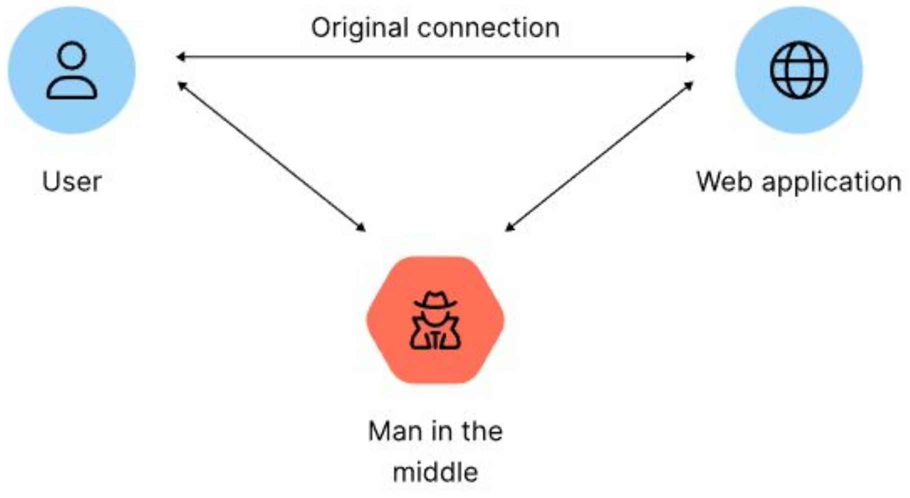


Man-in-the-Middle Attack

Explanation: An attacker intercepts and possibly alters the communication between two parties who believe they are directly communicating with each other.

How it Exploits Diffie-Hellman:

1. Attacker intercepts the public values exchanged.
2. Establishes independent keys with each party.
3. Decrypts, reads, and possibly alters the communication.



Mitigation Techniques:

- Use digital signatures.
- Employ Certificate Authorities (CAs).

Lab: Using OpenSSL for Setting Up a CA

Introduction to OpenSSL:

- A robust, full-featured open-source toolkit for the TLS and SSL protocols.
- Provides various cryptographic functions.

Steps to Set Up a Certificate Authority (CA):

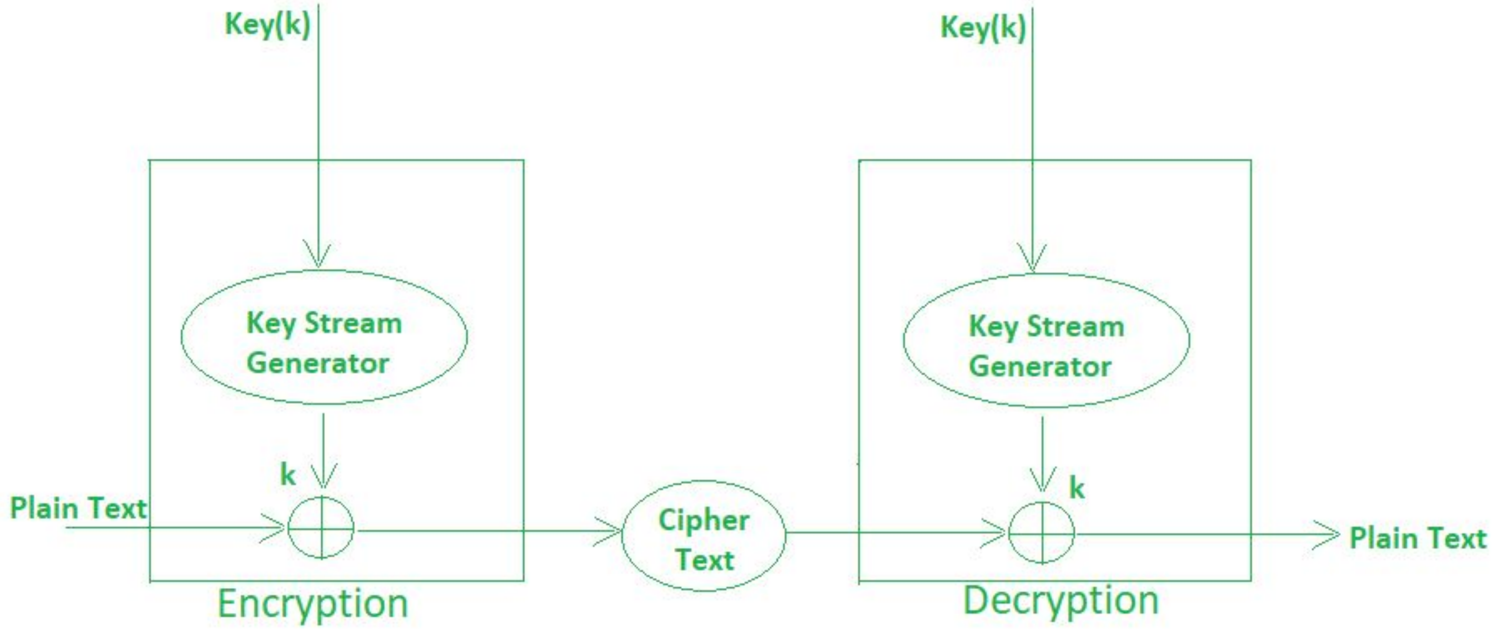
1. Install OpenSSL.
2. Generate a private key for the CA.
3. Create a self-signed root certificate.
4. Configure the OpenSSL CA directory structure.
5. Issue certificates.

Stream Ciphers

Definition: Encrypts plaintext one byte or bit at a time.

Examples:

- ○ **RC4:** Simple and fast, used in SSL/TLS, now considered insecure.
- **Applications:** Real-time data encryption like secure voice calls or video streaming.



Stream Ciphers

For Encryption,

- Plain Text and Keystream produces Cipher Text (Same keystream will be used for decryption.).
- The Plaintext will undergo XOR operation with keystream bit-by-bit and produces the Cipher Text.

Plain Text	10011001
Keystream	11000011
Cipher Text	01011010

For Decryption,

Cipher Text and Keystream gives the original Plain Text (Same keystream will be used for encryption.).
The Ciphertext will undergo XOR operation with keystream bit-by-bit and produces the actual Plain Text.

Cipher Text	01011010
Keystream	11000011
Plain Text	10011001

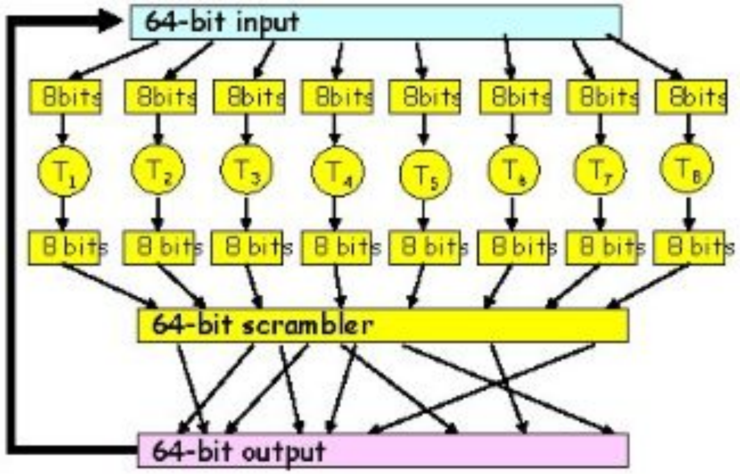
Block Cipher

Definition: Encrypts data in fixed-size blocks (e.g., 128-bit blocks).
Examples:

- **AES:** Secure, widely used, supports 128, 192, and 256-bit keys.
- **DES:** Outdated, replaced by AES.

Mode of Operations:

- **ECB (Electronic Codebook):** Simple, identical blocks of plaintext encrypted into identical ciphertext blocks.
- **CBC (Cipher Block Chaining):** Each plaintext block XORed with the previous ciphertext block before being encrypted.
- **CFB (Cipher Feedback):** Turns a block cipher into a stream cipher.
- **OFB (Output Feedback):** Similar to CFB but more secure.



Electronic Code Book (ECB)

Concept:

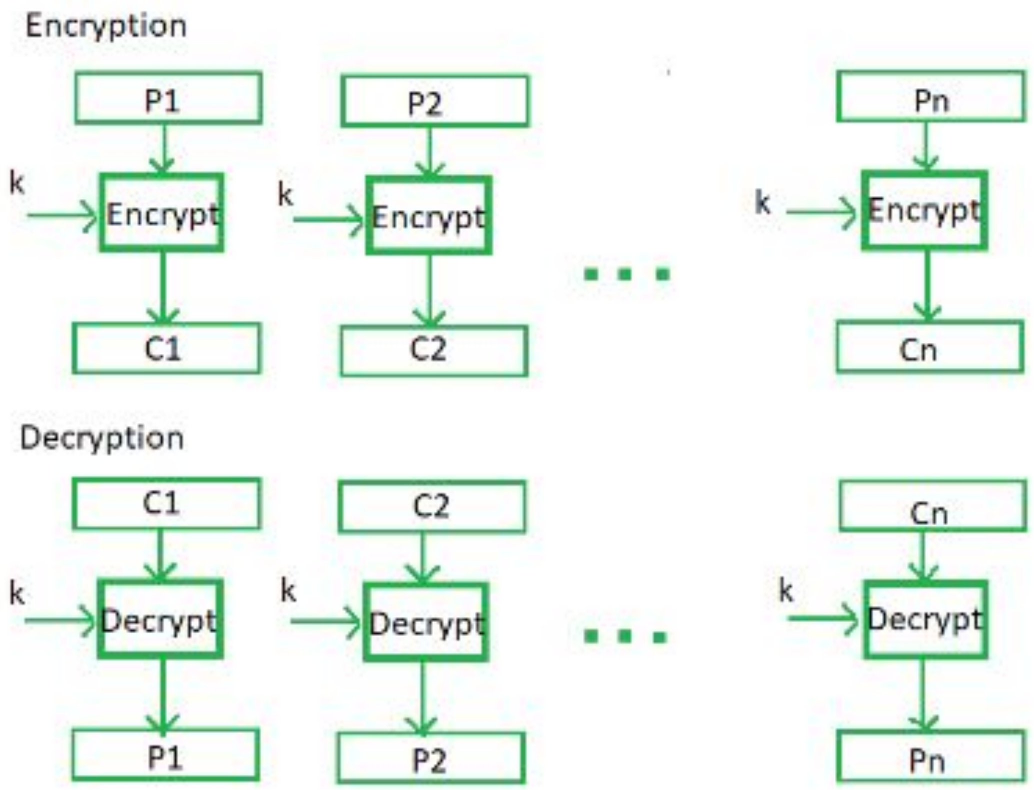
- The simplest mode.
- Each block of plaintext is encrypted independently using the same key.

Advantages:

- Easy to understand and implement.
- Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.

Disadvantages:

- Identical plaintext blocks produce identical ciphertext blocks, making it vulnerable to pattern analysis.
- Not suitable for encrypting large amounts of data.



Cipher Block Chaining (CBC)

Concept:

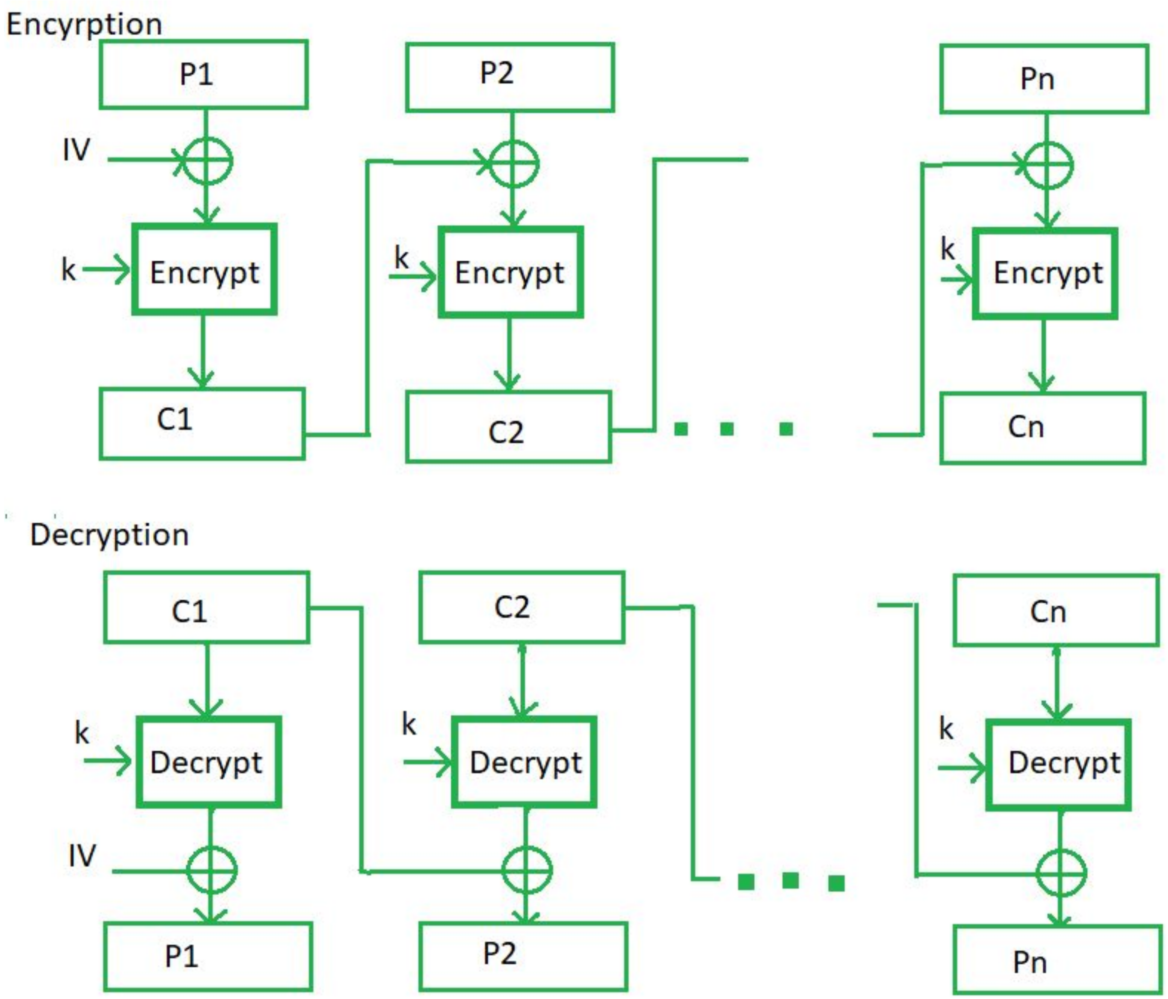
- Each plaintext block is XORed with the previous ciphertext block before being encrypted.
- The first block is XORed with an Initialization Vector (IV).

Advantages:

- Identical plaintext blocks result in different ciphertext blocks if different IVs are used.
- More secure than ECB.

Disadvantages:

- Requires IV, which must be unique and unpredictable.
- Errors propagate: a single bit error in a ciphertext block affects the corresponding and the next plaintext block during decryption.



Cipher Feedback (CFB)

Concept:

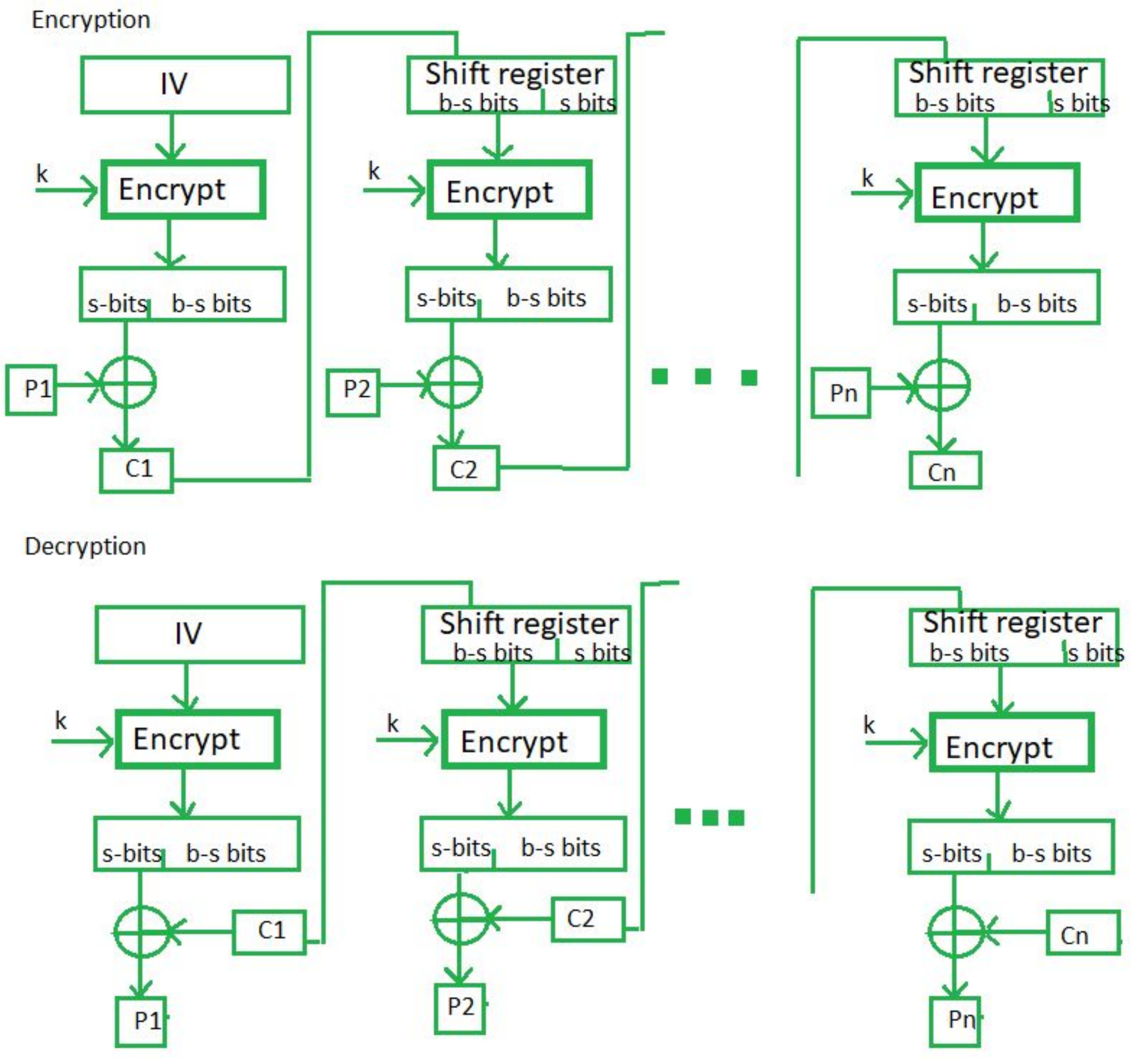
- Converts block cipher into a self-synchronizing stream cipher.
- Encrypts small segments (less than a block) of plaintext.

Advantages:

- Suitable for encrypting data of arbitrary size.
- Can start decrypting without needing the full ciphertext.

Disadvantages:

- Errors propagate: a bit error in ciphertext affects the corresponding segment in plaintext and the subsequent segment.



Output Feedback (OFB)

Concept:

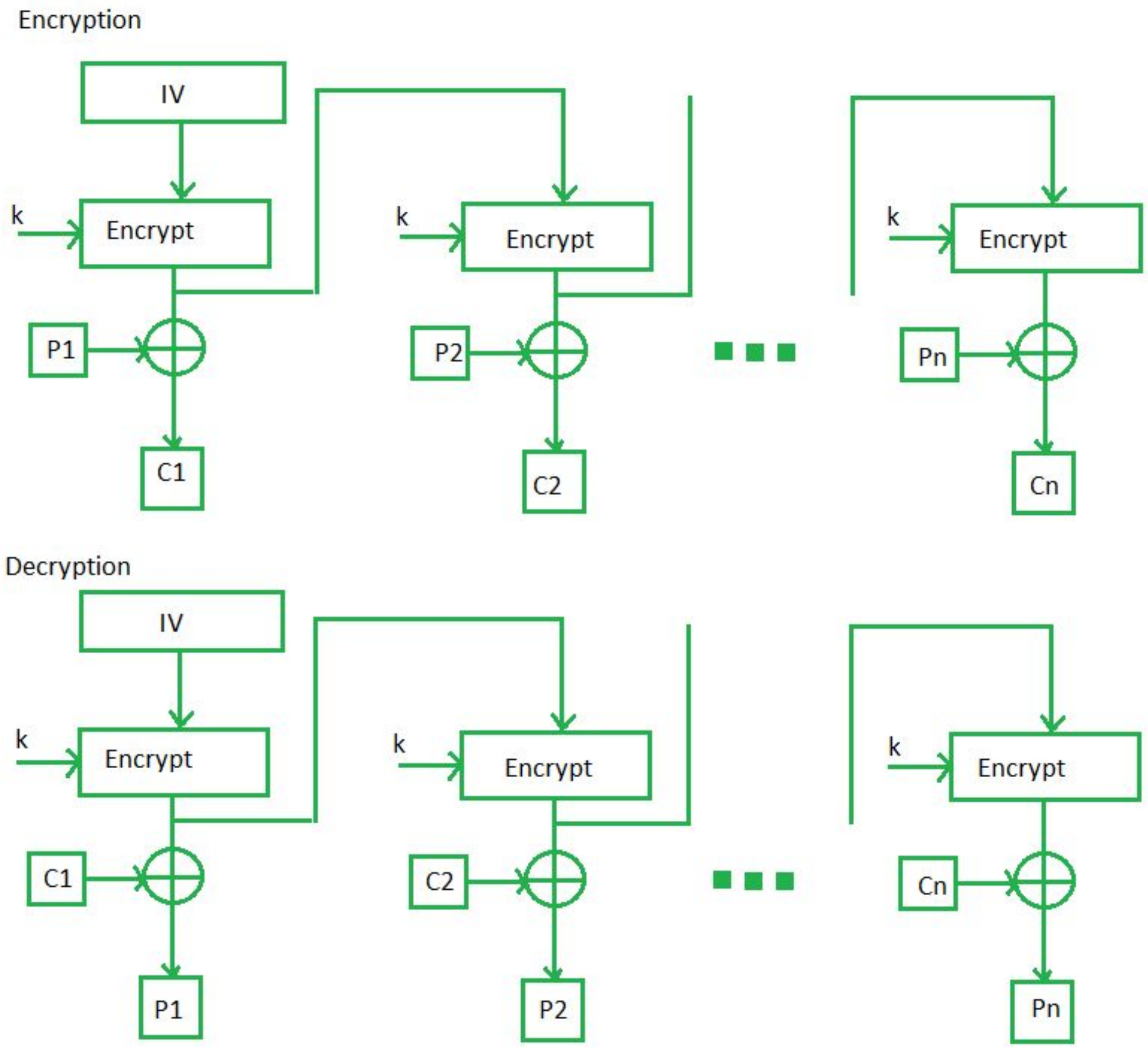
- Converts block cipher into a synchronous stream cipher.
- Generates keystream blocks, which are XORed with plaintext blocks.

Advantages:

- Errors do not propagate: a bit error in ciphertext affects only the corresponding bit in plaintext.

Disadvantages:

- Requires a unique and unpredictable IV.
- Synchronization required for decryption: both parties must use the same keystream sequence.



Counter (CTR) Mode

Concept:

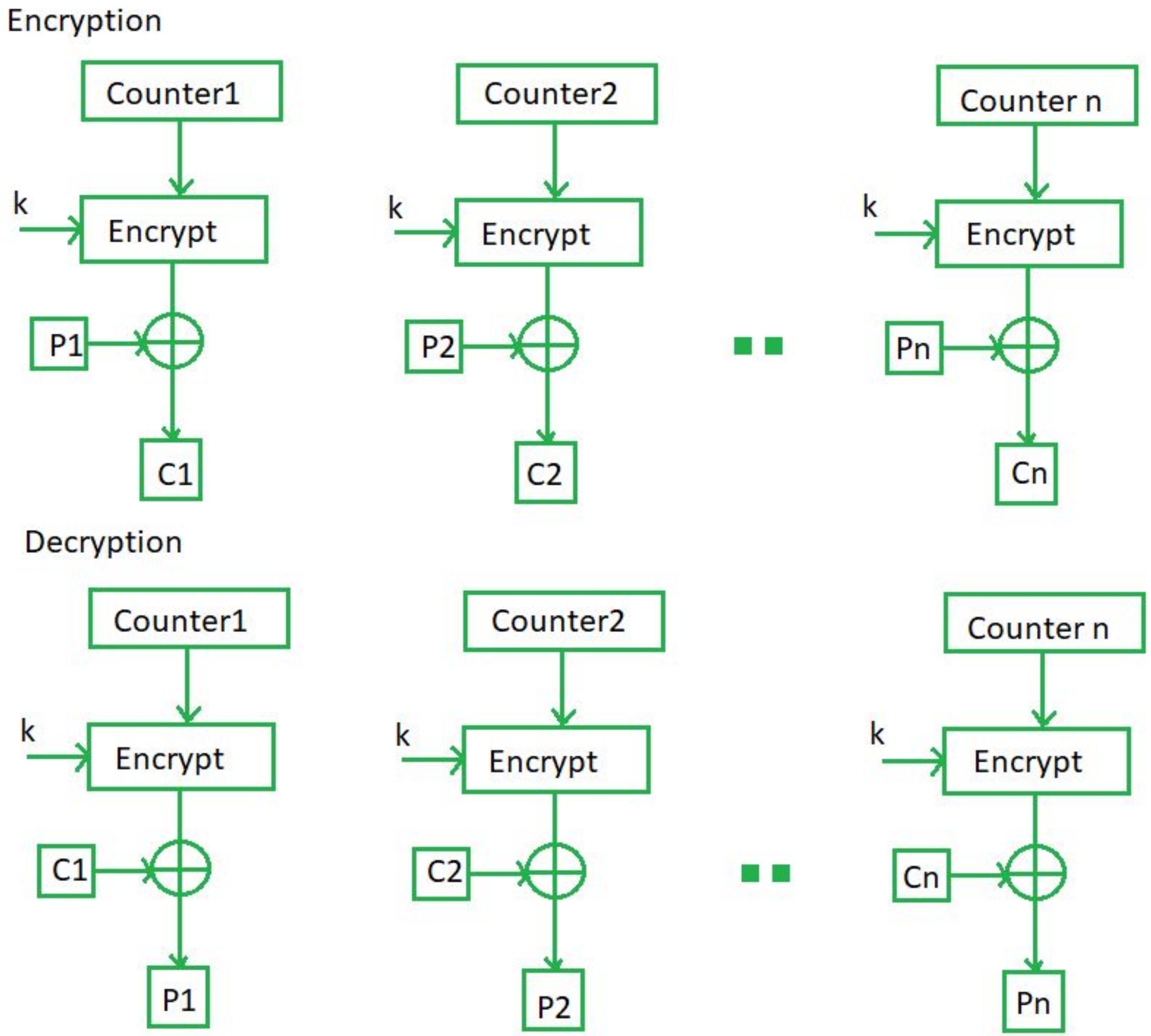
- Converts block cipher into a stream cipher.
- Encrypts successive values of a counter, which are then XORed with plaintext blocks.

Advantages:

- Parallelizable: encryption and decryption can be done in parallel, improving performance.
- Errors do not propagate.

Disadvantages:

- Requires a unique and predictable counter value.



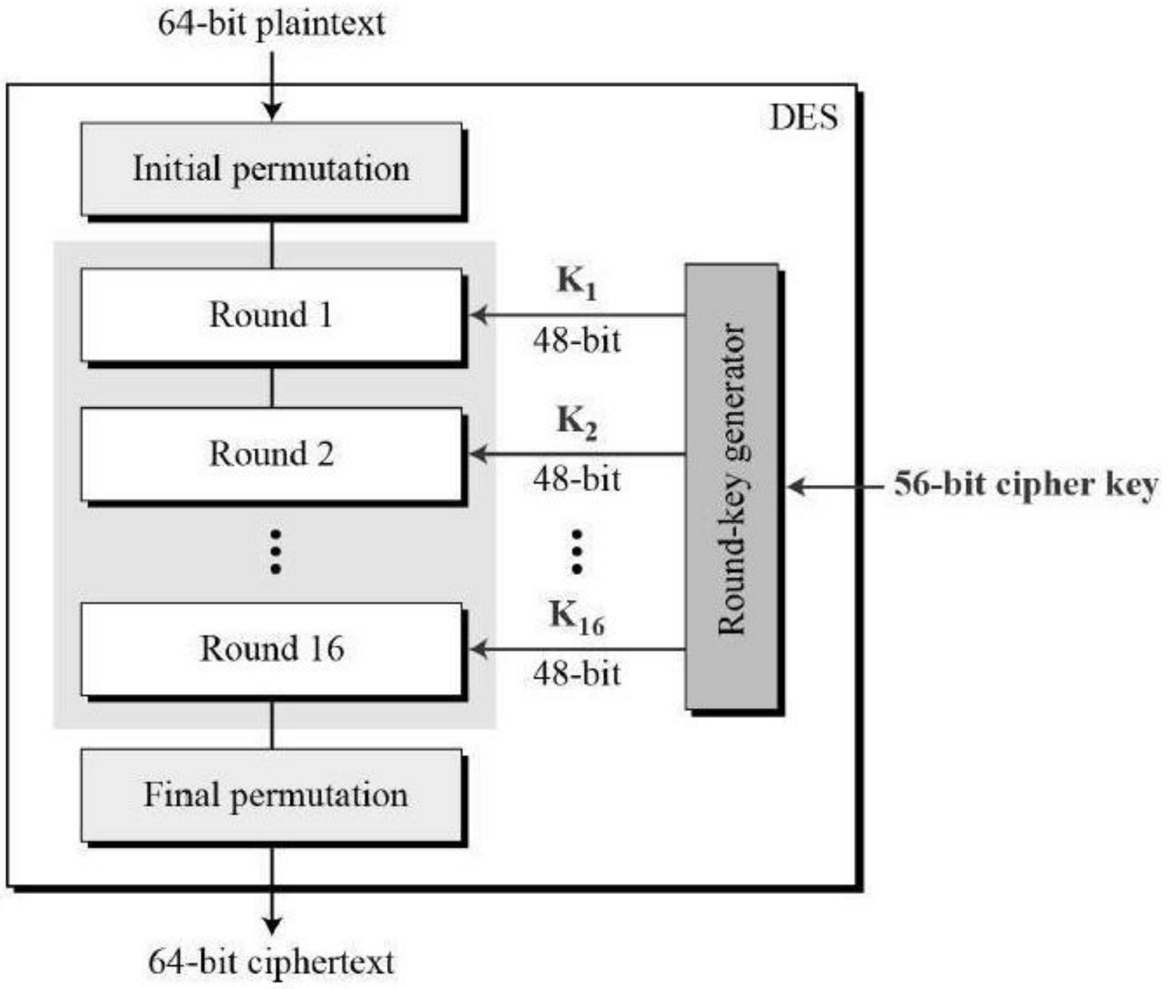
Data Encryption Standard (DES)

Overview:

- DES is an older symmetric-key algorithm for encryption.
- Developed in the 1970s by IBM and standardized by NIST in 1977.
- Encrypts data in 64-bit blocks using a 56-bit key.

Security:

- DES is considered insecure due to its short key length, which is vulnerable to brute-force attacks.



Data Encryption Standard (DES)

Initial Permutation (IP):

- The 64-bit plaintext block is permuted according to a fixed table.

Rounds of Processing:

- DES uses 16 rounds of the Feistel structure.
- Each round consists of the following:
  1. **Split:** The 64-bit block is split into two 32-bit halves: Left (L) and Right (R).
  2. **Round Function (F):**
     - R is expanded to 48 bits using an expansion function.
     - The expanded R is XORed with a 48-bit round key derived from the 56-bit key.
     - The result is passed through 8 S-boxes, reducing it back to 32 bits.
     - The output of the S-boxes is permuted.
  3. **XOR and Swap:** The result from the round function is XORed with L, then L and R are swapped.

Final Permutation (FP):

- After 16 rounds, the final L and R are concatenated and passed through the inverse of the initial permutation.

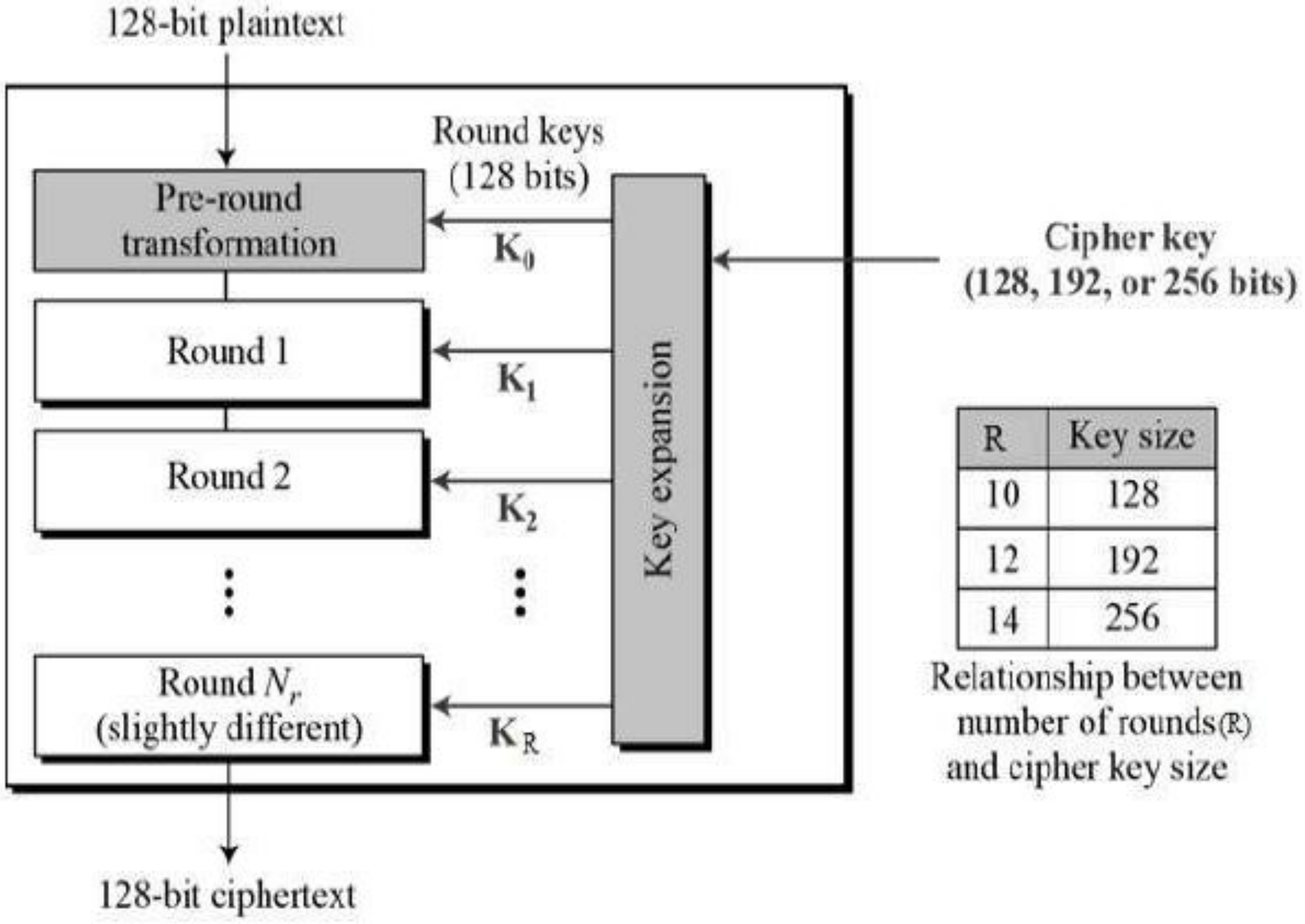
Advanced Encryption Standard (AES)

Overview:

- AES is the successor to DES and is more secure and efficient.
- Standardized by NIST in 2001 after a public competition.
- Encrypts data in 128-bit blocks with key lengths of 128, 192, or 256 bits.

Security:

- AES is considered very secure due to its larger key sizes and complex structure, making it resistant to all known practical attacks.



Advanced Encryption Standard (AES)

1.Initialization:

- The 128-bit plaintext block is arranged into a 4x4 matrix called the state.

2.Key Expansion:

- The key is expanded into multiple round keys.

3.Rounds of Processing:

- AES uses 10, 12, or 14 rounds depending on the key size.
- Each round consists of four operations:
  1. **SubBytes:**
     - Each byte in the state matrix is replaced with a corresponding byte from an S-box.
  2. **ShiftRows:**
     - Rows of the state matrix are shifted cyclically.
  3. **MixColumns (except in the last round):**
     - Columns of the state matrix are mixed using linear transformation.
  4. **AddRoundKey:**
     - The state matrix is XORed with the round key.

4.Final Round:

- The final round omits the MixColumns step.

DES vs AES

Difference	DES (Data Encryption Standard)	AES (Advanced Encryption Standard)
Key Length	56 bits	128, 192, or 256 bits
Block Size	64 bits	128 bits
Number of Rounds	16	10, 12, or 14 (depending on key size)
Structure	Feistel Network	Substitution-Permutation Network
Security	Vulnerable to brute-force attacks due to short key length	Highly secure, resistant to all known practical attacks

RSA(Rivest-Shamir-Adleman)

Definition: An asymmetric cryptographic algorithm.
How It Works:

- Key Generation: Involves two large prime numbers.
- Encryption: Uses the public key.
- Decryption: Uses the private key.

Applications: Secure data transmission, digital signatures

RSA(Rivest-Shamir-Adleman)

1. Key Generation:

- Select two distinct prime numbers, p and q

Compute n:

- Calculate n = p×q
- n is used as the modulus for both the public and private keys.

Compute Euler's Totient Function ϕ(n):

- Calculate ϕ(n) = (p−1)×(q−1)
- ϕ(n) is the number of integers less than nnn that are coprime with n.

Choose Public Exponent eee:

- Select an integer e such that 1 <e < ϕ(n) and gcd(e,ϕ(n)) = 1.
- e is typically chosen to be 65537 for its efficiency and security.

Compute Private Exponent ddd:

- Calculate d such that d x e ≡ 1 mod ϕ(n).
- d is the modular multiplicative inverse of e modulo ϕ(n).

Encryption

1. **Obtain Recipient's Public Key:**
   - Use the recipient's public key (e,n)
2. **Convert Message to Integer:**
   - Convert the plaintext message M into an integer m such that 0 ≤ m < n.
3. **Encrypt Message:**
   - Compute the ciphertext c using the formula

$$c = m^e \mod n.$$

Decryption

1. **Obtain Private Key:**
   - Use the private key (d,n).
2. **Decrypt Message:**
   - Compute the plaintext integer m using the formula

$$m = c^d \mod n.$$

3. **Convert Integer to Message:**
   - Convert the integer mmm back to the plaintext message M.

RSA-Example

1

Key Generation:

- Choose $p = 61$ and $q = 53$.
- Compute $n = p \times q = 61 \times 53 = 3233$.
- Compute $\phi(n) = (61 - 1) \times (53 - 1) = 60 \times 52 = 3120$.
- Choose $e = 17$ (since it is a common choice and satisfies the condition $\gcd(17, 3120) = 1$).
- Compute d such that $d \times 17 \equiv 1 \mod 3120$, which gives $d = 2753$.

Public Key: $(17, 3233)$

Private Key: $(2753, 3233)$

2

Encryption:

- Convert message $M = "HELLO"$ to an integer m (let's assume $m = 65$ for simplicity).
- Compute ciphertext $c = 65^{17} \mod 3233 = 2790$.

3

Decryption:

- Compute plaintext $m = 2790^{2753} \mod 3233 = 65$.
- Convert integer 65 back to message $M = "HELLO"$.

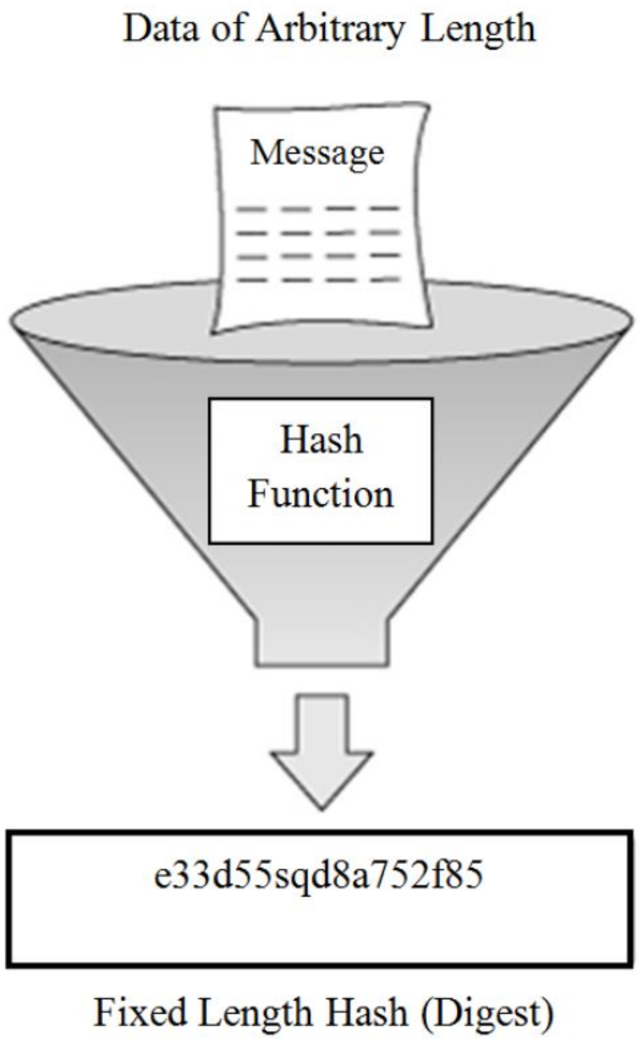
Introduction to Hash Functions

Definition: Converts input data of any size into a fixed-size hash value.

Properties:

- Deterministic
- Quick computation
- Pre-image resistance
- Small changes in input drastically change the hash
- Collision resistance

Examples: SHA-256, MD5

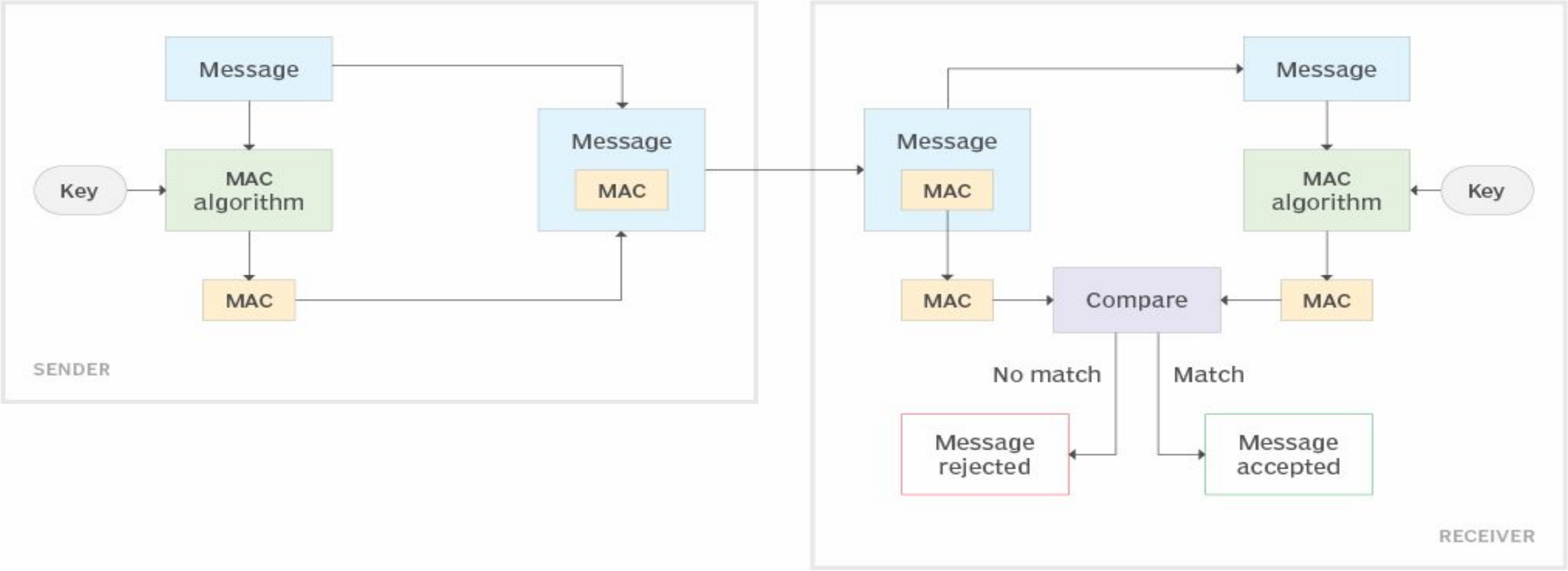


Message Authentication Code (MAC)

Definition: A short piece of information used to authenticate a message.

How It Works: Generated using a secret key and a cryptographic hash function.

Applications: Ensures data integrity and authenticity.

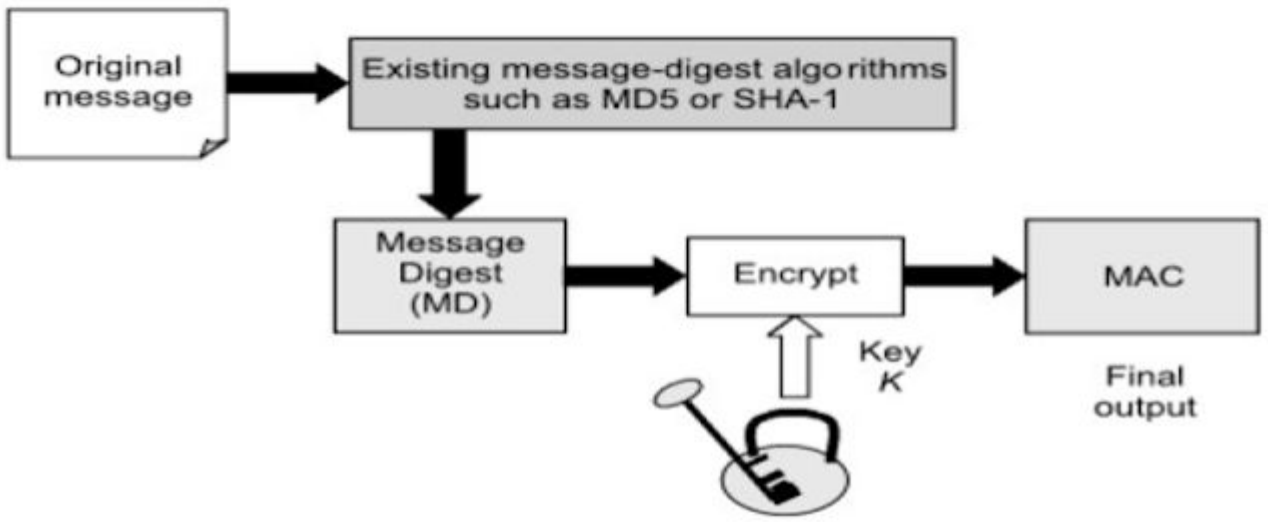


Hashed Message Authentication Code (HMAC)

Definition: A specific type of MAC involving a cryptographic hash function and a secret key.

How It Works:

- Combines a key with the message.
- Hashes the result.
- Hashes the result again with the key.



Examples: HMAC-SHA256, HMAC-MD5

Applications: Used in various internet protocols (e.g., SSL/TLS, IPsec)

Summary

- Recap of Key Points:
  - Symmetric and Asymmetric Key Cryptography
  - Diffie-Hellman Key Exchange & Man-in-the-Middle Attack
  - Setting up CA with OpenSSL
  - Stream Ciphers & Block Ciphers, RSA
  - Hash Functions, MAC, HMAC

Q&A

Reference

- Fortinet
  - https://www.fortinet.com/resources/cyberglossary/what-is-cryptography
- GeeksForGeeks
  - https://www.geeksforgeeks.org/
- Techtarget
  - https://www.techtarget.com/searchsecurity/
- OKTA
  - https://www.okta.com/identity-101/hmac/

Thank you