**Experiment No. 6**

**Aim:** Implementation and analysis of RSA cryptosystem and Digital signature scheme using RSA/El Gamal.

**Related Theory:**

**RSA Algorithm:**

RSA is an asymmetric system, which means that a key pair will be generated (we will see howsoon), a public key and a private key, obviously you keep your private key secure and pass around the public one.

The algorithm was published in the 70’s by Ron Rivest, Adi Shamir, and Leonard Adleman,hence RSA, and it sort of implement’s a trapdoor function such as Diffie’s one.RSA is rather slow so it’s hardly used to encrypt data, more frequently it is used to encrypt andpass around symmetric keys which can actually deal with encryption at a faster speed.RSA algorithm is asymmetric cryptography algorithm. Asymmetric actually means that itworks on two different keys i.e. Public Key and Private Key. As the name describes that the

Public Key is given to everyone and Private Key is kept private.

**An example of asymmetric cryptography:**

1. A client (for example browser) sends its public key to the server and requests for somedata.

2. The server encrypts the data using client’s public key and sends the encrypted data.

3. Client receives this data and decrypts it.Since this is asymmetric, nobody else except browser can decrypt the data even if a third partyhas public key of browser.The idea! The idea of RSA is based on the fact that it is difficult to factorize a large integer. Thepublic key consists of two numbers where one number is multiplication of two large primenumbers. And private key is also derived from the same two prime numbers. So if somebody canfactorize the large number, the private key is compromised. Therefore encryption strength totallylies on the key size and if we double or triple the key size, the strength of encryption increasesexponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024bit keys could be broken in the near future. But till now it seems to be an infeasible task.

**Why the RSA algorithm is used?**

RSA derives its security from the difficulty of factoring large integers that are the product of twolarge prime numbers. Multiplying these two numbers is easy, but determining the original primenumbers from the total - or factoring -- is considered infeasible due to the time it would takeusing even today's supercomputers.

The public and private key generation algorithm is the most complex part of RSA cryptography.Two large prime numbers, p and q, are generated using the Rabin-Miller primality test algorithm.A modulus, n, is calculated by multiplying p and q. This number is used by both the public andprivate keys and provides the link between them. Its length, usually expressed in bits, is calledthe key length.

**RSA security:**

RSA security relies on the computational difficulty of factoring large integers. As computingpower increases and more efficient factoring algorithms are discovered, the ability to factorlarger and larger numbers also increases.Encryption strength is directly tied to key size, and doubling key length can deliver an exponential increase in strength, although it does impair performance. RSA keys are typically1024- or 2048-bits long, but experts believe that 1024-bit keys are no longer fully secure against attacks. This is why the government and some industries are moving to a minimum key length of 2048-bits.

Barring an unforeseen breakthrough in quantum computing, it will be many years before longer keys are required, but elliptic curve cryptography (ECC) is gaining favor with many security experts as an alternative to RSA to implement public key cryptography. It can create faster,smaller and more efficient cryptographic keys.Modern hardware and software are ECC-ready, and its popularity is likely to grow, as it candeliver equivalent security with lower computing power and battery resource usage, making itmore suitable for mobile apps than RSA. Finally, a team of researchers, which included AdiShamir, a co-inventor of RSA, has successfully created a 4096-bit RSA key using acousticcryptanalysis; however, any encryption algorithm is vulnerable to attack.

**This is the original algorithm:**

1. Generate two large random primes, p and q, of approximately equal size such that their

product n=pq is of the required bit length, e.g. 1024 bits.

2. Compute n=pq and ϕ=(p−1)(q−1)

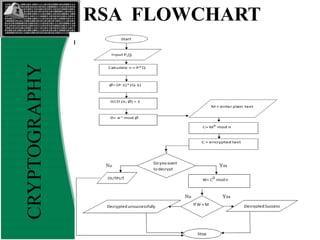
3. Choose an integer e, 1<e<ϕ, such that gcd(e,ϕ)=1.

4. Compute the secret exponent d, 1<d<ϕ, such that ed≡1modϕ.

5. The public key is (n,e) and the private key (d,p,q). Keep all the values d, p, q and ϕ secret.

[Sometimes the private key is written as (n,d) because you need the value of n when using d.

Other times we might write the key pair as ((N,e),d).



Example

An example of generating RSA Key pair is given below. (For ease of understanding, the primes

p & q taken here are small values. Practically, these values are very high).

● Let two primes be p = 7 and q = 13. Thus, modulus n = pq = 7 x 13 = 91.

● Select e = 5, which is a valid choice since there is no number that is common factor of 5 and

(p − 1)(q − 1) = 6 × 12 = 72, except for 1.

● The pair of numbers (n, e) = (91, 5) forms the public key and can be made available to anyone

whom we wish to be able to send us encrypted messages.

● Input p = 7, q = 13, and e = 5 to the Extended Euclidean Algorithm. The output will be d =29.

● Check that the d calculated is correct by computing −de = 29 × 5 = 145 = 1 mod 72

● Hence, public key is (91, 5) and private keys is (91, 29).

**Encryption and Decryption**

Once the key pair has been generated, the process of encryption and decryption are relatively

straightforward and computationally easy.

RSA Encryption

● Suppose the sender wish to send some text message to someone whose public key is (n, e).

● The sender then represents the plaintext as a series of numbers less than n.

● To encrypt the first plaintext P, which is a number modulo n. The encryption process is simple

mathematical step as −

C = Pe mod n

● In other words, the ciphertext C is equal to the plaintext P multiplied by itself e times and then

reduced modulo n. This means that C is also a number less than n.

● Returning to our Key Generation example with plaintext P = 10, we

get ciphertext C –

C = 105 mod 91

RSA Cryptosystem

This cryptosystem is one the initial system. It remains most employed cryptosystem even today.

The system was invented by three scholars Ron Rivest, Adi Shamir, and Len Adleman and hence, it is

termed as RSA cryptosystem. We will see two aspects of the RSA cryptosystem, firstly

generation of key pair and secondly encryption-decryption algorithms.

**Implementation/Code:**

**import math**

**def gcd(a, h):**

**temp = 0**

**while (1):**

**temp = a % h**

**if (temp == 0):**

**return h**

**a = h**

**h = temp**

**p = 3**

**q = 7**

**n = p\*q**

**e = 2**

**phi = (p-1)\*(q-1)**

**while (e < phi):**

**if (gcd(e, phi) == 1):**

**break**

**else:**

**e = e+1**

**k = 2**

**d = (1 + (k\*phi))/e**

**msg = 12.0**

**print("Message data = ", msg)**

**c = pow(msg, e)**

**c = math.fmod(c, n)**

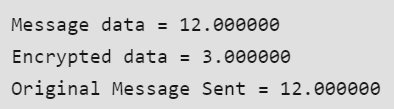
**print("Encrypted data = ", c)**

**m = pow(c, d)**

**m = math.fmod(m, n)**

**print("Original Message Sent = ", m)**

**Output:**

****