Analysis of Encryption Algorithms (RSA, SRNN and 2 key pair) for Information Security

Sarika Y. Bonde[1]
[1]Research Scholar
North Maharashtra University
Jalgaon, (M.S) INDIA
[1]sarika_apatil@rediffmail.com

Prof.Dr. U. S. Bhadade[2]
[2]Professor and Head of IT department
S. S. B. T. College of Engineering & Technology
Bambhori, Jalgaon, (M.S) INDIA
[2]umeshbhadade@rediffmail.com

***Abstract*- To transmit confidential information like e-mails, banking transactions, credit card details etc. over the unsecured network like internet Security is essential. There are more chances to read the information that is being transferred through network of computers or internet by other people. Now a days most of the countries are affected Ransomware attack. To avoid this we need a secure way to protect our data. So to protect information, we need to encrypt/decrypt information by using cryptography algorithms. In cryptography the data which is to be transmitted from sender to receiver in the network must be encrypted using the encryption algorithm and receiver can view the original data by using decryption technique. Mostly used asymmetric key encryption algorithm is RSA, which uses two keys one for encryption and another for decryption. Long time for the encryption process is the disadvantage of RSA algorithm. To overcome this problem, this paper performs comparative analysis of three algorithms: RSA, short range natural number (SRNN) and two key pair algorithm considering parameters encryption / decryption time. The programs are implemented using JAVA library function. The class BigInteger is used to hold large prime numbers and keys so that it difficult for hacker to guess the data. The algorithms are executed successfully on different size of text files. Comparing the encryption time results using normal RSA algorithm, SRNN and two key pair algorithm, it was proved that SRNN algorithm takes less time for encryption.**

Keywords- Cryptography, RSA, encryption, decryption, SRNN, Key pair, BigInteger.

I. INTRODUCTION

In security of data cryptography plays very important role. The meaning of cryptography is to transfer sensitive information across insecure networks like internet so that it cannot be read by anyone except the person whom we want to send it. The information is hides by using cryptography. Two types of cryptography algorithms are Symmetric and Asymmetric key cryptography.

To encrypt and decrypt data, symmetric key cryptography uses same key. Block ciphers and Stream ciphers are the two types of symmetric key cryptography**.** Groups of bits having fixed-length are called as blocks. A block cipher algorithm operated on such blocks, with an unvarying transformation that is specified by a symmetric key. Examples of Block Ciphers are: BLOWFISH, RC2, DES, CAST-128, 3DES and IDEA.

To encrypt the data using stream cipher, cryptographic key and algorithm are applied on one bit at a time in a data stream. Examples of stream Ciphers are: SEAL and RC4.

Private keys and public keys are the two keys which are used in asymmetric key cryptography. For encryption Public key is used is known to the public and for decryption private key is used which is known only to the user. Examples of Asymmetric cryptographic algorithms are: Diffie-Hellman, RSA [1, 2].

A. Organization

This paper has organized into V sections. Section II is RSA Algorithm. Section III presents Literature Survey. Section IV is the implementation. Section V is the experimental results and analysis.

II. RSA ALGORITHM

RSA algorithm is developed by Ronald Rivest, Adi Shamir, and Leonard Adleman in 1978. RSA algorithm is suitable for signing as well as for encryption. The RSA algorithm involves three steps: key generation, encryption and decryption [1,2].

a) Key generation:

RSA involves two keys, public key and a private key. Messages encrypted with the public key can only be decrypted by using the private key. e is released as the public key exponent and d is kept as the private key exponent. The steps for key generation are explained below:

i. Select p and q where $p \neq q$ and both p and q are prime numbers.
ii. Determine $n = p \times q$
iii. Compute $\varphi(n) = (p-1) \times (q-1)$,where φ is Euler's totient function.
iv. Choose an integer e such that gcd $(\varphi(n), e) = 1$ and $1 < e < \varphi(n)$
v. Evaluate d as $d \equiv e^{-1} \pmod{\varphi(n)}$
vi. Public Key (PU) = $\{e, n\}$
vii. Private Key (PR) = $\{d, n\}$

b) Encryption process :

The steps for encryption of message in order to get the cipher-text are explained below :

i. Obtain a plain text M such that $M < n$.
ii. Compute the cipher text as $C = M^e \bmod n$

c) Decryption process:

The steps for decryption of cipher-text in order to get the original message are explained below :

i. Get the cipher text C.
ii. Calculate the plain text as $M = C^d \bmod n$

978-1-5386-4008-1/17/$31.00 ©2017 IEEE

III. LITERATURE SURVEY

To enhance the speed of encryption, decryption process, to increase the level of security modified version of RSA algorithm was developed by many researchers.

Sonal Sharma et al. [3] proposed short range natural number algorithm which is similar to RSA algorithm with some modification. They used two natural numbers in pair of keys (public, private), which increases the security of the cryptosystem. The programs are implemented using Java library functions. From the result it is found that SRNN with modulus length 1024 bits gives good speed and security.

Punita Meelu, Rajni Meelu [4] has implemented RSA using different text size in JAVA Eclipse Platform. The results of encryption and decryptions of RSA are given in seconds. They conclude that encryption requires more time as compared to decryption timing for the same text size over the same key.

K. Sheela, E. George Dharma Prakash Raj [5] has proposed a new algorithm called as InKeSi (increased key size), in which the brute force attack in Existing SRNN (Short Range Natural Number) algorithm can be avoided by increasing the key size. In the proposed algorithm, the key size is increased by 512 bit to 1024 bit in SRNN algorithm which provides 100% security than the 512bit SRNN algorithm i.e. high security.

M.Sreedevi[6]has propose SR_2N algorithm which uses extremely large number that has two prime factors along with two natural numbers in pair of keys. Also gives comparison between SRNN and proposed SR_2N algorithm and found that SR_2N algorithm provides better security with more processing speed. The programs are implemented using java library functions.

R. Mahaveerakannan et.al[7] has proposed Secure Data Transaction Natural Number Algorithm (STNN) and provides an analytical study of RSA, SRNN and STNN algorithm. STNN algorithm is moreover a public key cryptography algorithm with increasing security but processing speed is slow compared to RSA algorithm.

Priyanka P. Koshti et al [8] has used two key pairs to improve efficiency of an existing RSA cryptosystem. For data encryption they used one small size key pair and another large size key for encrypt key component (n=p*q, where p & q are chosen prime numbers) of the small size key pair. The results of encryption and decryptions of two key pairs RSA on different text size was given in milliseconds.

From the literature survey it is concluded that short range natural number algorithm which is modified RSA algorithm gives high safety.

IV. IMPLEMENTATION

A. Implementation of Short Range Natural Number (SRNN) Algorithm:

The SRNN algorithm is modified RSA algorithm[3]. The SRNN algorithm uses extremely large number that has two prime factors (similar to RSA) and two short range natural numbers in pair of keys. This modification increases the security of this cryptosystem. Implementation of SRNN algorithm involves three steps: key generation process, encryption and decryption process [3,5].

a) Key Generation process:

i. 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.
ii. Compute n= p x q.
iii. Compute phi = (p-1) (q-1).
iv. 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.
v. Pick short range natural number u randomly such that u < phi-1.
vi. Pick another short range natural number a randomly such that phi > a > u. And compute u^a .
vii. Find d such that- e * d mod ((p-1)*(q-1)) = 1
viii. 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.

b) Encryption process:

Sender does the following:-

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

c) Decryption process:

Recipient does the following:-

i. Uses his private key (d, a, u) to compute $m = (v^e\ c)^d \bmod n$

Where $v = u^{\text{phi-a}} \bmod n$.

ii. Extracts the plaintext from the integer representative m.

B. Implementation of RSA algorithm using two key pairs:

The two different key pair one of small size (public_key1, private_key1, n1) and another of very large size (public_key2, private_key2, n2) is generated using same existing RSA key generation algorithm. Because in RSA algorithm if we take large size key then its take more time for encryption and decryption operation and if we select small size key then security is compromised . Implementation of two key pair RSA algorithm involves two steps: encryption and decryption process [8].

a) Encryption process:

Step1. Encrypt data with public key of small size key (public_key1)

Step2. Encrypt n1 of small key pair with public key (public_key2) of large key pair.

Step3. Transmit results of step 2 n step3 to receiver.

978-1-5386-4008-1/17/$31.00 ©2017 IEEE

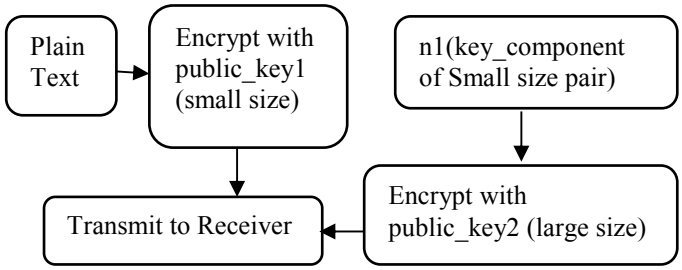

Fig. 1: Encryption process of RSA algorithm using 2 key pair

b)Decryption process:

Step1. First decrypt n1 with private_key2.

Step2. Now we have n1, so we can decrypt encrypt data with private_key1.

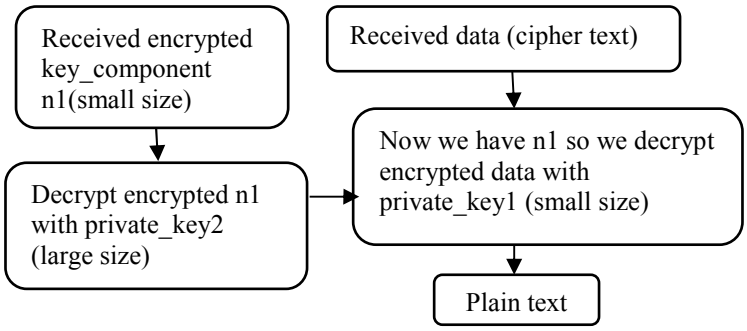

Fig. 2: Decryption process of RSA algorithm using 2 key pair

The programs are implemented using JAVA library functions. The class BigInteger is used to hold large prime numbers and keys so that it difficult for a hacker to guess the data.

V. EXPERIMENTAL RESULTS & ANALYSIS

Experimental results for 512 bit key size and 1024 bit key size RSA, short range natural number (SRNN) and two key pair algorithm are shown in Table1 and Table 2. Encryption time is the time which an algorithm takes to convert plain text to a cipher text. Decryption time is the time which an algorithm takes to get plain text from a cipher text.

Table 1: Comparisons of 512 bit key size RSA, SRNN and two key pair algorithm based on Encryption / Decryption Time (in Milliseconds)

Text Size	Encryption Time (ms)			Decryption Time (ms)		
	RSA	2-key pair	SRNN	RSA	2- key pair	SRNN
1KB	12	27	18	30	81	60
2KB	25	61	24	76	167	133
5KB	72	107	55	268	488	448
10KB	155	111	71	573	1037	854
15KB	246	154	92	802	1318	1218
20KB	323	201	135	1087	1712	3295
25KB	457	226	174	2058	2265	2418
30KB	500	229	177	2132	2606	2475
35KB	571	303	265	2245	3154	2745
40KB	619	309	275	2317	3546	3422
45KB	725	345	296	2402	5557	5593
50KB	769	389	315	2736	6064	4248
55KB	862	403	347	3018	6300	6066
60KB	969	435	388	3259	6714	6530
65KB	1033	448	366	3629	7498	5554
70KB	1058	476	419	5460	7576	6043
75KB	1120	497	465	5376	7656	7641
80KB	1167	502	479	5914	8795	6870
85KB	1255	554	515	4849	9143	7315
90KB	1337	560	543	6389	9620	8204
95KB	1424	596	555	5369	10046	9909
100KB	1468	613	568	7007	10609	10243
150KB	2083	907	617	9286	16516	13513

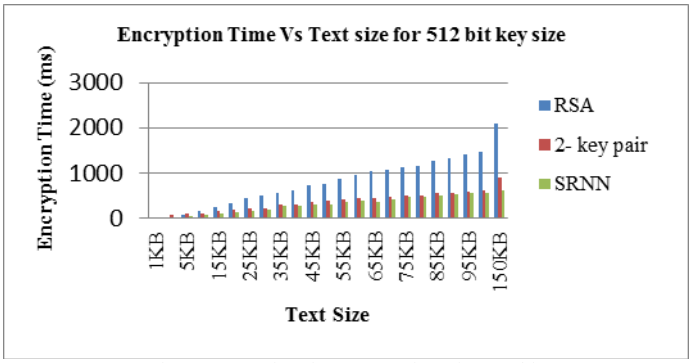

Fig.3: Encryption time comparison for 512 bit

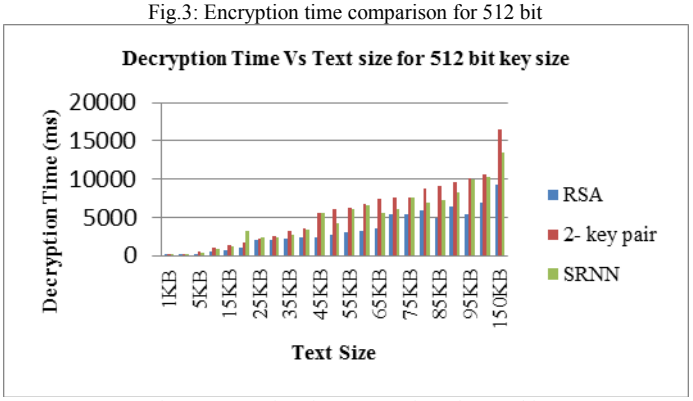

Fig.4: Decryption time comparison for 512 bit

978-1-5386-4008-1/17/$31.00 ©2017 IEEE

Table 2: Comparisons of 1024 bit key size RSA, SRNN and two key pair algorithm based on Encryption / Decryption Time (in Milliseconds)

Text Size	Encryption Time (ms)			Decryption Time (ms)		
	RSA	2- key pair	SRNN	RSA	2- key pair	SRNN
1 KB	35	11	17	119	181	255
2 KB	85	60	26	400	470	585
5 KB	327	97	64	1050	1521	1807
10 KB	624	219	101	2225	3244	3784
15 KB	948	273	149	3302	4659	5479
20 KB	1246	329	179	4486	6309	7310
25 KB	1499	367	197	5628	7831	9234
30 KB	1949	450	246	6831	9671	11040
35 KB	2251	453	410	8030	11066	12858
40 KB	2589	589	405	9390	13433	14658
45 KB	2981	653	428	10212	14180	15860
50 KB	3009	748	460	11236	15851	18240
55 KB	3423	771	475	12426	17268	19755
60 KB	3671	862	499	13505	18877	21828
65 KB	3868	913	539	14889	20228	23596
70 KB	4162	995	561	15893	22239	24990
75 KB	4463	1016	647	16841	23587	26686
80 KB	4981	1148	675	18248	25400	29515
85 KB	5151	1183	667	19772	26907	32138
90 KB	5545	1184	742	21141	28460	33718
95 KB	5970	1238	762	22666	31851	36118
100 KB	6401	1360	766	24124	34282	39406
150 KB	10018	2102	1158	39381	53026	62720

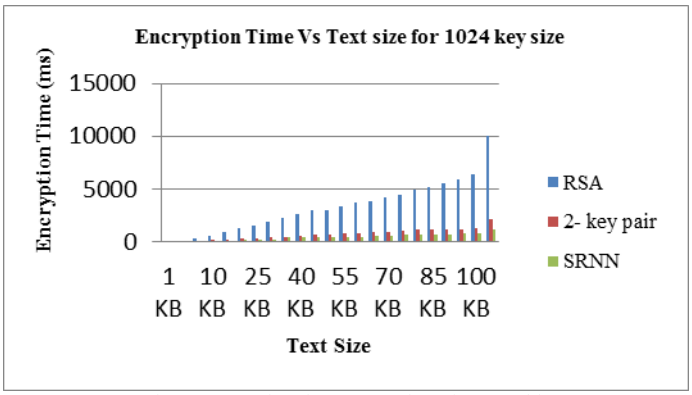

Fig.5: Encryption time comparison for 1024 bit

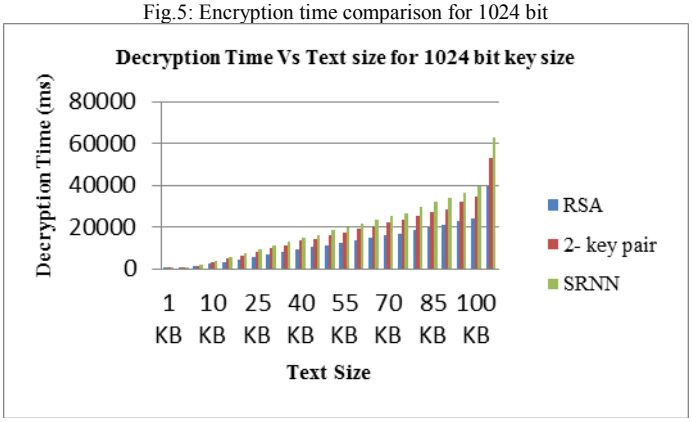

Fig.6: Decryption time comparison for 1024 bit

The outcome of this research work is as given below:

i. To encrypt text, RSA algorithm and 2 key pair algorithm takes longer time than SRNN algorithm. Thus, in relation with encryption speed, SRNN algorithm is better than RSA algorithm and 2 key pair algorithm.

ii. To decrypt text, 2 key pair algorithm and SRNN algorithm takes longer time than RSA algorithm. Thus, in relation with decryption speed, RSA algorithm is better than 2 key pair algorithm and SRNN algorithm.

CONCLUSION

In communication security encryption algorithm plays an essential role where encryption time is the major issue of concern. For performance evaluation RSA algorithm, 2 key pair algorithm and short range natural number (SRNN) algorithm are used. The algorithm successfully executes for different size of text files i.e. 1 KB to 150KB, using JAVA library functions. From the experimental result it was concluded that SRNN algorithm consumes least encryption time and RSA consume longest encryption time. Also RSA consumes least decryption time as compared to SRNN and two key pair algorithm.

REFERENCES

[1] Bruce Schneir, "Applied Cryptography", 2nd edition, John Wiley & Sons, 2007.

[2] William Stallings, "Cryptography and Network Security", Pearson Education, Fourth Edition, 2007.

[3] Sonal Sharma, Jitendra Singh Yadav and Prasant Sharma "Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), ISSN: 2277 128X ,Volume 2, Issue 8, August 2012, pp.134-138.

[4] Punita Meelu,Rajni Meelu, "Implementation Of Public Key Cryptographic System: RSA", International Journal of Information Technology and Knowledge Management, July-December 2012, Volume 5, No. 2, pp. 239-242.

[5] K. Sheela, E. George Dharma Prakash Raj ,"InKeSi-Increased Key Size Method in SRNN Public Key Cryptography Algorithm", International Journal of Computer Science and Mobile Computing (IJCSMC), ISSN 2320–088X, Vol. 2, Issue. 8, August 2013, pp.219 – 223.

[6] M.Sreedevi, "Threshold Sr2n Public key Cryptosystem", International Journal of Engineering Trends and Technology (IJETT), ISSN: 2231-5381, Volume 31, Number 1, January 2016, pp.15-17.

978-1-5386-4008-1/17/$31.00 ©2017 IEEE

[7] R. Mahaveerakannan ,C. Suresh Gnana Dhas, "Customized RSA Public Key Cryptosystem Using Digital Signature of Secure Data Transfer Natural Number Algorithm", International Journal of Computer Technology and Application (I J C T A), 9(5), 2016, pp. 543-548

[8] Priyanka P. Koshti, Dr.U.S.Bhadade, " Implementation and analysis of modifications in RSA algorithm", International Journal of Advance Research in Engineering, Science & Technology (IJARAST), e-ISSN: 2393-9877, p-ISSN: 2394-244 Volume 3, Issue 7, July-2016,pp.46-50

978-1-5386-4008-1/17/$31.00 ©2017 IEEE