Int. J. Chem. Sci.: 14(4), 2016, 2273-2278
ISSN 0972-768X
www.sadgurupublications.com

RSA ENCRYPTION USING THREE MERSENNE PRIMES

Ch. J. L. PADMAJA[a*], V. S. BHAGAVAN[a] and B. SRINIVAS[b]

[a]Department of Mathematics, KL University, Vaddeswaram (A.P.) INDIA
[b]Government Polytechnic for Minorities, GUNTUR (A.P.) INDIA

ABSTRACT

As the internet evolves and computer networks become accessible, most business transactions are done over the internet. Potential threat to the information is always present and attempts are being made to provide maximum security over the network. One of such new techniques is using multiple prime numbers for RSA cryptosystem, which is not easily breakable. Also, typical prime numbers are used in order to strengthen the algorithm to ensure safe data exchange. In this paper, we used three Mersenne prime numbers to construct a new RSA cryptosystem which provides more efficiency and reliability over the network. Mathematics Subject Classification (2010): 11T71, 14G50, 68P25, 94A60

Key words: Factorization, Mersenne prime, Symmetric key.

INTRODUCTION

The fundamental objective of cryptography is to enable two people to communicate over an insecure channel in such a way that an opponent cannot understand what is being said. The sender encrypts a plain text using a predetermined key and sends the resultant cipher text over the channel. An intruder, upon seeing the cipher text over the channel cannot determine what the original message was; but the receiver who knows the encryption key can decrypt the cipher text and reconstruct the plain text[1].

Mathematically, a cryptosystem is a five-tuple (P, C, K, E, D) where:

1. P is the finite set of possible plaintexts
2. C is the finite set of possible cipher texts.
3. K is the keyspace, finite set of possible keys.
4. For each k Є K, there is an encryption rule e_k Є E and a corresponding decryption rule d_k Є D.

[*]Author for correspondence; E-mail: padmajachivukula@gmail.com

Each e_k : P → C and dk: C → P are functions such that $d_k(e_k(x)) = x$ for every x Є P.

Public key cryptosystem

In symmetric key cryptosystems or private key cryptosystems, d_k (decryption key) is either the same as e_k (encryption key) or easily derived from it. It means the exposure of either e_k or d_k renders the system insecure.

In a public key cryptosystem, it is computationally infeasible to determine d_k given e_k. The encryption rule e_k is public key, which could be published in a directory. The decryption rule d_k is the private key which is kept secret with the receiving person[1].

The best known and most widely used public key system is RSA, which was first proposed by Rivest et al.[2] It is an asymmetric (public key) cryptosystem based on number theory. Its security is based on the difficulty of factorization of the large number which is a well known mathematical problem.

Breaking RSA is at least as hard as factoring n. Factoring large numbers is not probably hard but no efficient algorithm exists today to break in a reasonable amount of time. In order to strengthen the security of RSA several modifications in the algorithm are made. A modified RSA encryption was suggested by Ivy et al.[3] using 'n' prime numbers instead of two. A further attempt was made at the use of two Mersenne primes because of the fact that small primes are used to generate large Mersenne primes[4].

An attempt has been made in this paper to present the mathematics behind the RSA cryptosystem, existing algorithm and propose a new algorithm using three Mersenne primes numbers.

Mathematics behind RSA

This cryptosystem uses computations in *Zn*. Let n = p x q, where p and q are distinct primes. Let P = C = Z_n, and $K = (n, p, q, e, d)$.

Define $e_K(x) = x^e \bmod n = y$ *and*

$d_K(y) = y^d \bmod n$ *for* $(x,y$ Є $Zn)$.

The public values are (n,e) and secret values are (p, q, d). Note that $\varphi(n) = (p-1)(q-1)$.

We will now verify the decryption.

Since ed ≡ 1(mod $\varphi(n)$), there is an integer k such that ed = $k.\ \varphi(n)+1$

We have $(x^e)^d = x^{k.\ \varphi(n)+1} = x^{1+k(p-1)(q-1)} = x\ .(x^{p-1})^{k.(q-1)}$o $x\ (mod\ p)$

[Since by Fermat theorem, $x^{p-1} \equiv 1$ mod p]

Therefore, $x^{ed} \equiv x \pmod{p}$.

Similarly, $x^{ed} \equiv x$ (mod q).

By definition, there must be two integers r and s such that $x^{ed}-x = p.r = q.$s.

Since p and q are distinct primes, either p must be a factor of s or q must be a factor of r due to the unique prime factorization.

There has to be an integer s' such that $x^{ed}-x = qs = qps'$, i.e. $n = p.\ q$ divides $x^{ed}- x.$

$x^{ed} \equiv x \pmod{n}$. Therefore, $d_k\ (e_k(x)) = d_K\ (x^e \bmod n) = (x^e \bmod n)^{\ d} \bmod n = x^{ed} \bmod n = x \bmod n$ and $e_K\ (d_K(x)) = x\ in\ Zn.$

The idea behind today's RSA cryptosystem is –

1. Prime number generation is easy. It is easy to find a random prime number of given size.
2. Multiplication is easy. Given p and q, it is easy to find their product n = p.q
3. Factoring is hard. Given such n, it is deemed to be quite hard to recover the prime numbers p, q.

Existing RSA algorithm

Key generation: Select two prime numbers, p and q and $p \neq q$.

Calculate n = p x q.

Calculate φ(n) = (p -1) x (q-1).

Select integer e such that gcd (e, φ (n)) = 1; 1 < e < φ (n).

Calculate private key $d = e^{-1}$ (mod φ (n)).

Then public key = {e, n}.

Private key = {d, n}.

Encryption: $c = m^e$ mod n.

Decryption: $m = c^d$ mod n. Where, c - cipher text, m - message

Mersenne primes

Mersenne number is named after a French Monk, Marin Mersenne who studied these numbers in early 17th century. These numbers are of the form 2^p-1 where p is itself a prime number and are represented as $M_p = 2^p$-1.

The first few Mersenne prime numbers are 1, 3, 7, 15, 31, 63, 127, 255,...

A total of 48 Mersenne primes are known as of October 2014. The largest known Mersenne prime[5] is $2^{57,885,161} – 1$.

Proposed RSA algorithm

To increase the complexity of the RSA algorithm, a new component *r* is added and explained below.

Key generation: Receiver chooses three Mersenne primes *p, q* and *r.*

Computes $n = p.q.r$ and $\varphi(n) = (p\text{-}1)(q\text{-}1)(r\text{-}1)$

Chooses e, $1 < e < \varphi(n)$ *such that* $gcd(e, \varphi(n)) = 1$

Finds *d* such that $ed = 1 \bmod \varphi(n)$.

Now the receiver's public key pair is (*n, e*) and private key is *d.*

Encryption: To encrypt a message *m,* the sender computes,

$c = m^e \bmod n$ and sends this to the receiver.

Decryption: To decrypt *c*, the receiver computes

$m= c^d \bmod n$

Implementation

We carry an RSA encryption with artificially small parameters. B is the message originator and A is the receiver.

Key generation: A chooses three Mersenne primes $p = 3, q =7$ and $r = 15$

Computes $n = p.q.r = 315$ and $\varphi(n) = (p\text{-}1)(q\text{-}1)(r\text{-}1) = 168$.

Chooses e = 79 such that $gcd(e, \varphi(n)) = 1$

Using extended Euclidean algorithm finds $d =151$ such that $ed = 1 \bmod \varphi(n)$.

Now A's public key pair is ($n = 315, e =79$), while A's private key is $d =151$.

Encryption: To encrypt a message $m = 52$, B computes,

$c = m^e \bmod n = 52^{79} \bmod 315$

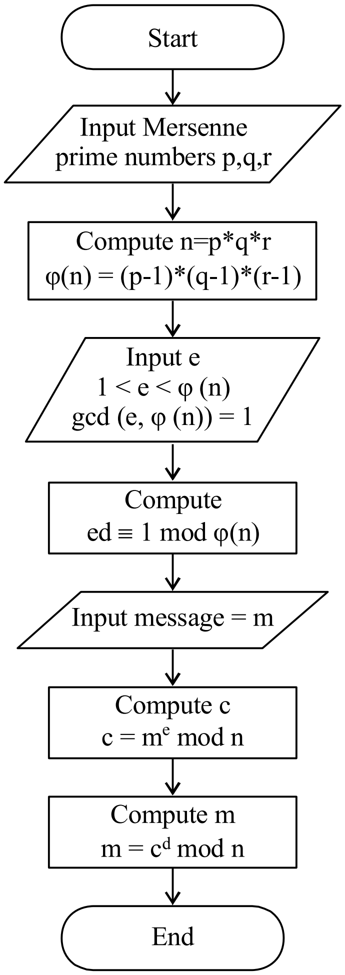
3.6666051757388353917113336844361e+135 mod 315 = 178 and sends this to A.

Decryption: To decrypt *c*, A computes

$m = c^d \bmod n = 178^{151} \bmod 315$

= 6.5075924802805196719974005781156e + 339 mod 315 = 52

The flow chart given below explains the proposed RSA algorithm, which is adapted here in this paper.



CONCLUSION

In this paper, we have proposed RSA algorithm using three Mersenne primes as against the traditional RSA algorithm of two normal primes. The important aspect of RSA cryptography is the selection of public key and generation of private key. Public key can be generated randomly. The security of RSA is based on the hope that the encryption function is one way and so it is computationally infeasible for an intruder to decrypt a cipher text. Mersenne primes are used to enhance the security. The strength of large prime number is dependent on three factors, p, q, and r. It is deemed to be difficult to break the large number into three. If any algorithms are developed in future that makes factorization of a large prime feasible, usage of three primes instead of two can increase the strength of that algorithm. Here, the algorithm is generated using only small numbers, but real problems can be implemented using MATLAB for verification of the proposed algorithm.

REFERENCES

1. D. R. Stinson, Cryptography, Theory and Practice, 3rd Ed., CRC Press, Boca Raton, FL (2002).
2. R. Rivest, A. Shamir and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of ACM, **21(2)**, 120-126 (1978).
3. B. P. U. Ivy, P. Mandiwa and M. Kumar, A Modified RSA Cryptosystem Based on ‘n’ Prime Numbers, Int. J. Engg. Comput. Sci., **1(2)**, 62-66 (2012).
4. S. Pund and C. Desai, Implementation of RSA Algorithm using Mersenne Prime, Int. J. Networking & Parallel Computing, **1(2)**, 33-41 (2013).
5. https://en.wikipedia.org/wiki/Mersenne_prime.

Revised : 13.08.2016 *Accepted : 16.08.2016*