

Information Security

Prepared by Dr. Ahmed Omar

IT373

Information Security

Lecture 3



Asymmetric Cryptography & Key Management

Lecture 3

IT373

Information Security

Prepared by
Dr. Ahmed Omar

Agenda

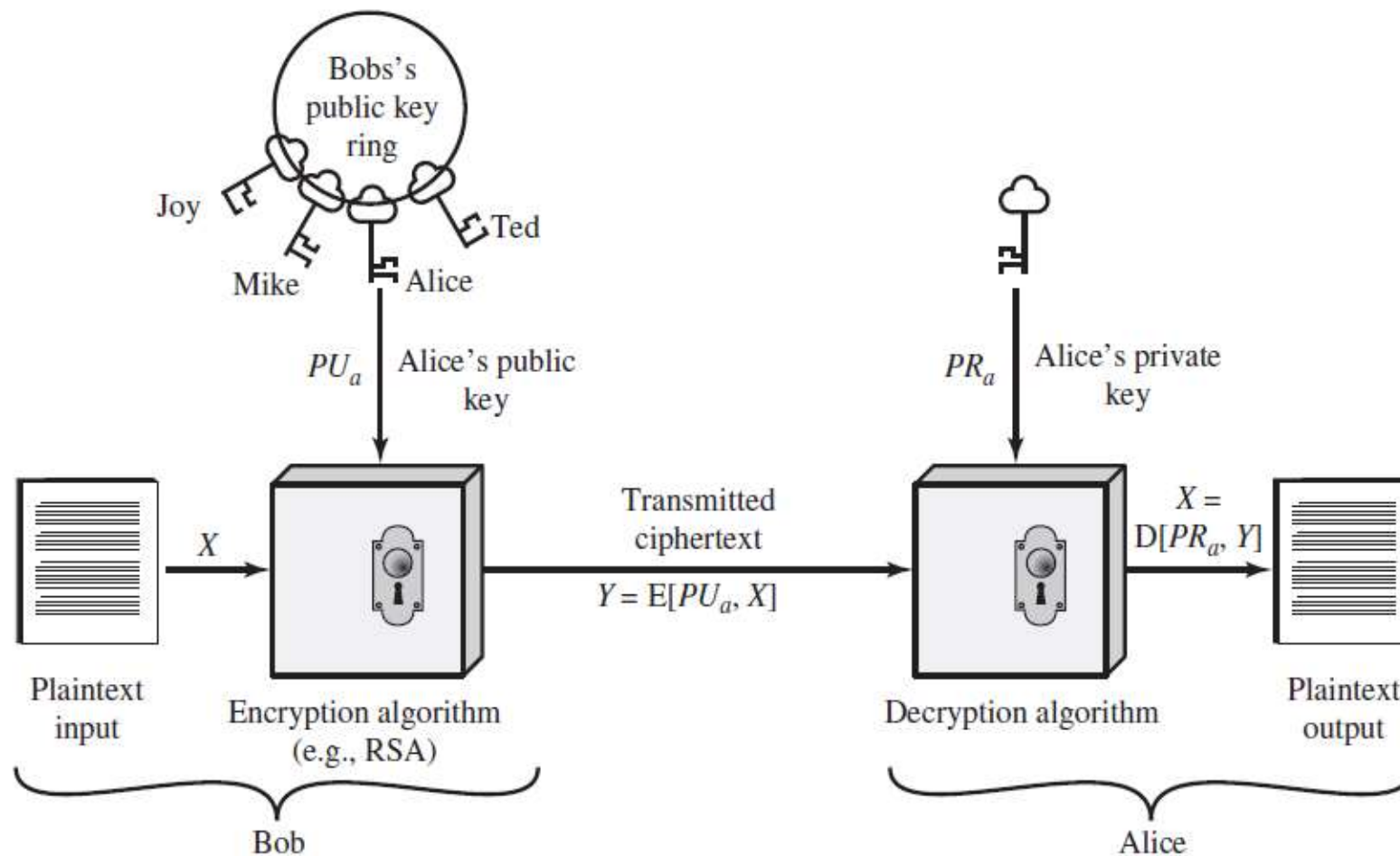
- ❑ Introduction to Asymmetric Cryptography
- ❑ Comparison with Symmetric Cryptography
- ❑ Public and Private Keys
- ❑ Common Asymmetric Encryption Algorithms
- ❑ Key Management Fundamentals
- ❑ End-to-End Encryption (E2EE)

Introduction to Asymmetric Cryptography

Asymmetric cryptography, also known as **public-key cryptography**, is a system that uses a pair of keys [a **public key** and a **private key**] for encryption and decryption.

- ❑ The main advantage is that the keys are not identical and do not need to be shared in advance.
- ❑ This approach enables **secure communication**, **digital signatures**, and **authentication** without requiring both parties to exchange a secret key beforehand

How Asymmetric Encryption Works



(a) Encryption with public key

How Asymmetric Encryption Works

❑ Encryption Process:

- Alice wants to send message to Bob
- Alice obtains Bob's public key
- Alice encrypts message with Bob's public key
- Only Bob can decrypt with his private key

❑ Key Security Principle:

- Even if public key is known, private key cannot be computed
- Security relies on mathematical problem difficulty

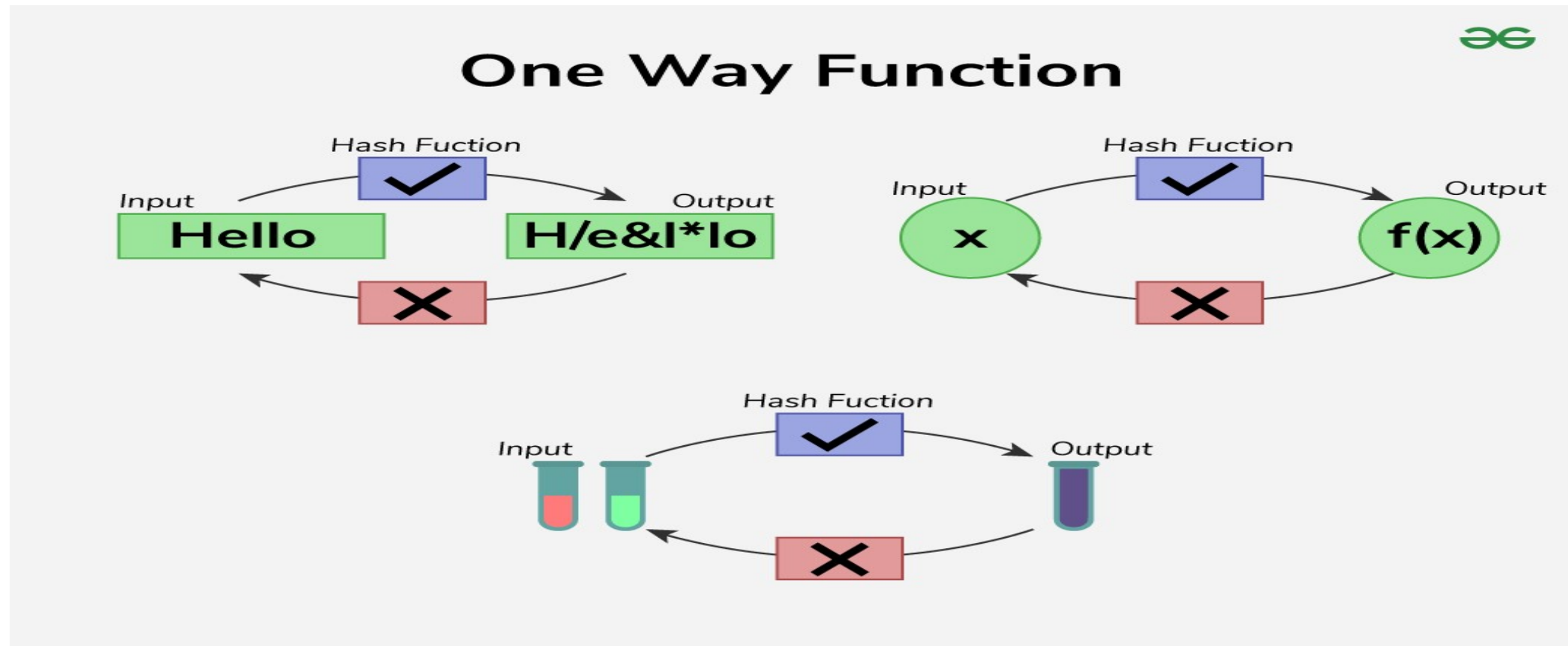
Comparison with Symmetric Cryptography

- ❑ In **symmetric cryptography**, the same key is used to encrypt and decrypt information. While it is **fast** and **efficient** for large data volumes, it suffers from the challenge of secure key distribution.
- ❑ **Asymmetric cryptography** addresses this problem by using key pairs: one for encryption and one for decryption. It enhances scalability, especially in environments with many users.

one-way function

- ❑ The security underpinning asymmetric cryptography is rooted in computationally hard mathematical problems, specifically the concept of one-way functions and, more particularly, trapdoor functions.

one-way function



Public and Private Keys

- ❑ The **public key** is intended to be distributed widely, while the **private key** must remain confidential. Messages encrypted with a public key can only be decrypted with the **matching** private key, and vice versa.
- ❑ For example, to send a confidential message to Bob, Alice encrypts it with Bob's public key. Only Bob can decrypt it using his private key.



PUBLIC
KEY



PRIVATE
KEY

Public and Private Keys

Public Key: Can be freely shared

- Used to encrypt data
- Used to verify digital signatures
- Cannot decrypt messages it encrypted

Private Key: Must remain secret

- Used to decrypt data
- Used to create digital signatures
- Mathematically linked to public key

Key Relationship:

- Mathematically related but computationally infeasible to derive one from the other



PUBLIC
KEY



PRIVATE
KEY

Core Principles

☐ Key Pair Generation:

- Mathematically related but distinct keys
- Public key: shared openly
- Private key: kept secret

Core Principles

☐ Functional Properties:

- Data **encrypted** with public key → decrypted with private key
- Data encrypted with private key → decrypted with public key
- Keys cannot be derived from each other

Common Asymmetric Encryption Algorithms

RSA (Rivest-Shamir-Adleman)

- Based on factoring large prime numbers
- Widely used for secure data transmission
- Key sizes: 1024, 2048, 4096 bits

ECC (Elliptic Curve Cryptography)

- Based on algebraic structure of elliptic curves
- Smaller key sizes than RSA for equivalent security
- More efficient for mobile and IoT devices

DSA (Digital Signature Algorithm)

- Used primarily for digital signatures
- Based on discrete logarithm problem
- Cannot be used for encryption/decryption

Key Management Fundamentals

- ❑ Key management refers to the processes and procedures involved in **generating, storing, distributing, and managing cryptographic keys.**
- ❑ **Importance:**
 - Ensures **confidentiality, integrity, and availability** of encrypted information.
 - Enables secure communication between parties.
 - Prevents single points of failure in security systems.
 - Protects against unauthorized access, loss, or compromise

Key Management Fundamentals

Key Management Aspects:

1. Distribution of public keys

- ☐ How public keys are made available to communicating parties.
- ☐ Methods to ensure authenticity of distributed keys.

2. Use of public-key encryption to distribute secrets

- ☐ Secure exchange of symmetric keys.
- ☐ Protection of sensitive information during transmission

Public Key Distribution Methods



Public Announcement

Broadcasting public keys to everyone

Weakness: Vulnerable to forgery



Publicly Available Directory

Keys stored in trusted directories

Weakness: Still vulnerable to tampering



Public Key Authority

Real-time access to secure directory

Strength: Tighter control over distribution



Public Key Certificates

Identity bound to public key via certificate

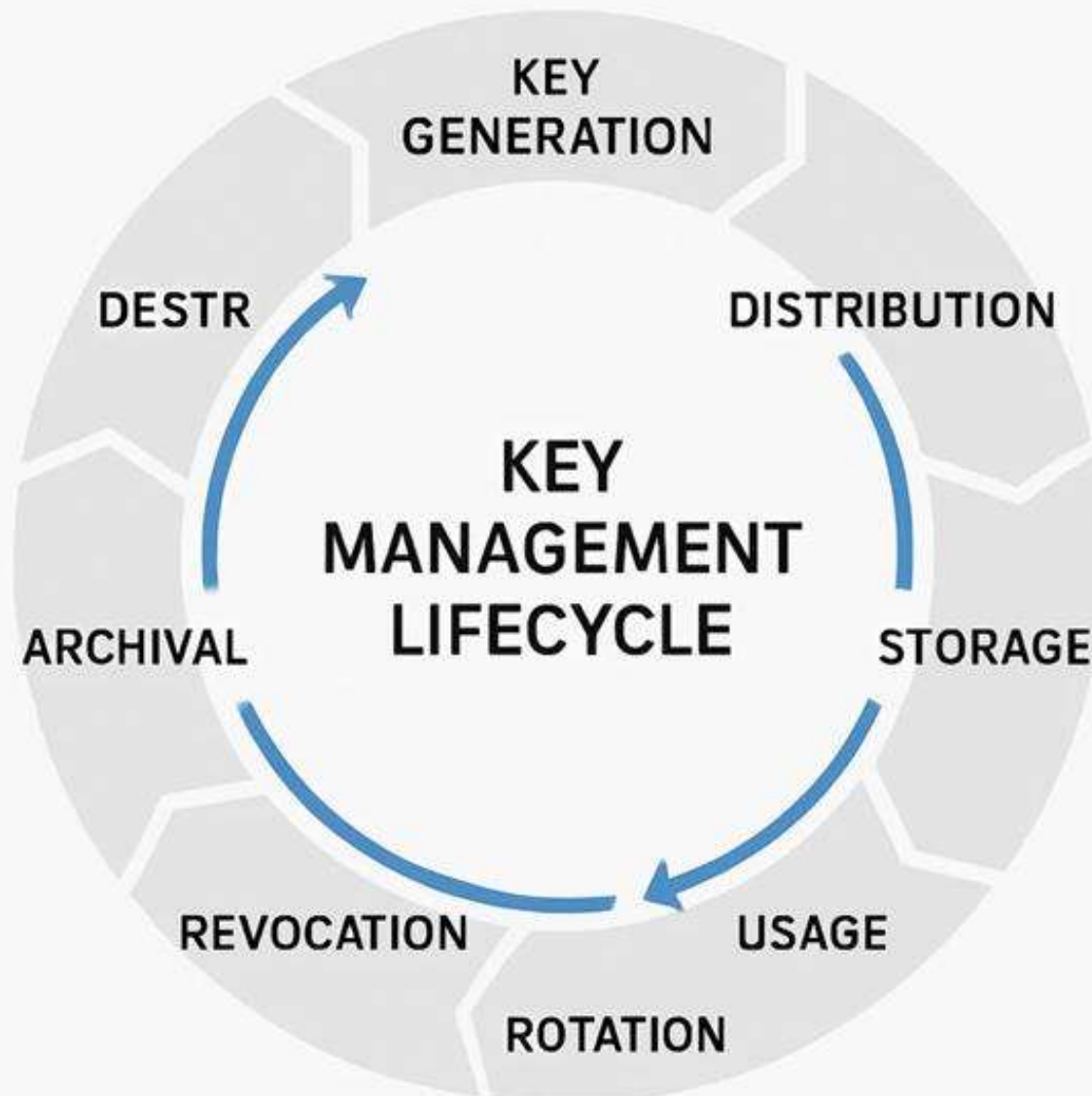
Strength: No real-time access needed

Public Key Infrastructure (PKI)

- ❑ A framework for **creating**, **distributing**, and **validating** public key certificates.

- ❑ **Key Components:**
 - Certificate Authority (CA): Issues and signs certificates
 - Registration Authorities (RAs): Verify identities.
 - Certificate Database : Stores issued certificates
 - Certificate Revocation List (CRL): Lists invalid certificates

Key Management Lifecycle



Key Management Lifecycle

1. **Key Generation:** Creating keys using secure algorithms.
2. **Key Distribution:** Securely sharing keys between parties
3. **Key Storage:** Protecting keys in secure storage systems
4. **Key Usage:** Using keys for encryption, decryption, signing
5. **Key Rotation:** Periodically updating keys to reduce risk
6. **Key Revocation:** Invalidating compromised or expired keys
7. **Key Archival:** Securely storing old keys for reference
8. **Key Destruction:** Securely erasing keys when no longer needed

Performance Considerations

❑ Computational Overhead:

- 100-1000 times slower than symmetric cryptography
- Key sizes much larger than symmetric keys
- Limited to small amounts of data for direct encryption

❑ Optimization Strategies:

- Hybrid systems for bulk encryption
- Hardware acceleration
- Algorithm selection based on use case

Advantages and Disadvantages

Asymmetric Cryptography Trade-offs

□ Advantages:

- Eliminates key distribution problem
- Enables digital signatures
- Supports non-repudiation
- Scales well for large networks

Advantages and Disadvantages

Asymmetric Cryptography Trade-offs

❑ Disadvantages:

- Computationally expensive
- Complex key management
- Requires trusted infrastructure

When to Use Asymmetric Cryptography

□ Primary Applications:

- Secure key exchange
- Digital signatures
- Authentication protocols
- Certificate-based systems

When to Use Asymmetric Cryptography

☐ Not Suitable For:

- Bulk data encryption
- Real-time communications requiring high speed
- Resource-constrained environments

RSA Algorithm

- ❑ **RSA** (Rivest-Shamir-Adleman) is a widely used public-key cryptographic algorithm. It's an asymmetric encryption method, meaning it uses a **pair of keys**: a public key for encryption and a private key for decryption. Its security relies on the practical difficulty of factoring the product of two very large prime numbers.

How RSA Works

1. Key Generation

- ❑ Choose two large prime numbers, p and q : These primes should be randomly chosen and kept secret. The larger the primes, the more secure the encryption.
- ❑ Compute the modulus n , $n=p \times q$, This value, n , is part of the public key.
- ❑ Calculate Euler's : $\varphi(n)=(p-1) \times (q-1)$
- ❑ Choose a public exponent e :
 - e must be an integer such that $1 < e < \varphi(n)$.
 - e must be coprime to $\varphi(n)$ (i.e., their greatest common divisor, GCD, is 1: $\text{GCD}(e, \varphi(n))=1$).
- ❑ Compute the private exponent d : Calculate d such that
 $d = e^{-1} \pmod{\varphi(n)}$.

Public Key = (n, e)

Private Key = (n, d)

2. Message Encryption

- ❑ First, **M** is converted into a **numerical representation** (e.g., using ASCII values). It's crucial that $0 \leq M < n$. If the message is larger than n , it's typically broken into smaller blocks.
- ❑ **Encrypt** the message M into ciphertext C using the formula:
$$C = M^e \pmod n$$

2. Message Decryption

□ Plaintext: $M = C^d \bmod n$

How RSA Works

1. Key Generation

- ❑ $p=3$.
- ❑ $q=7$
- ❑ $n=p \times q$ $n=3 \times 7=21$
- ❑ $\varphi(n)=(p-1) \times (q-1)$ $2 \times 6=12$
- ❑ $e=7$
- ❑ $d=7^{-1} \pmod{12}=7$

2. Encryption

- $M=4$
- $C=M^e \pmod{n}$ $C=4^7 \pmod{21} = 4$

3. Decryption

$M = C^d \pmod{n}$ $4^7 \pmod{21} = 4$

Elliptic Curve Cryptography (ECC)

Modern Alternative to RSA

❑ Mathematical Foundation:

- ✓ Based on elliptic curves over finite fields
- ✓ Equation: $y^2 = x^3 + ax + b$
- ✓ Security relies on elliptic curve discrete logarithm problem

Elliptic Curve Cryptography (ECC)

Advantages:

- ☐ Smaller key sizes for equivalent security
- ☐ Faster computations
- ☐ Lower power consumption
- ☐ Ideal for mobile and IoT devices

Choosing the Right Algorithm

❑ Factors to Consider:

- Security requirements
- Performance constraints
- Compliance requirements
- Implementation complexity

End-to-End Encryption (E2EE)

End-to-End Encryption (E2EE) is a secure communication model where only the sender and intended recipient(s) can read transmitted data. Third parties—including service providers, hackers, and governments—cannot access the decrypted content.

Key Principles

- **Encryption at Origin:**
Data is encrypted on the sender's device before transmission.
- **Decryption at Destination:**
Data is decrypted only on the recipient's device.
- **No Intermediary Access:**
Servers/routers handle encrypted data but cannot decrypt it.

End-to-End Encryption (E2EE)



End-to-End Encryption (E2EE)

Contrast with Other Models

Encryption Type	Who Can Access Data?	Example
End-to-End (E2EE)	Only sender & recipient	Signal, WhatsApp (chats)
Transport Encryption	Sender, recipient, and server	HTTPS (TLS), email (SMTP)
No Encryption	Anyone on the network	HTTP, plaintext SMS

End-to-End Encryption (E2EE)

Why E2EE Matters

1. Privacy:

- Prevents mass surveillance (e.g., by ISPs, governments, or the service itself).

2. Security:

- Mitigates server breaches (e.g., stolen data is useless without keys).

3. Trust Minimization:

- No need to trust service providers (e.g., Meta, Telegram).

Diffie–Hellman Ratchet

The "Diffie–Hellman Ratchet" is a crucial component of the **Double Ratchet Algorithm**, a key management algorithm used in secure messaging applications like **Signal**, **WhatsApp**, and **Facebook Messenger**. Its primary purpose is to enhance the security of ongoing conversations by frequently updating encryption keys, providing properties like forward secrecy and post-compromise security.

Summary

- Asymmetric Cryptography Basics
- Key Asymmetric Algorithms
- Key Management Fundamentals
- Key Management Lifecycle
- End-to-End Encryption (E2EE)



Thank You

