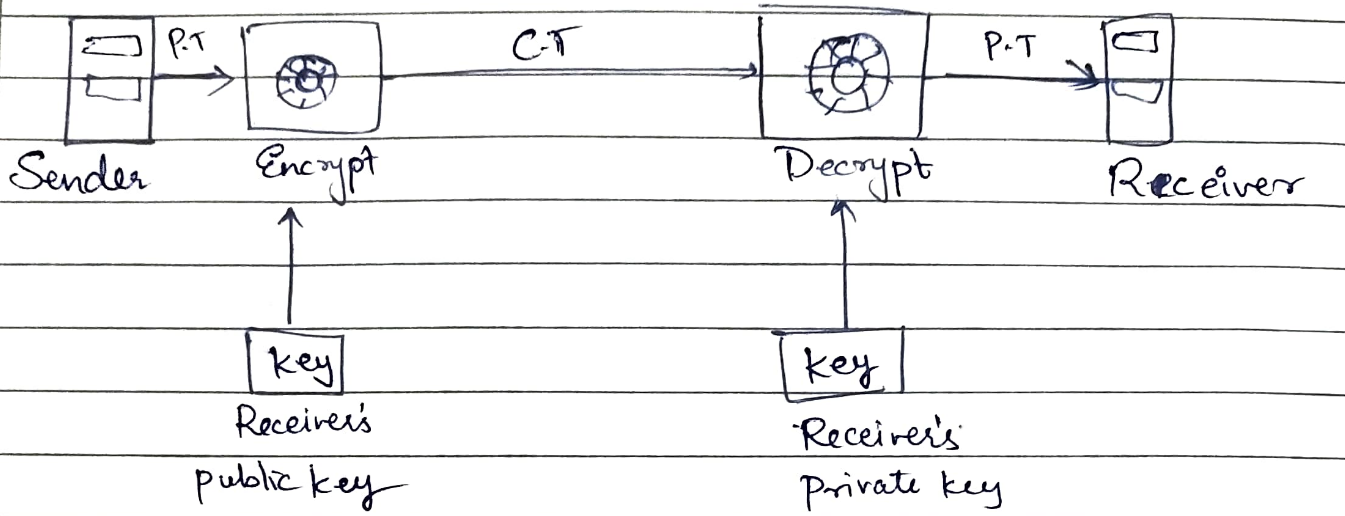
Q. Public key Cryptography



→ Public key encryption consists of 6 things -

Plaintext — Public key

Ciphertext — Encryption Algo

Private key — Decryption Algo.

→ Different keys are used for encryption & decryption.

→ It is assymetric encryption scheme.

→ Each receiver possesses a unique decryption key, generally referred to as private key

→ Receiver needs to publish an encryption key → public key

→ Involves 3rd party ~~to~~ which certifies that a particular key belongs to a specific entity.

→ Algo is complex enough to prohibit attacker from deducing the P.T from C.T and public key.

→ Developed to address 2 issues →

(i) key distribution - how to have secure communication to send the secret key

(ii) digital signatures - how to verify a message comes intact from the claimed sender.

Q. R S A ALGORITHM.

* best known & widely used public key scheme.
* uses large integers (eg. 1024 bits)
* makes use of expression with exponentials.
* plaintext is encrypted in blocks, with each block having a binary value less than n.
* ~~ece~~ Plaintext block M, Cipher Text block C.

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

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n.$$

Public key = {e, n}.
Private key = {d, n}.

$$ed \bmod \phi(n) = 1$$

$$ed = 1 \bmod \phi(n).$$

$$d = e^{-1} \bmod \phi(n).$$

* Security of RSA

(1) Brute force key search - trying all possible private keys.
(2) Mathematical Attacks - based on difficulty of computing $\phi(n)$, by factoring modulus n.
(3) Timing Attacks - these depend on the running time of the decryption algorithm.
(4) Chosen Ciphertext Attacks - this attack exploits properties of the RSA algorithm.

Q. Diffie-Hellman Key Exchange

* Method of public exchange of a secret key.
* public key distribution scheme can not be used to exchange arbitrary messages.
* public key distribution scheme can establish / compute a common key rather than sending it.
* public key dist- scheme is known only to the 2 participants
* value of key depends on the participants.
* based on exponentiation in a finite field.
* Security relies on the difficulty of computing discrete logarithms.

$$K = (Y_B)^{X_A} \bmod q$$
$$= (\alpha^{X_B} \bmod q)^{X_A} \bmod q$$
$$= \alpha^{X_B X_A} \bmod q$$

$$= (\alpha^{X_A})^{X_B} \bmod q$$
$$= (Y_A)^{X_B} \bmod q.$$

$$X_B = d\log_{\alpha,q}(Y_B)$$

* Man-In-the-Middle Attack:

The key exchange protocol is vulnerable to such attack because it doesnot authenticate the participants.

Q. Elliptic Curve Cryptography ECC.

* offers security like RSA and Doffie-Hellman but with smaller bit sizes, which acts as an advantage
* an elliptic curve is defined by an equation in 2 var x and y with coeff.
* Consider a cubic elliptic curve of form

$y^2 = x^3 + ax + b.$ x, y, a, b are real numbers.

* Zero Point / Point at Infinity 'O' -

If three points on an elliptic curve lie on a straight line, their sum is 'O'.

→ 'O' serves as the additive inverse, i.e.,

$O = -O$

$P + O = O + P = O.$

$P \neq O, \quad Q \neq O.$

$P + (-P) = P - P = O.$

Q. R.S. Algo -

Step 1 — Select p, q. 2 prime nos. (private)

Step 2 — $n = p.q$ (public, calculate).

Step 3 - e with $\gcd(\phi(n), e) = 1$

& $1 < e < \phi(n)$ (public)

Step 4 - $d = e^{-1} \bmod \phi(n)$ (private, calculated)

Q. SHA-1.

* originally designed by NIST & NSA.
* produces 160-bit hash values
* designed for compatibility with increased security provided by the AES cipher.

Q. SHA-512

* Message digest size - 512
* Message size < 2^{128}
* Block size - 1024
* Word size - 64
* No. of Steps → 80.
  - ↳ updating 512 bit buffer
  - ↳ updating 64 bit value W_t desired from the current message block.
  - ↳ a round constant based on cube root of first 80 prime numbers.

* Diagram

Q. HMAC

* specified as internet standard RFC2104.
* uses hash function on the message:

$$HMAC_K = Hash[(K^+ \text{ XOR } opad) \| Hash[(K^+ \text{ XOR } ipad) \| M]]$$

→ K^+ is the key padded out of size.

* Diagram

* opad, ipad are specified padding constraints.
* overhead is just 3 more has calculations than the message needs alone.
* any hash funcn can be used (SHA 512, MD5, whirlpool).
* Security - relates to Hash funcn used.
  - choose hash funcn based on speed vs security const.

Q. CMAC

* overcome message size limitation (CBC-MAC) using 2 keys & padding
* widely used in govt. and industry
* adopted by NIST SP800-38B. + Diagram
* Cipher based Message Authentication Code

Q.

SYMMETRIC	ASYMMETRIC.
* Only 1 key is used	* Two diff. keys are used.
* Same key used to encrypt and decrypt.	* public key for encryption and private key for decryption.
* Simpler method.	* Complicated cuz of 2 keys.
* faster	* Process is slower.
* length of key - 128/256 bits.	* length of keys 1024/2048 bits.
* used for transferring larger chunks of data.	* used for smaller transactions.
* the secret is shared.	* private key is not shared.
* higher risks of security.	* More secure
* Eg: RC4, DES, AES.	Eg: ~~RAS~~ RSA, ECC, Diffie

Q. Extended Euclidian

$$\boxed{P_0 = 0 \quad , \quad P_1 = 1}$$

$$P_i = (P_{i-2} - P_{i-1} \cdot q_{i-2}) \bmod n.$$

Eg: $15^{-1} \bmod 26$

		q_i
0:	$26 = 15 \times 1 + 11$	1
1:	$15 = 11 \times 1 + 4$	1
2:	$11 = 4 \times 2 + 3$	2
3:	$4 = 3 \times 1 + 1$	1
4:	$3 = 1 \times 3 + 0$	3

$P_0 = 0 \quad P_1 = 1$

$P_2 = (P_0 - P_1 q_0) \bmod 26$

$P_2 = (0 - 1) \bmod 26$

$= -1 \bmod 26 = 25$

$P_3 = P_1 - P_2 q_1 = (1 - 25) \bmod 26$

$= -24 \bmod 26 = 2$

$P_4 = (P_2 - P_3 q_2) \bmod 26 = (25 - 4) \bmod 26$

$= 21$

$P_5 = P_3 - P_4 q_3 = (2 - (21 \times 1)) \bmod 26$

$= -19 \bmod 26$

$= 7$

$15^{-1} \bmod 26 = 7$.