**RSA description and algorithm**

RSA stands for Rivest, Shamir, and Adleman, they are the inventors of the RSA cryptosystem. RSA is one of the algorithms used in PKI (Public Key Infrastructure), asymmetric key encryption scheme. RSA is a block chiper, it encrypt message in blocks (block by block). The common size for the key length now is 1024 bits for P and Q, therefore N is 2048 bits, if the implementation (the library) of RSA is fast enough, we can double the key size.

**Key Generation 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) phi = (p-1)(q-1).

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

4. Compute the secret exponent *d*, 1 < d < phi, such that ed ≡ 1 (mod phi).

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

n is known as the *modulus*.

e is known as the *public exponent* or *encryption exponent* or just the *exponent*.

d is known as the *secret exponent* or *decryption exponent*.

**Encryption**

Sender A does the following:-

1. Obtains the recipient B's public key (n, e).

2. Represents the plaintext message as a positive integer *m* .

3. Computes the ciphertext c = me mod n.

4. Sends the ciphertext *c* to B.

**Decryption**

Recipient B does the following:-

1. Uses his private key (n, d) to compute m = cd mod n.

2. Extracts the plaintext from the message representative *m*.

**A very simple example of RSA encryption**

This is an extremely simple example using numbers you can work out on a pocket calculator (those of you over the age of 35 45 can probably even do it by hand).

1. Select primes p=11, q=3.

2. n = pq = 11.3 = 33 phi = (p-1)(q-1) = 10.2 = 20

3. Choose e=3 Check gcd(e, p-1) = gcd(3, 10) = 1 (i.e. 3 and 10 have no common factors except 1), and check gcd(e, q-1) = gcd(3, 2) = 1 therefore gcd(e, phi) = gcd(e, (p-1)(q-1)) = gcd(3, 20) = 1

4. Compute d such that ed ≡ 1 (mod phi) i.e. compute d = e-1 mod phi = 3-1 mod 20 i.e. find a value for d such that phi divides (ed-1) i.e. find d such that 20 divides 3d-1.

Simple testing (d = 1, 2, ...) gives d = 7 Check: ed-1 = 3.7 - 1 = 20, which is divisible by phi.

5. Public key = (n, e) = (33, 3) Private key = (n, d) = (33, 7).

This is actually the smallest possible value for the modulus n for which the RSA algorithm works.

Now say we want to encrypt the message m = 7, c = me mod n = 73 mod 33 = 343 mod 33 = 13. Hence the ciphertext c = 13.

To check decryption we compute m' = cd mod n = 137 mod 33 = 7. Note that we don't have to calculate the full value of 13 to the power 7 here. We can make use of the fact that a = bc mod n = (b mod n).(c mod n) mod n so we can break down a potentially large number into its components and combine the results of easier, smaller calculations to calculate the final value.

One way of calculating m' is as follows:- m' = 137 mod 33 = 13(3+3+1) mod 33 = 133.133.13 mod 33 = (133 mod 33).(133 mod 33).(13 mod 33) mod 33 = (2197 mod 33).(2197 mod 33).(13 mod 33) mod 33 = 19.19.13 mod 33 = 4693 mod 33 = 7

**Example 1:**

Using small numbers for clarity, here are results of an example run:

enter prime p: 47

enter prime q: 71

n = p\*q = 3337

(p-1)\*(q-1) = 3220

Guess a large value for public key e then we can work down from there.

enter trial public key e: 79

trying e = 79

Use private key d: 1019

Publish e: 79

and n: 3337

cipher = char^e (mod n) ----------------- char = cipher^d (mod n)