Nishita Lath
2101CS53

Lab 2 - Asymmetric Encryption

→ **Introduction**: Asymmetric Encryption, also known as public key encryption, is a revolutionary method in cryptography that uses a pair of keys: a public key for encryption and a private key for decryption. Unlike symmetric encryption, these keys are mathematically related but distinct, allowing secure communication without the need for shared keys.

1. **Core Mathematical Principles behind asymmetric Encryption**

a.) key Pair Generation:- Asymmetric encryption relies on generating a pair of keys: public key (shared openly for encryption) and private key (kept secret for decryption)

b) Prime Numbers and Modular arithmetic :- The security of many asymmetric algorithms: such as RSA, relies on the difficulty of certain mathematical problems.

- Integer Factorization Problem - Factoring Large Numbers into primes is computationally infeasible.
- Discrete logarithm problem: Used in algorithms like Diffie - Hellman & ElGamal.

c.) RSA Encryption formula :- Encryption with a public key

$$c = p^e \bmod n$$

p = plaintext, e = Public exponent, n: Product of two large primes.

Decryption with private key.

$$p = c^d \bmod n$$

d: Private exponent divided from e, p & q.

d) Elliptic Curve Cryptography (ECC): ECC relies on the mathematical properties of elliptic curves over finite fields

$$y^2 = x^3 + ax + b.$$

The difficulty of solving the elliptic curve discrete logarithm problem ensures safety.

2. Key Management Challenges

a) Key Pair Distribution: While the public key can be shared openly, Ensuring its authenticity is critical to prevent man-in-the-middle attacks.

b) Private Key Security: The private key must be securely stored to avoid compromise. Loss of private key results in data being unrecoverable.

c) Scalability: Unlike symmetric encryption, key management is simpler as each user only requires one key pair. However, managing digital certificates can still be complex.

d) Certificate Authorities (CAs):- Public Key Infrastructure (PKI) depends on trusted certificate authorities to verify public keys, adding administrative overhead.

3. Performance Characteristics

a) Speed: Asymmetric encryption is computationally intensive compared to symmetric encryption due to complex mathematical operations.

b) Resource Usage: High CPU and memory usage makes asymmetric encryption less suited for resource constrained environments like IoT devices.

c) Hybrid Systems: Many systems combine asymmetric and symmetric encryption for overall better performance.

Example: TLS / SSL protocols.

4. Security strengths & vulnerabilities :

a.) Strengths :
- No key sharing - It eliminates the need for securely sharing a secret key.
- Scalability : suitable for large-scale systems with many users.
- Digital signatures - Supports authentication and non-repudiation.

b.) Vulnerabilities :
- Public Key compromises loss or theft of a private key compromises security.
- Man-in-the-middle Attack : If the public key is tampered, encryption can be bypassed.
- Quantum Computing Threats : Algorithms like RSA & ECC may become insecure in the future due to quantum computing

5. Real World Application & Use Cases :

a.) Secure Communication : Protocols like TLS/SSL use asymmetric encryption to secure internet communication
Email Encryption PGP (Pretty Good Privacy) relies on asymmetric key.

b.) Digital Signatures : Asymmetric encryption verifies the authenticity & integrity of documents, contracts, & software (eg in blockchain systems).

c.) Authentication : Used in multifactor authentication systems & SSH connections to ensure secure access.

d) Cryptocurrency - Cryptographic wallets use asymmetric encryption to manage private & public keys for secure transactions.