Name : Prem Vinod Bansod
Roll No : 41310
Assignment No : 04 (ICS)

Title: RSA algorithm.

Problem Statement: To implement a RSA algorithm.

Objective:

- The Basic Concepts of RSA.
- General structure of RSA.
- Logical implementation of RSA.

Theory:

Introduction:

RSA (Rivest–Shamir–Adleman) is one of the first public-key cryptosystems and is widely used for secure data transmission. It is an asymmetric cryptographic algorithm. Asymmetric means that there are two different keys. This is also called public key cryptography, because one of them can be given to everyone. In such a cryptosystem, the encryption key is public and it is different from the decryption key which is kept secret (private). In RSA, this asymmetry is based on the practical difficulty of the factorization of the product of two large prime numbers, the "factoring problem". The acronym RSA is made of the initial letters of the surnames of Ron Rivest, Adi Shamir, and Leonard Adleman, who first publicly described the algorithm in 1978.

A user of RSA creates and then publishes a public key based on two large prime numbers, along with an auxiliary value. The prime numbers must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, and if the public key is large enough, only someone with knowledge of the prime numbers can decode the message feasibly. Breaking RSA encryption is known as the RSA problem.

RSA is a relatively slow algorithm, and because of this, it is less commonly used to directly encrypt user data. More often, RSA passes encrypted shared keys for symmetric key cryptography which in turn can perform bulk encryption-decryption operations at much higher speed.

OPERATIONS

The RSA algorithm involves four steps: key generation, key distribution, encryption and decryption.

A basic principle behind RSA is the observation that it is practical to find three very large positive integers e, d and n such that with modular exponentiation for all integers m (with $0 \leq m < n$):

$$(m^e)^d = m \pmod{n}$$

and that even knowing e and n or even m it can be extremely difficult to find d.

In addition, for some operations it is convenient that the order of the two exponentiations can be changed and that this relation also implies:

$$(m^d)^e = m \pmod{n}$$

RSA involves a public key and a private key. The public key can be known by everyone, and it is used for encrypting messages. The intention is that messages encrypted with the public key can only be

decrypted in a reasonable amount of time by using the private key. The public key is represented by the integers n and e; and, the private key, by the integer d(although n is also used during the decryption process. Thus, it might be considered to be a part of the private key, too). m represents the message.

a. Key generation

The keys for the RSA algorithm are generated the following way:

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

similar in magnitude but differ in length by a few digits to make factoring harder. Prime integers

can be efficiently found using a primarily test.

2. Compute $n = pq$.

n is used as the modulus for both the public and private keys. Its length, usually expressed in

bits, is the key length.

3. Compute $\lambda(n) = \mathrm{lcm}(\lambda(p), \lambda(q)) = \mathrm{lcm}(p - 1, q - 1)$, where λ is Carmichael's totient function.

This value is kept private.

4. Choose an integer e such that $1 < e < \lambda(n)$ and $\gcd(e, \lambda(n)) = 1$; i.e., e and $\lambda(n)$ are co-prime.

5. Determine d as $d \equiv e^{-1} \pmod{\lambda(n)}$; i.e., d is the modular multiplicative inverse of e modulo λ(n).

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the private (or decryption) exponent d, which must be kept secret. p, q, and λ(n) must also be kept secret because they can be used to calculate d.

b. Key distribution

Suppose that Bob wants to send information to Alice. If they decide to use RSA, Bob must know Alice's public key to encrypt the message and Alice must use her private key to decrypt the message. To enable Bob to send his encrypted messages, Alice transmits her public key (n, e) to Bob via a reliable, but not necessarily secret, route. Alice's private key (d) is never distributed.

c. Encryption

After Bob obtains Alice's public key, he can send a message M to Alice. To do it, he first turns M (strictly speaking, the un-padded plaintext) into an integer m (strictly speaking, the padded plaintext), such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the cipher text c, using Alice's public key e, corresponding to

$$c \equiv m^e \pmod{n}$$

Bob then transmits c to Alice.

d. Decryption

Alice can recover m from c by using her private key exponent d by computing

$$m \equiv c^d \pmod{n}$$

Given m, she can recover the original message M by reversing the padding scheme.

STEPS OF RSA ALGORITHM

STEP-1: Select two co-prime numbers as p and q.

STEP-2: Compute n as the product of p and q.

STEP-3: Compute (p-1)*(q-1) and store it in z.

STEP-4: Select a random prime number e that is less than that of z.

STEP-5: Compute the private key, d as e * mod -1 (z).

STEP-6: The cipher text is computed as * mod n.

STEP-7: Decryption is done as * mod n .

EXAMPLE OF RSA ALGORITHM

1. Select primes: p = 17 ; q = 11
2. Calculate n = pq = 17 x 11 = 187
3. Calculate ø(n) = (p–1)(q-1) = 16x10 = 160
4. Select e: GCD(e,160) = 1 ; choose e = 7
5. Derive d: de = 1 mod 160 and d < 160 Get d = 23 since 23x7 = 161 = 10x160+1
6. Publish public key: PU = {7,187}
7. Keep private key secret: PR = {23,187}

RSA Encryption and Decryption :

- Sample RSA encryption/decryption is:
- given message M = 88 (nb 88 < 187)

  Encryption:
- C = 88 7 mod 187 = 11

  Decryption:
- M = 11 23 mod 187 = 88

CONCLUSION: Hence, RSA algorithm was successfully implemented.