Module - 3
Public key Encryption

Asymmetric keys → two related keys, a public key & a private key that are used to perform complementary operations such as encryption & decryption or signature generation & signature verification

public-key cryptography is developed to address two key issues

① key distribution: how to have secure communication in general without having to trust a KDC with your eyes

② Digital signatures: how to verify a message comes intact from the claimed sender

Public-key Cryptography

a public key / two-key / asymmetric cryptography involves the use of two keys:

→ a public-key, which may be known by anybody, and can be used to encrypt messages & verify signatures

→ a private key known only to the recipient, used to decrypt messages, and sign (create) signature

→ infeasible to determine private key from public

→ this is called asymmetric because those who encrypt messages or verify signatures cannot decrypt message or create signatures

Publickey encryption has 6 integredients

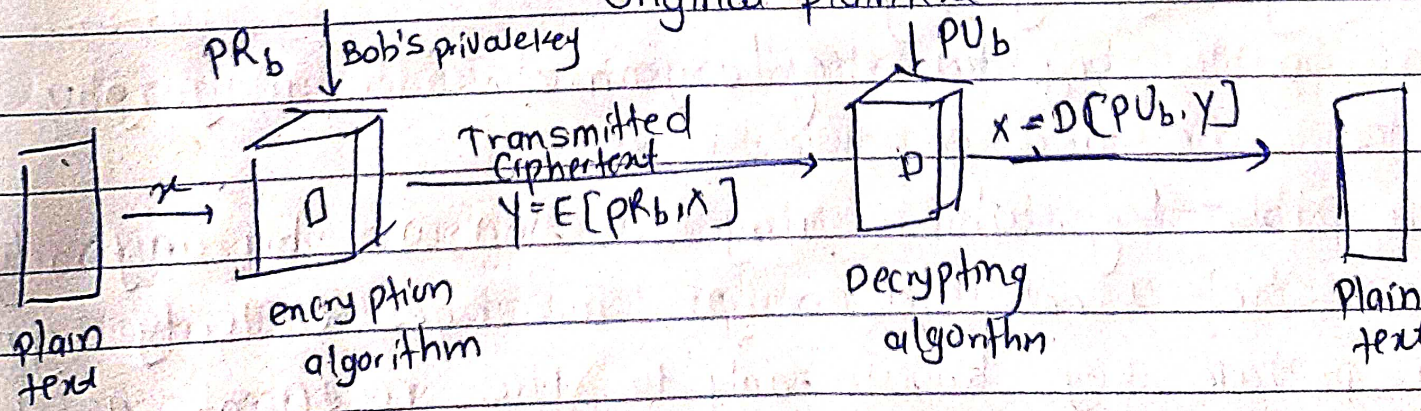
① Plain Text - readable message or data that is fed into the algorithm as input.

② Encryption algorithm - performs various transformations on the plain text

③ public & private keys - pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.

④ Cipher text - encrypted message produced as output. depends on plaintext & key. for a given message two keys produce two different ciphertext

⑤ Decryption algorithm - accepts the ciphertext & the matching key, produces the Original plaintext.



→ each user generates a pair of keys to be used for the encryption & decryption of messages

→ each user places one of the two keys in a public register or other accessible file this is public key. the companion key is kept private. each of the user maintains a collections of public keys obtained from others.

→ if bob wishes to send a confidential message to alice, bob encrypts the message using alice's public key

→ when alice receives the message, she decrypts it using her private key

→ For some source A that produce a message in plaintext $X = [x_1, x_2 \ldots x_M]$ message is intended for destination B. B generates a related of keys, public key PU_b and a private key PR_b.

with message X & encryption key PU_b as i/p forms the ciphertext $Y = [y_1, y_2 \ldots y_N]$:

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

the intended receiver, in possession of the matching private key is able to invert the transformation

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

Applications of public key

public-key systems are characterized by the use of cryptographic algorithm with two keys, one held private & one available publicly.

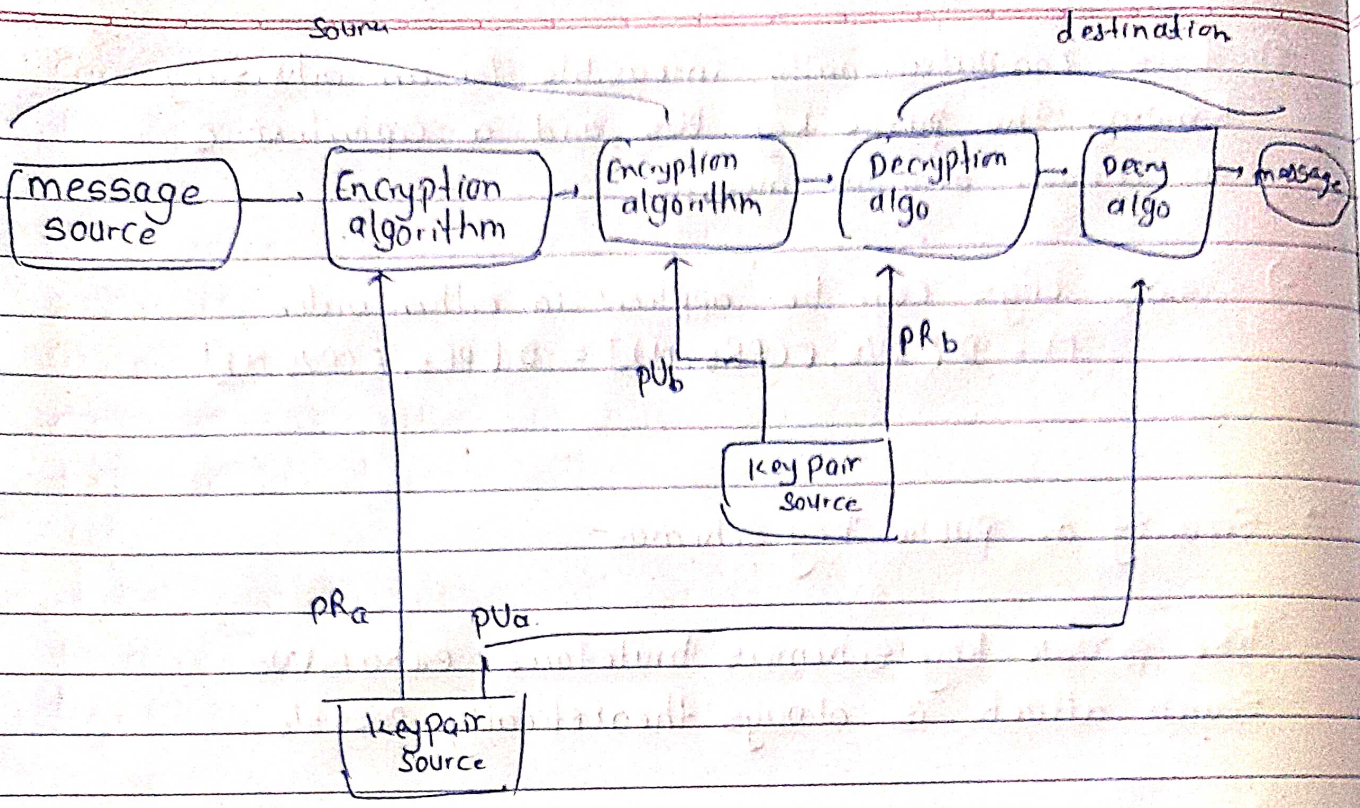
Depending on the application the sender uses either the sender's private key or receiver's public key or both to perform some type of cryptographic functions

Classification of public-key cryptosystem

① encryption/decryption: sender encrypts a message with the recipient's public key and recipients decrypts the msg with it's private key (provide secrecy)

② Digital signature: 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 which is a secret key for symmetric encryption generated for use for a particular transaction (or session) and valid for short period of time



Requirements for public-key Cryptography

① It is computationally easy for a party B to generate a key pair (public key PU_b, private key PR_b)

② it is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext

$$C = E(PU_b, M)$$

③ it is computationally easy for a receiver B to decrypt the resulting ciphertext using a private key to recover original message

$$M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$$

④ it is computationally infeasible for an adversary, knowing the public key, PU_b to determine the private key PR_b

⑤ it is computationally infeasible for an adversary, knowing the public key, PU_b and a ciphertext C to recover original message.

⑥ two keys can be applied in either order

$$M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$$

Security of public key schemas

like private key schemas bruteforce exhaustive search attack is always theoretically possible

keys used are too large (>512 bits)

Security relies on a large enough difference in difficulty between easy (en/decrypt) and hard (cryptonalyze) problems

RSA

The algorithm uses large int (1024)

security due to cost of factoring large no.

factorization takes $O(e^{\log n \log\log n})$ operation

RSA key setup

each user generates a public/private key pair by:

① Selecting two large primes at random - p, q

② Computing their system modulus $n = p \cdot q$

$$\phi(n) = (p-1)(q-1)$$

③ selecting at random the encryption key e
$1 < e < \phi(n)$, $\gcd(e, \phi(n)) = 1$

④ decryption key can be
$e \cdot d \equiv 1 \bmod \phi(n)$

- Public encryption key : $PU = \{e, n\}$
- Private decryption key : $PR = \{d, n\}$

→ to encrypt a message M the sender:
obtains public key of recipient $PU = \{e, n\}$
Computes: $C = M^e \bmod n$, where $0 \leq M < n$

→ to decrypt the ciphertext C the owner:
uses their private key $PR = \{d, n\}$
computes : $M = C^d \bmod n$

eg
- $p = 17$ & $q = 11$
- $n = p * q = 17 \times 11 = 187$
- $\phi(n) = (p-1)(q-1) = (17-1)(11-1) = 160$
- $e : \gcd(e, 160) = 1 \therefore e = 7$
- $d * e = 1 \bmod 160 \Rightarrow d < 160$
  $d \cdot 7 = 1 \bmod 160$
  $d = 23$
  $23 \times 7 = 161$

- public key = {7, 187}
- private key = {23, 187}