Volume 2, Issue 8, August 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm

Sonal Sharma
Assistant Professor
MIT, Kota, Rajasthan, India

Jitendra Singh Yadav
Assistant Professor
SBCET, Jaipur, Rajasthan, India

Prashant Sharma
Assistant Professor
MIT, Kota, Rajasthan, India

Abstract-In asymmetric key cryptography, also called Public Key cryptography, two different keys (which forms a key pair) are used. One key is used for encryption & only the other corresponding key must be used for decryption. No other key can decrypt the message – not even the original (i.e. the first) key used for encryption. The beauty of this scheme is that every communicating party needs just a key pair for communicating with any number of other communicating parties. Once some one obtains a key pair, he /she can communicate with any one else.
The Short Range Natural Number (SRNN) algorithm is similar to RSA algorithm with some modification. This modification increases the security of the cryptosystem. In this algorithm we have an extremely large number that has two prime factors (similar to RSA). In addition of this we have used two natural numbers in pair of keys (public , private).These natural numbers increase the security of the cryptosystem. So its name is "Modified RSA Public Key Cryptosystem using Short Range Natural Number Algorithm".

Keywords- Cryptography, RSA, Key, Symmetric Key and Asymmetric Key Cryptography.

I. INTRODUCTION

Symmetric-key cryptography is based on the sender and receiver of messages knowing and using the same secret key. The sender uses the secret key to encrypt the message and the receiver uses the same secret key to decrypt it. The main problem of symmetric key cryptography is getting the sender and receiver to agree on the same secret key without anyone else knowing it. Because all keys in a symmetric key cryptosystem must remain secret, symmetric key cryptography often has difficulty providing secure key management, especially in open systems with a large number of users. To solve this problem, Diffie and Hellman introduced a new approach to cryptography and, in effect, challenged cryptologists to come up with a cryptographic algorithm that met the requirements for public-key systems . Public-key cryptography is used where each user has a pair of keys, one called the public key and the other private key. Each user's public key is published while the private key is kept secret and thereby the need for the sender and the receiver to share secret information (key) is eliminated. The only requirement is that public keys are associated with the users in a trusted (authenticated) manner using a public key infrastructure (PKI) . The public key cryptosystems are the most popular, due to both confidentiality and authentication facilities.

PKC depends upon the existence of one-way functions, or mathematical functions that are easy to compute whereas their inverse function is relatively difficult to compute. Generic PKC employs two keys that are mathematically related although knowledge of one key does not allow someone to easily determine the other key. One key is used to encrypt the plaintext and the other key is used to decrypt the ciphertext. The important point here is that it does not matter which key is applied first, but that both keys are required for the process to work . Because pair of keys are required, this approach is also called asymmetric cryptography[1,2].

The Rivest-Shamir-Adleman (RSA) cryptosystem is one of the best known publickey cryptosystems for key exchange or digital signatures or encryption of blocks of data. RSA uses a variable size encryption block and a variable size key. The key-pair is derived from a very large number, n, that is the product of two prime numbers chosen according to special rules; these primes may be 100 or more digits in length each, yielding an n with roughly twice as many digits as the prime factors.
The public key information includes n and a derivative of one of the factors of n; an attacker cannot determine the prime factors of n (and, therefore, the private key) from this information alone and that is what makes the RSA algorithm so secure.
RSA's safety is due to the difficulty in factoring large prime numbers. The main arithmetic operation in the RSA Cryptosystem is modular exponentiation defined as $C = M^e \bmod n$ for encryption and $M = C^d \bmod n$ for decryption, where C is the cipher, M
is the message, e is the public key, *d* is the private key, and n is the modulus[3,4].

RSA algorithm has some important parameters affecting its level of security and speed. By increasing the modulus length plays an important role in increasing the complexity of decomposing it into its factors.This will increase the length of private key and hence difficult to be decrypted without knowing the decryption key.When the length of message is changed then the length of encrypted message will proportionally change, hence larger chunks are selected to obtained larger encrypted message to increase the security of the data in use[5]. RSA -1024 bits is good for last 20 years but now Bernstain described circuitry for fast factorization. It is entirely possible that an organization with suffientely deep pockets can build a large scale version of his circuits and effectively crack an RSA 1024 bit message in a relatively short period of time, which could range any where from a number of minutes to some days [7,8]. Time analysis of RSA algorithm performed by varing its parametes[9].We use natural numbers in pair of keys in addition to existing parametes of RSA.Then after simulations of results on basis of speed and security we compare the RSA and new algorithm . We use fast modulation method in RSA for big exponential calculation [10-12].

The SRNN algorithm is similar with RSA with some modification. SRNN algorithm is also a public key cryptography algorithm. In this algorithm we have extremely large number that has two prime factors (similar to RSA) .In addition of this we have used two short range natural numbers in pair of keys. This modification increases the security of the cryptosystem. So its name is short range natural number public key algorithm

II. RSA CRYPTOSYSTEM

A. RSA Key Genration, Encryption, Decryption Process

The following steps are there to determine the values of **e, d and n.**

- Choose two very large (100+ digit) prime numbers p and q.
- Set n equal to p * q.
- Choose any large integer, e, such that
  GCD (e, ((p-1) * (q-1)) = 1
- Find d such that e * d mod ((p-1)*(q-1)) = 1

The public key is the number (n, e). Although these values are publicly known, it is computationally infeasible to determine d from n and e if p and q are large enough. To encrypt a message, M, with the public key, creates the cipher, C, using the equation:

$C = M^e \bmod n$ e: Public Key

The receiver then decrypts the cipher with the private key using the equation:

$M = C^d \bmod n$ d: Private Key

Now, this might look a bit complex and, indeed, the mathematics does take a lot of computer power, given the large size of the numbers; since p and q may be 100 digits (decimal) or more, d and e will be about the same size and n may be over 200 digits. Nevertheless, a simple example may help. In this example, the values of p, q, e and d are purposely chosen to be very small and the reader will see exactly how badly these values perform, but hopefully the algorithm will be adequately demonstrated.

III. MODIFIED RSA PUBLIC KEY CRYPTOSYSTEM USING SHORT RANGE NATURAL NUMBER ALGORITHM

A. Introduction

The SRNN algorithm is similar with RSA with some modification. SRNN algorithm is also a public key cryptography algorithm. In this algorithm we have extremely large number that has two prime factors (similar to RSA) .In addition of this we have used two short range natural numbers in pair of keys. This modification increases the security of the cryptosystem. So its name is short range natural number public key algorithm

B. SRNN Key Generation, Encryption, Decryption Process

1) Key Generation process

- Generate two large random primes, p and q, of approximately equal size such that their product n = p x q is of the required bit length, e.g. 1024 bits.
- Compute n= p x q.
- Compute phi = (p-1) (q-1).
- Choose an integer e, 1 < e < phi, such that gcd (e, phi) = 1. Compute the secret exponent d, 1<d<phi, such that (e x d) mod phi = 1.
- Pick short range natural number u randomly such that u < phi-1.

- Pick another short range natural number a randomly such that phi > a > u. And compute u^a .
- Find d such that- $e * d \mod ((p-1)*(q-1)) = 1$
- The public key is (n,e,u^a) and the private key is (d,a,u). The values of p, q, and phi should also be kept secret.

2) Encryption process

Sender does the following:-

- Obtains the recipient's public key (n,e,u^a)
- Represents the plaintext message as a positive integer m.
- Computes the cipher text $c = (m\, u^a)^e \mod n$.
- Sends the cipher text c to recipient.

3) Decryption process

Recipient does the following:-

- Uses his private key (d, a, u) to compute $m = (v^e c)^d \mod n$ Where $v = u^{phi-a} \mod n$.
- Extracts the plaintext from the integer representative m.

C. Implementation of SRNN Algorithm

Here programs are implemented using Java library functions. We have implemented the SRNN cryptosystem in two forms: a console mode implementation, as well as a user friendly GUI implementation. We focus on the user friendly GUI mode implementation here .At start default values of input filename, encrypt filename, decrypt filename, key_size, message block size or chunk size , prime numbers p and q are used and user have flexibility to change these values as and when required. If user want to generate new public and private keys then new frame is appeared to select new keys.

Any practical implementation of the SRNN cryptosystem would involve working with large integers. One way of dealing with this requirement would be to write our own library that handles all the required functions but we use BigInteger library (Java) in Java. Here use of natural numbers make it more secure algorithm.

Issues with this version of the implementation

- The final decoded text does not have line feeds that the original plaintext had.
- The program errors out if the ASCII value of the message or chunk size m is greater than the modulus length n. It must be lower to perform modulus operations.
- In practical, the public exponent in the SRNN algorithm is usually much smaller than the private exponent.

IV. COMPARISON BETWEEN RSA ALGORITHM AND SRNN ALGORITHM

Table 1 shows the general comparison between RSA and SRNN algorithm. In both algorithms we found that by increasing modulus length n security increase, speed decrease and when chunk size m increases both security and speed increases.

From key generation point of view SRNN algorithm is bit of slower then RSA algorithm. From encryption point of view both algorithms are working almost same .In case of SRNN algorithm only one multiplication operation is additional for each chunk calculation. So when chunk size increases we found both algorithms are giving almost same time.

From decryption point of view SRNN algorithm is much slower then RSA algorithm. Overall performance we found that SRNN algorithm is better in security but slower in speed.following fig. 3.4 shows that when modulus length is increases speed of SRNN algorithm is decreases with respect to RSA algorithm.

Difference of SRNN and RSA with modulus length 1024 bits are approximately 5080 milliseconds (SRNN 1024 bits > RSA 1024 bits) whereas difference of RSA 2048 bits and SRNN 1024 bits are 5338 milliseconds (RSA 2048 bits > SRNN 1024 bits). Hence SRNN with modulus length 1024 bits are is good balance between speed and security.

Table 1: Comparison between RSA and SRNN Algorithm.

S.No.	RSA Algorithm	SRNN Algorithm
1	It uses different key for encryption and decryption	It also uses different key for encryption and decryption
2	It provide digital signature for authentication.	It may also provide digital signature for authentication.
3	It is best suited for multi user environment.	It may also suited for multi user environment.
4	It has less security.	It increased security.
5	Process speed is fast.	Process speed is slow.

Following Table 2 shows the Comparison on the basis of Recorderd Time.

Modulus Length (n)	**Chunk Size (m)**	**RSA Algorithm**				**SRNN algorithm**						**Diff.**
		Key Gen. Time (x)	**Enc Time (y)**	**Dec. Time (z)**	**Total RSA Time (t=x+y+z)**	**Natural Number (Random) u**	**Natural Number (Random) a**	**Key Gen. Time (x1)**	**Enc. Time (y1)**	**Dec. Time (z1)**	**Total SRNN Time t1 = x1+y1+z1**	**SRNN – RSA Time t1-t**
Bits	Bits	MS	MS	MS	MS	MS	MS	MS	MS	MS	MS	MS
256	128	52	58	345	455	147	202	64	97	837	998	543
512	256	170	80	905	1155	83	164	182	115	2274	2571	1416
1024	512	550	118	2700	3368	143	251	589	164	7695	8448	5080
2048	1024	3965	197	9624	13786	151	240	4333	230	28663	33226	19440
4096	2048	26773	369	39780	66922	153	230	57813	390	221363	279566	212644

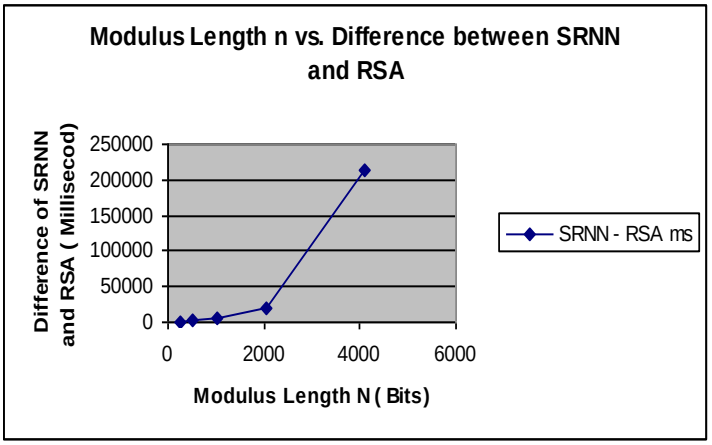

Fig. 1 Modulus Length (n) versus Speed Difference between SRNN and RSA Algorithm

V. CONCLUSION

We have proposed a method for implementing a public-key cryptosystem whose security rests in part on the difficulty of factoring large numbers. If the security of our method proves to be adequate, it permits secure communications to be established without the use of couriers to carry keys.

The security of this system needs to be examined in more detail. In particular, the difficulty of factoring large numbers should be examined very closely. Once the method has withstood all attacks for a sufficient length of time it may be used with a reasonable amount of confidence.

We found RSA with modulus length 1024 bits with chunk size 512 bits is better in speed but some weaker in security . In case of SRNN algorithm if we generate randomly small range of natural numbers **u** and **a** session to session will be a better solution for balance between speed and security. If we take SRNN with modulus length 1024 bits with chunk size 512 is a feasible solution from speed as well as security.In this way SRNN with modulus length 1024 bits may be a optimize solution which make balance between speed and security.

REFERENCES

[1] William Stallings, Cryptography and Network Security, Pearson Education, hird Edition. AD97] M. Ajtai and C. Dwork, A public-key cryptosystem with worst-case/average-case equivalence, Proc.29th ACM STOC (1997), 284-297.
[2] Atul Kahate , Cryptography and Network Security , Tata McGraw-Hill Publishing Company Limited.
[3] RSA Algorithm , DI Management Services is a computer programming consultancy based in Sydney, Australia.
[4] Carnegie Mellon Software Engineering Institute "Public Key Cryptography".
[5] Sensitivity of Changing the RSA Parameters on the Complexity and Performance of the Algorithm By Allam Mousa ; Journal of Applied Science 5 (1) :60-63,2005 ISSN 1607 – 8926.Asian Network for Scientific Information.
[6] An Analysis of Shamir's Factoring Device by Robert D. Silverman , RSA Laboratories ,May 3 , 1999.
[7] KetuFile White Papers "Symmetric vs. Asymmetric Encryption ", a division of Midwest research corporation.
[8] RSA Laboratories : Technical Notes and Papers.
[9] A fast implementation of the RSA algorithm using the GNU MP library. By Rajorshi Biswas,Shibdas Bandyopadhyay,Anirban Banerjee, IIIT – Calcutta.
[10] Wikipedia, the free encyclopedia : Natural Numbers.
[11] Perfomance Comparisons, Desion, and Implementation of RC5 Symmetric Encryption Core using Reconfigurable Hardware By Omar Elkeelany and Adegoke Olabisi .Journal of computers, vol. 3 No 3,march 2008.
[12] Fast implementation of RSA 1993

