SNS COLLEGE OF ENGINEERING
Kurumbapalayam (Po), Coimbatore – 641 107
AN AUTONOMOUS INSTITUTION
Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai

DEPARTMENT OF CSE (IOT & CS INCLUDING BCT)

Public Key Cryptography Algorithms: RSA and Diffie-Hellman

1. RSA Algorithm

The **RSA (Rivest-Shamir-Adleman)** algorithm is one of the most widely used public key cryptosystems. It enables both **encryption** and **digital signatures**. RSA's security relies on the **difficulty of factoring large numbers**.

Key Generation:

1. **Select two large prime numbers** ppp and qqq.
2. **Compute the modulus** n=p×qn = p \times qn=p×q. This modulus is used as part of the public key and is also used for encryption and decryption.
3. **Calculate the Euler's totient function** φ(n)=(p−1)(q−1)\varphi(n) = (p-1)(q-1)φ(n)=(p−1)(q−1). This function is important in the creation of the private and public exponents.
4. **Choose an encryption exponent** eee such that 1<e<φ(n)1 < e < \varphi(n)1<e<φ(n), and eee is coprime with φ(n)\varphi(n)φ(n) (i.e., the greatest common divisor of eee and φ(n)\varphi(n)φ(n) is 1). Common values for eee are 3 or 65537.
5. **Calculate the private exponent** ddd, which is the modular inverse of eee modulo φ(n)\varphi(n)φ(n). This means e×d≡1 (mod φ(n))e \times d \equiv 1 \ (\text{mod} \ \varphi(n))e×d≡1 (mod φ(n)).

The **public key** consists of (n,e)(n, e)(n,e), and the **private key** consists of (n,d)(n, d)(n,d).

Encryption and Decryption:

- **Encryption**: Given a plaintext message MMM, it is converted into an integer mmm such that 0≤m<n0 \leq m < n0≤m<n. The ciphertext CCC is then computed as: C=memod nC = m^e \mod nC=memodn
- **Decryption**: The recipient, who knows the private key (n,d)(n, d)(n,d), can decrypt the ciphertext CCC as follows: m=Cdmod nm = C^d \mod nm=Cdmodn The plaintext message MMM is then obtained.

Digital Signatures:

- To **sign** a message, the sender computes the signature SSS by applying their private key to the message hash:

  S=H(M)dmod nS = H(M)^d \mod nS=H(M)dmodn

  where H(M)H(M)H(M) is a cryptographic hash of the message MMM.

- To **verify** the signature, the receiver computes:

  H(M)=Semod nH(M) = S^e \mod nH(M)=Semodn

  If this matches the hash of the received message, the signature is valid.

Security:

The security of RSA is based on the **factoring problem**: the difficulty of factoring large composite numbers into their prime factors. The larger the modulus nnn, the stronger the encryption.

---

2. Diffie-Hellman Key Exchange

The **Diffie-Hellman key exchange** algorithm allows two parties to securely exchange a shared secret key over an insecure channel. It is widely used in protocols such as SSL/TLS.

Key Exchange Process:

1. **Choose a large prime number ppp** and a **primitive root ggg** modulo ppp, which is commonly shared publicly by both parties. Both ppp and ggg are public values.
2. Each party generates a private key:
   - **Alice** chooses a private key aaa (randomly selected).
   - **Bob** chooses a private key bbb (randomly selected).
3. Each party computes their **public key** using their private key:
   - **Alice** computes her public key A=gamod pA = g^a \mod pA=gamodp.
   - **Bob** computes his public key B=gbmod pB = g^b \mod pB=gbmodp.
4. The two parties exchange their public keys AAA and BBB over the insecure channel.
5. Now, each party can compute the **shared secret**:
   - **Alice** computes the shared secret as s=Bamod ps = B^a \mod ps=Bamodp.
   - **Bob** computes the shared secret as s=Abmod ps = A^b \mod ps=Abmodp.

   Both Alice and Bob now share the same secret value sss, which can be used as a symmetric key for further encryption and communication.

Security:

The security of Diffie-Hellman relies on the **discrete logarithm problem**: given ggg, ppp, and gxmod pg^x \mod pgxmodp, it is computationally infeasible to determine xxx. This ensures that even if an attacker intercepts the public keys, they cannot compute the shared secret without solving the discrete logarithm problem.

Key Distribution in Public Key Cryptosystems

Key distribution is a fundamental aspect of public key cryptography. It refers to how public keys are securely shared between parties, ensuring that the public key being used is the correct one for the intended recipient.

1. Public Key Infrastructure (PKI)

PKI is a system that uses **digital certificates** and **Certificate Authorities (CAs)** to manage key distribution. PKI ensures that public keys are authentic and can be trusted by all parties.

- **Certificate Authorities (CAs)**: A CA is a trusted entity that verifies the identity of the key owner and signs their public key with a digital signature. This creates a **digital certificate** that associates the public key with the identity of the key holder. Examples of CAs include Let's Encrypt, DigiCert, and GlobalSign.
- **Digital Certificates**: These certificates contain the public key, the identity of the entity, the CA's signature, and other metadata. The certificate is signed by the CA, making it trustworthy. The recipient can verify the authenticity of the certificate by checking the CA's signature.
- **Revocation**: Public keys and certificates can be revoked if compromised. This is usually done through **Certificate Revocation Lists (CRLs)** or the **Online Certificate Status Protocol (OCSP)**.

2. Web of Trust (WoT)

The **Web of Trust** is a decentralized model of key distribution, most famously used in **Pretty Good Privacy (PGP)**. It does not rely on central authorities but rather on mutual trust between users.

- In WoT, users directly exchange public keys with each other and verify the authenticity of each other's keys. If User A trusts User

B’s key, they can sign it, and other users who trust User A can also trust User B's key.
- Trust is based on personal relationships and verified signatures from other trusted individuals.

3. Direct Key Exchange

In some cases, public keys are exchanged through **trusted channels** (e.g., in person, via a secure website, or through a private messaging system). This avoids the need for a CA or centralized authority.

- **Physical Exchange**: Two parties can exchange their public keys directly, for instance, by scanning QR codes that contain the key or using secure channels like encrypted email.
- **Secure Websites**: Websites often use **SSL/TLS certificates** to securely transmit public keys. The server’s public key is included in the certificate, which is signed by a CA, ensuring its authenticity.

4. Key Agreement Protocols

In scenarios where no pre-shared keys exist, and key distribution is needed for symmetric encryption, **key exchange algorithms** like Diffie-Hellman or **Elliptic Curve Diffie-Hellman (ECDH)** can be used to securely exchange symmetric keys over an insecure channel.

5. Hybrid Cryptosystems

A hybrid cryptosystem combines asymmetric and symmetric cryptography. For instance, in **TLS/SSL**:

- Asymmetric cryptography (e.g., RSA or Diffie-Hellman) is used to exchange a symmetric key securely.
- Once the symmetric key is agreed upon, symmetric encryption (e.g., AES) is used for efficient data encryption.