UNIT 3 :Public Key Cryptosystems:

LIST OF TOPIC

Principles of public key cryptosystems: Public-key cryptosystems, Applications for Public-key cryptosystems, Requirements for Public-KeyCryptography, Public-Key Cryptanalysis, RSA Algorithm, The Security of RSA, Key management.

Principles of public key cryptosystems:

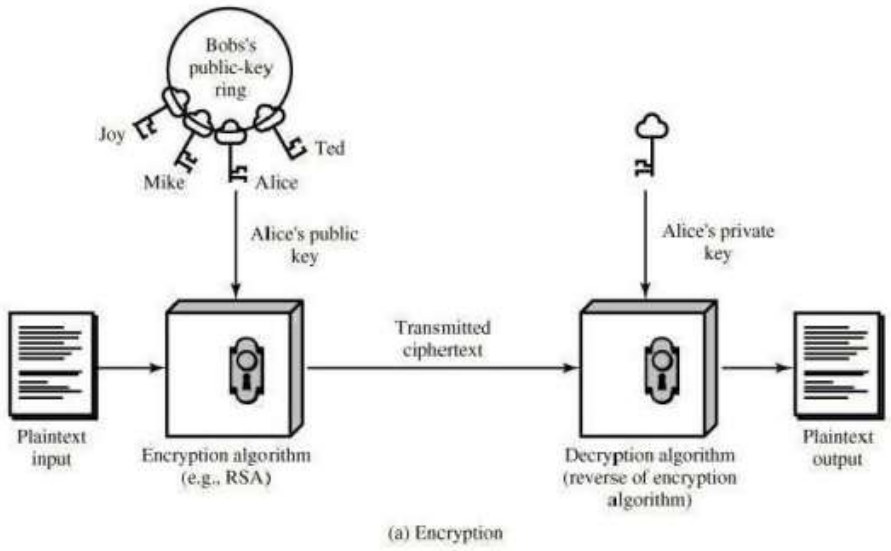
Public key cryptosystems, also known as asymmetric cryptosystems, rely on the use of two distinct but mathematically linked keys: a public key and a private key. Here are the key principles behind them:

1. **Key Pair Generation**: Each user generates a pair of keys: a public key, which is shared with everyone, and a private key, which is kept secret. The public key is used for encryption or verification, while the private key is used for decryption or signing.
2. **Encryption and Decryption**: Data encrypted with a public key can only be decrypted by the corresponding private key. This ensures that only the intended recipient, who possesses the private key, can decrypt the data.
3. **Digital Signatures**: A sender can use their private key to create a digital signature for a message. This signature can be verified by anyone who has access to the sender's public key, confirming the authenticity and integrity of the message.
4. **Authentication**: Public key cryptography can be used to verify the identity of a user. When a message is signed with a private key, anyone can verify the signature using the public key, ensuring that the message came from the claimed sender.
5. **Key Exchange**: Public key cryptosystems often include methods for securely exchanging symmetric keys. For instance, protocols like RSA or Diffie-Hellman can be used to agree on a shared symmetric key over an insecure channel.
6. **Mathematical Foundations**: The security of public key cryptosystems is based on mathematical problems that are computationally hard to solve. For example, RSA relies on the difficulty of factoring large composite numbers, while Elliptic Curve Cryptography (ECC) relies on the difficulty of solving the elliptic curve discrete logarithm problem.
7. **One-Way Functions**: Public key cryptosystems often use one-way functions, which are easy to compute in one direction but hard to invert. This ensures that while encryption is straightforward, decryption (without the private key) is infeasible.
8. **Security Assumptions**: The security of these systems depends on certain assumptions about the hardness of specific computational problems. As computational capabilities and mathematical techniques evolve, cryptographic protocols need to be updated to maintain security.

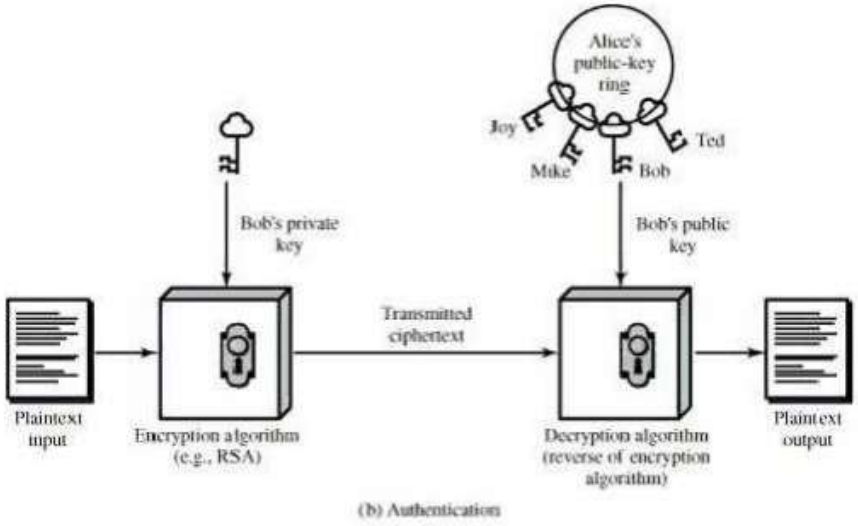
Define Public Key Cryptography

Public-key cryptography is a cryptographic system that uses two separate keys, one of which is secret and the other one is public. The algorithms used for public key cryptography are functions.

Public Key Cryptosystem



(a) Encryption



(b) Authentication

- A public-key encryption scheme has six parts.

o **Plaintext:** This is the readable message or data that is fed into the algorithm as input.

o **Encryption algorithm:** The encryption algorithm performs various transformations on the plaintext.

o **Public and private keys:** This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.

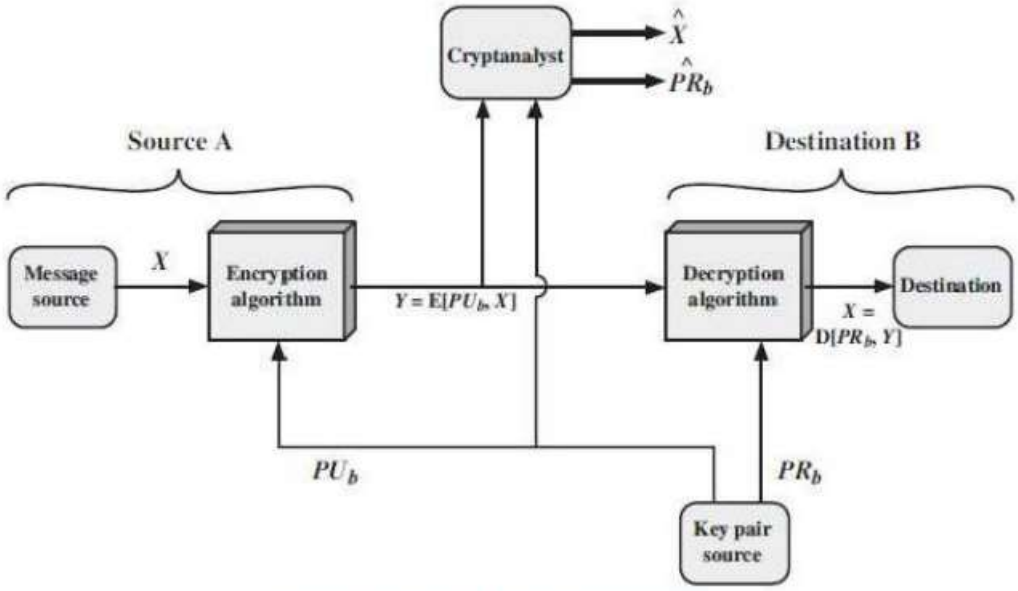
o **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the key

o **Decryption algorithm**: This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

- Any cryptosystem are designed to meet following goal 1. Secrecy (Encryption) 2. Authentication

- Now we will discuss how it is maintain in public key cryptosystem

Public Key Cryptosystem: Secrecy (Confidentiality)



Encryption using public key cryptography

- The essential steps are the following.
  - Each user generates a pair of keys to be used for the encryption and decryption of messages.
  - Each user places one of the two keys in a public register or other accessible file. This is the public key. The other key is kept private.
  - If A wishes to send a confidential message to B, A encrypts the message using B's public key.
  - When B receives the message, it decrypts it using the private key. No other recipient can decrypt the message because only B knows B's private key.
  - As long as a user's private key remains protected and secret, incoming communication is secure.
  - At any time, a system can change its private key and publish the companion public key to replace its old public key.
- Suppose there is some source A that produces a message in plaintext, $X = [X_1, X_2, \ldots, X_M]$ and sends it to B.
- B generates a related pair of keys: a public key, PU_b, and a private key, PR_b. $PU_b$is publicly available and therefore accessible by A.
- With the message X and the encryption key PU_b as input, A forms the ciphertext $Y = [Y_1, Y_2, \ldots, Y_N]$:

$$Y = E(PU_b, X)$$

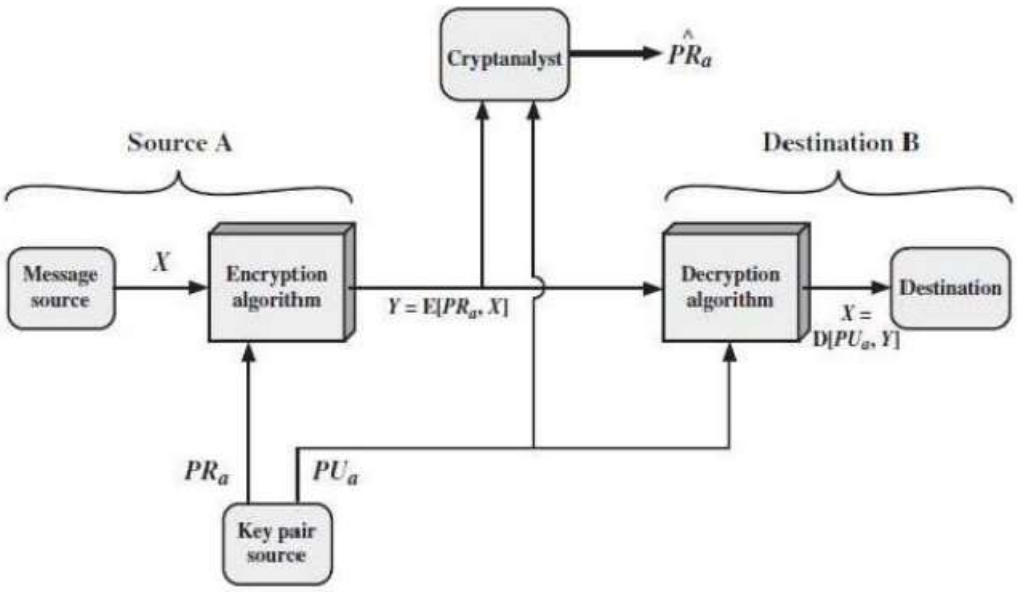
- The intended receiver, having the matching private key, is able to decrypt the message:

$$X = D(PR_b, Y)$$

- An adversary, observing Y and having access to Pu_bonly, may attempt to recover X and/or PR_b.

- If the adversary is interested only in this particular message, then the focus of effort is to recover X by generating a plaintext estimate. $\hat{X}$
- Whereas if the adversary is interested in being able to read future messages as well, then he attempts to recover PR_b by generating an estimate $P\hat{R}_b$

Public Key Cryptosystem: Authentication



Authentication using public key cryptography

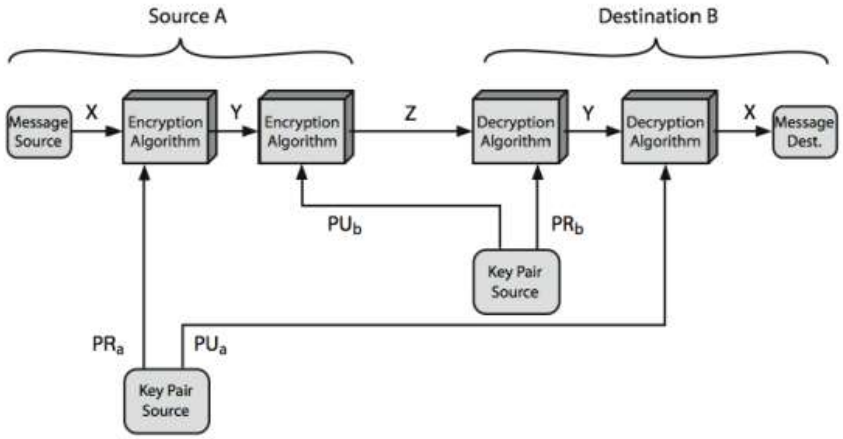
- However, the above scheme does not provide authentication of sender as, anyone having access to the public key can encrypt the message.
- Public-key encryption can be used to provide authentication in the following manner:
  - When A wishes to send a message to B where confidentiality is not needed but authentication is required, A encrypts the message using PR_a.
  - Anyone having access to PU_a can decrypt the message. However, one thing is sure that the message originated from A since no one except A could have encrypted the message using PR_a.
- A prepares a message to B and encrypts it using A's private key before transmitting it.

  Y = E (PRa, X)
- B can decrypt the message using A's public key.

  X = D (PUa, Y)
- Because the message was encrypted using A's private key, only A could have prepared the message. Therefore, the entire encrypted message serves as a digital signature.
- In addition, it is impossible to alter the message without access to A's private key, so the message is authenticated both in terms of source and in terms of data integrity.
- However, the entire message needs to be stored to bring up in case of dispute.
- A more efficient way of achieving the same results is to encrypt a small block of bits that is a function of the document.
- Such a block, called an authenticator.
- It must have the property that it is infeasible to change the document without changing the authenticator.

- If the authenticator is encrypted with the sender's private key, it serves as a signature.

Public Key Cryptosystem: Authentication and Secrecy



- Authentication and Secrecy both can be achieved by combining above both techniques.
  - First sender A encrypt message X with private key of A.

    Y = E (PRa, X)
  - Then again A encrypt Y with public key of B.

    Z = E (PUb, Y)
  - Then send Z.
  - Only B can decrypt Z as it is encrypted with public key of B. So it gives Secrecy.

    Y = D (PRb, Z)
  - Now Y can be decrypted with public key of A. So it gives authentication.

    X = D (PUa, Y)
- So by using public key cryptography we can achieve secrecy and authentication.

Applications of Public Key Cryptography

- Applications of public-key cryptosystems can classified into three categories:
  1. **Encryption /decryption:** The sender encrypts a message with the recipient's public key.
  2. **Digital signature:** The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.
  3. **Key exchange:** Two sides cooperate to exchange a session key. Several different methods are possible.

Brute force attack

- This attack includes trying all the alternate keys until the correct key is found.
- Counter measure to this is use large keys.
- However, public-key systems depend on the use of some sort of invertible mathematical function which is really time-consuming and increases overhead.
- Thus, there is a tradeoff. The key size must be large enough to make brute-force attack impractical but small enough for practical encryption and decryption.
- Secure keys are long enough to make encryption decryption really slow.

Computation of private key from public key

- In this attack, some characteristics of algorithm are exploited to calculate the private key from public key.
- This attack needs many known or chosen plaintext-ciphertext pairs.
- To date it has not been mathematically proven that this form of attack is infeasible for a particular algorithm. Thus any given algorithm is suspect.

Probable message attack

- In this attack, the opponent has some idea about the plaintext and he uses this information to find the private key.
- Suppose that a message consist only a 56-bit DES key.
- An adversary could encrypt all possible 56-bit DES keys using the public key and could discover the encrypted key by matching the transmitted ciphertext.
- Thus, no matter how large the key size of the public-key scheme, the attack is reduced to a brute-force attack on a 56-bitkey.
- This attack can be prevented by appending some random bits to such simple messages.

The RSA Algorithm

- RSA algorithm processes plaintext blocks, with each block having a binary value less than some number n.
- The block size must be less than or equal to $\log_2(n) + 1$.
- Steps for RSA:
  - Select two large prime numbers **p** and **q.**
  - Calculate n = pq.
  - Calculate $\phi(n) = (p - 1)(q - 1)$.
  - Select e such that e is relatively prime to $\phi(n)$.
    $\text{Gcd}(\phi(n),e) = 1, 1 < e < \phi(n)$
  - Compute d such that $d*e \equiv 1 \pmod{\phi(n)}$.
- RSA is a public key algorithm with public key PU = {e, n} and private key PR = {d, n}.
- Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C:

$$C = M^e \bmod n$$

$$M = C^d \bmod n$$

$$\mathbf{M= (M^e)^d \bmod n}$$

- For the above equation to be true, d must be an inverse of e.
- D can be calculated from e using extended Euclid's algorithm.
- Both sender and receiver must know the value of n.

- The sender knows the value of e, and only the receiver knows the value of d.
- RSA can also be subjected to various attacks like brute-force attack, various mathematical attacks, timing attacks and chosen ciphertext attacks.
- Some of these attacks exploit the mathematical characteristics of RSA.

Security Aspects of RSA

RSA's security hinges on several key factors:

1. **Difficulty of Factoring Large Integers**:
   - The primary security of RSA comes from the fact that while nnn is known (since it's part of the public key), it is extremely difficult to factorize nnn back into ppp and qqq. This process is known as the integer factorization problem, which has no known efficient (polynomial-time) solution for sufficiently large numbers.
2. **Key Size**:
   - The security of RSA is proportional to the size of nnn. Typically, 2048-bit or 4096-bit keys are used today, as smaller keys can be vulnerable to factorization attacks using modern computing power.
3. **Attacks on RSA**:
   - **Mathematical Attacks**: These include trying to factorize nnn or find ddd without knowing ppp and qqq. Advanced algorithms like the General Number

Field Sieve (GNFS) are the most effective for factorization but still require immense computational resources for large keys.

- **Timing Attacks**: RSA operations can sometimes leak information based on the time taken to execute the decryption or signature operations. These attacks exploit variations in execution time to deduce the private key. To counteract this, constant-time algorithms or blinding techniques are used.
- **Padding Attacks**: Improper padding can lead to vulnerabilities like the Bleichenbacher attack. Modern RSA implementations use secure padding schemes like Optimal Asymmetric Encryption Padding (OAEP).

4. **Cryptographic Best Practices**:
   - **Key Generation**: Use strong random number generators to ensure the primes ppp and qqq are large and random.
   - **Padding Schemes**: Implement strong padding techniques such as OAEP to prevent various attacks.
   - **Key Management**: Protect private keys with strong access controls and encryption to prevent unauthorized access.
   - **Regular Key Rotation**: Periodically change keys to limit the damage if a key is compromised.

Key management

Key management is a critical aspect of public key cryptography, ensuring the secure handling of cryptographic keys throughout their lifecycle. These keys, essentially secret codes, are used to encrypt and decrypt sensitive data, ensuring its confidentiality and integrity. The process involves generating, storing, distributing, and managing these keys to prevent unauthorized access or loss, which could compromise the entire security system.

The importance of effective key management cannot be overstated. It's the cornerstone of secure communication and data protection in the digital age. Without proper key management, the security of encrypted information is at risk, leaving digital assets vulnerable to cyber threats.

Key management encompasses various aspects, including the distribution of public keys and the use of public-key encryption to distribute secrets. Public keys can be distributed through various methods, such as public announcement, publicly available directories, public-key authorities, and public-key certificates. Each method has its own advantages and disadvantages, with public announcement being the most vulnerable to forgery.

Public key cryptography relies on the use of key pairs, consisting of a public key and a private key. The public key can be shared freely, while the private key must be kept secret. This system allows for secure communication, as only the holder of the private key can decrypt messages encrypted with the corresponding public key.

The secure management of private keys is paramount. If a private key is compromised, an unauthorized individual could gain access to sensitive data or impersonate the legitimate owner. This highlights the importance of robust key management practices, including secure storage, access control, and regular key rotation.

Key management is a complex process that requires careful planning and implementation. It involves establishing clear policies and procedures for key generation, distribution, storage, and destruction. Organizations must also consider the potential risks of key compromise and develop strategies for recovery and zeroization.

Effective key management is essential for maintaining the security of digital assets and protecting sensitive information from unauthorized access. It requires a comprehensive approach that addresses all aspects of the key lifecycle, from generation to destruction. By implementing robust key management practices, organizations can significantly enhance their cybersecurity posture and mitigate the risks of data breaches.

Assignment 3

1. Explain the basic concept of a public-key cryptosystem and how it differs from symmetric-key cryptography.
2. Describe the general structure of a public-key cryptosystem.
3. What are the main components of a public-key cryptosystem, and how do they interact?
4. Explain how public key cryptography ensures confidentiality during data transmission.
5. Explain how public key cryptography ensures confidentiality during data transmission.
6. Explain how public key cryptography ensures confidentiality and authentication during data transmission.
7. Explain the application of public-key cryptosystems in detail.
8. What is public-key cryptanalysis? Describe some common methods used in cryptanalysis of public-key cryptosystems.
9. Describe the RSA algorithm, including key generation, encryption, and decryption processes.
10. Explain the RSA Algorithm
11. Perform RSA encryption and decryption for a message M=15 using p=7,q=11 and e=5
12. Perform RSA encryption and decryption for a message M=11 using p=5,q=11 and e=7
13. Perform RSA encryption and decryption for a message M=11 using p=17,q=23 and e=3
14. Explain The Security of RSA in detail
15. Explain key management in detail.