AIM : Write a program in C++ to implement RSA algorithm for key generationand cipher verification

OBJECTIVE :

To study

- Concept of Public key and Private Key.
- Public key algorithm
- Working of RSA algorithm

THEORY:

Asymmetric/Public Key Algorithm:

Public key algorithms were evolved to solve the problem of key distribution in symmetric algorithms. This is achieved by using one key for encryption and a different but related key for decryption. These algorithms are designed such that it is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key. Also in some algorithms, such as RSA, either of the two related keys can be used for encryption, with the other used for decryption.

A public key encryption scheme has six ingredients:

- **Plaintext:** This is readable message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The algorithm performs various transformations on the plaintext.
- **Public and private key:** This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.
- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.
- **Decryption algorithm:** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

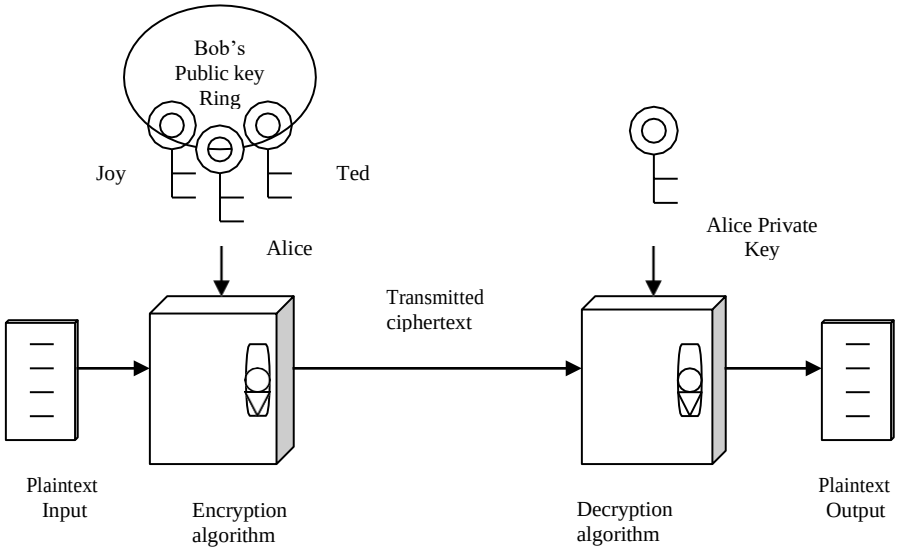

Fig. 1.1: Public key cryptography

The essential steps in public key algorithm are as follows:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or the other accessible file. This is the public key. The companion key is kept private. Each user maintains a collection of public keys obtained from other parties participating in communication.
3. If A wishes to send a confidential message to B, A encrypts the message using B's public key.
4. When B receives the message, B decrypts it using B's private key. No other recipient can decrypt the message because only B knows B's private key.

RSA Algorithm

RSA (which stands for Rivest, Shamir and Adleman who first publicly described it), an algorithm for public-key cryptographyinvolves three steps key generation, encryption and decryption.

RSA is a block cipher with each block having a binary value less than some number n. That is the block size must be less than or equal to log2 (n). Encryption

&decryption are of the following form, for some plaintext block M and ciphertext block C:

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

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

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. Thus, this is a public-key encryption algorithm with a public key of PU = {e, n} and a private key of PR = {d, n}. For this algorithm to be satisfactory for public key encryption, the following requirements must meet:

1. It is possible to find values of e, d, n such that $M^{ed} = M \bmod n$ for all M<n.
2. It is relatively easy to calculate M^e and C^d for all values of M<n.
3. It is infeasible to determine d given e and n.

Algorithm

1. Key generation

The keys (public key and private key) for the RSA algorithm are generated as:

1. Choose two distinct prime numbers p and q.

   For security purposes, the integers p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primality test.
2. Compute $n = pq$.

   n is used as the modulus for both the public and private keys
3. Compute $\varphi(n) = (p-1)(q-1)$, where φ is Euler's totient function
4. Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$, i.e. e and φ (n) are co prime.
   - e is released as the public key exponent.
   - e having a short bit-length and small Hamming weight results in more efficient encryption - most commonly 0x10001 = 65537. However, small values of e (such as 3) have been shown to be less secure in some settings.

5. Determine $d = e^{-1} \bmod \varphi(n)$; i.e. d is the multiplicative inverse of e mod φ(n).
    - This is more clearly stated as solve for d given $(d*e) \bmod \varphi(n) = 1$
    - This is often computed using the extended Euclidean algorithm.
    - d is kept as the private key exponent.

Public key : PU = {e, n}
Private Key : PR = {d, n}

2. Encryption

Alice transmits her public key (e, n) to Bob and keeps the private key secret. Bob then wishes to send message M to Alice.He computes the ciphertext c corresponding to

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

This can be done quickly using the method of exponentiation by squaring. Bob then transmits C to Alice.

3. Decryption

Alice can recover M from C by using her private key exponent d via computing

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

Example

1. Select two prime numbers, p = 17 and q = 11.
2. Calculate n = pq = 17*11 = 187.
3. Calculate Ø(n) = (p-1)(q-1) = 16*10 = 160.
4. Select e such that relatively prime to Ø(n) = 160 and less than Ø(n); we choose e = 7.
5. Determine d such that de ≡ 1 (mod 160) and d < 160. The correct value is d = 23, because 23*7 = 161 = 10*160+1; d can be calculated using the extended Euclid's algorithm.

The resulting keys are public key PU = {7, 187} and private key PR = {23, 187}. The example shows the use of these keys for plaintext input of M=88.

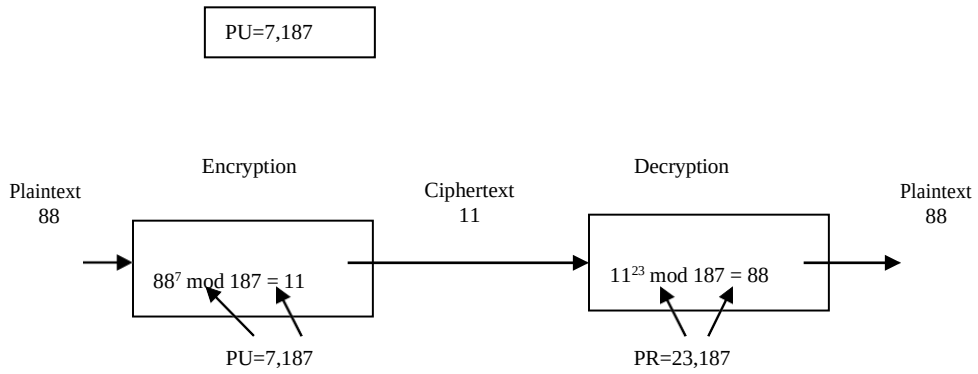

Fig. 1.2 : Example of RSA.

INPUT:

- Two prime numbers p = 17 and q = 11.
- Select e = 7.
- Plaintext = 88.

OUTPUT:

- PU = 7, 187.
- PR = 23, 187.
- Ciphertext = 11.

FAQs:

1. What are symmetric and asymmetric ciphers?
2. Why **s**trong Primes are necessary in RSA?
3. Alice wants to generate a pair of RSA public and private keys. She starts by selecting two primes p = 5 and q = 7. Compute n, φ(n).
4. For long messages, RSA will be applied in blocks. If the block is very small, say it contains only one letter in each block, will the encryption be secure?
5. What are the possible attacks on RSA?

PRACTISE ASSIGNMENTS:

1. Implement RSA for text input.
2. Implement RSA for client server system.