Security improvement and Speed Monitoring of RSA Algorithm

Rana M Pir
Lecturer
Leading university, sylhet Bangladesh

***Abstract* - Cryptography includes the science and method to make safe the information. Public key cryptography through the design of key pair-up is a huge penetrates for the conventional cryptography. RSA is one of mainly worn asymmetric key encryption technique. It uses two diverse keys to encrypt and decrypt important to protect broadcast of messages. This investigate job essentially focuses on rising the safety of RSA algorithm by means of two supplementary random standards for the calculation of N, and thus to keep the comparable decryption speediness the value of N formed by only two leading digit is worn. That helps to hold on to the equal decryption hustle. Proposed method will boost the factoring difficulty of the benchmark algorithm up to least 6 times for sure.**

***Key Terms* - RSA, Modular Arithmetic, Cryptography, Cryptosystem, private-key, public-key**

I. INTRODUCTION

Cryptography is a system to cover the data in excess of announcement strait. The urban touchable and hardware equipment are not enough to defend the data starting unauthorized parties. Consequently, the experts and the developers have to construct and expand safety systems, defend the information and to stop the attackers beginning singing with the extremely significant basis. For this motivation, the expression "Encryption" brought elsewhere, and it's the major feature that must be obtainable in defence scheme and get for a genuine procedure to influence and produce the safety scheme. It is a skill to conceal the information from strangers. As the knowledge grows date by date, the requirement of information security above message conduit is augmented to high degree. To secure the information, cryptography is worn, and this crypto classification can be illustrious in two chief types: Secrete-Key Cryptography and Public Key Cryptography.

II. STANDARD RSA ALGORITHM

RSA is one of the largely worn asymmetric input encryption algorithms [9]. It uses numerous keys for encryption and decryption foremost to protected broadcast of communication messages. RSA installation enhanced if cost of the key is extended, as it becomes complicated to outline absent the factors of n. The algorithm includes three dissimilar phases [6]:

Phase 1: Key
Production Phase 2:
Encryption Phase 3:
Decryption

Phase 1: Key Production

RSA includes two keys: public and private key. Public key is worn to encrypt the message and private key is worn to decrypt the message. The key production takes spaces as below:

STEP 1: Choose any two huge prime numbers P and Q.
STEP 2: calculate N by means of the known procedure
N= P * Q

STEP 3: calculate Euler's totient meaning
Ø Ø = (P-1) * (Q-1)
STEP 4: make a decision the public key supporter E so as to
1 < E < Ø and, E and Ø are co-prime
Which revenue that GCD (E, Ø) = 1
STEP 5: decide private key advocate D from side to side
the given method:
D *E= 1*mod (Ø)
This means that D is the multiplicative opposite of E*mod ((Ø).

Now, here the public key includes public key exponent E and N. And private key includes private key exponent D & N.

Public Key: (N, E) Private Key: (N, D)

Phase 2: Encryption

To encrypt any message, this method converts the agreed communication into an numeral number by means of a appropriate stuffing method. Then subsequent method is worn to produce encrypted communication **C**:

C = M ^ E mod (N)

Phase 3: Decryption

Subsequent formula is worn to decrypt the encrypted communication: M = C ^ D mod (N)

Example of RSA Cryptosystem:

Phase 1: Key Production

STEP 1: Get any two big prime statistics P and Q.
Prefer P = 7 and Q = 17

STEP 2: Calculate N by means of the agreed method N= P * Q
N=3*11 = 119

STEP 3: Calculate Euler's totient meaning ∅ N
∅ N = (P-1) * (Q-1)
∅ N = 6 * 16
∅ N = 96

STEP 4: Decide the public key supporter E such that 1 < E < ∅ N and, E and ∅ N are to be co-prime which resources that GCD (E, ∅ N) = 1
Let E = 5

STEP 5: terminate private key fan D from side to side the particular means: D *E≡ 1*mod (∅ N). This method that D is the (multiplicative conflicting of E)*mod ((∅ N).
D = 77 [As ED = 1 mod 96, D = E^{-1} mod 96 = 77]

Public Key is (E, N) = (5, 119) Private Key is (D, N) = (77, 119)

Phase 2: Encryption (Presume Message is M=20, Public Key (7, 33))

Cipher Text determination will be intended from side to surface equation C = M ^ E mod (N)
C= M ^ E mod (N)
C=20^5 mod 119 = 3200000%119 = 90

Phase 3: Decryption (Cipher Text=90, Private Key (77, 119))

M = C ^ D mod (N)
M=90^{77} *mod 119 = 20*

III. SECURE EXECUTION OF RSA ALGORITHM

Secure execution of RSA algorithm by means of two casual figures for the calculation of N value. The similar price is worn to produce E and D price. :

Phase 1: Key Production

STEP 1: Get any of the two big prime figures P and Q., Also get two random figures R1 and R2. STEP 2: Calculate N by means of the known method
N=P*Q*R1*R2

STEP 3: Calculate Eular's totient meaning ∅
∅ = (P-1) * (Q-1)*(R1-1)*(R2-1)

STEP 4: Decide the public key advocate E such that
1 < E < ∅ and, E and ∅ are co-prime
Which earnings with the intention of GCD (E, ∅)

STEP 5: Resolve private key supporter D from side to side the given formula:
D *E= 1*mod (∅)

This means that D is the multiplicative opposite of E*mod ((∅).

Now, here the public key includes public key supporter E and N. And private key includes private key supporter D & N.

Public Key: (N, E) Private Key: (N, D)

Phase 2: Encryption

To encrypt any communication, the algorithm converts the specified communication into numeral figure by means of a appropriate stuffing method. Then subsequent prescription is worn to produce encrypted communication **C**:

C = M ^ E mod (N)

Phase 3: Decryption

Subsequent procedure is worn to decrypt the encrypted communication:
M = C ^ D mod (N)

Example of Secure RSA Algorithm with the two random numbers:

Let's believe that, we contain to transmit a communication whose cost is 10 i.e. m=10.

Phase 1: Key Production

STEP 1: Obtain two prime figures x and y. P=5 and Q=3 Obtain any two casual figures X and Y. X=4 and Y=6
STEP 2: N = (5*3*4*6) => N = 360
STEP 3: ϕ (N) = (5-1)*(3-1)*(4-1)*(6-1) =4*2*3*5 => ϕ (N) = 120
STEP 4: GCD (E, 120) = 1 therefore, E= 7 because its co-prime to 120
STEP 5: 7 * D = 1 mod (120) => D = 103
Public Key = (7,360)
Private Key = (103,360)

Phase 2: Encryption

Encryption of simple text by means of public key mechanism (7,360).
$C=M^E$ mod (N)
$C=10^7$mod (360) = 280

Phase 3: Decryption

Decryption of communication by means of Private Key mechanism (103, 360)
$M=C^D$ mod (N) = C = 280 ^ 103 mod (360) = 10

IV. High Speed and Secure Implementation of RSA Algorithm

RSA includes two keys: public key and private key. Public key is worn to encrypt and private key is worn to decrypt the communication. The key production takes seats as follow:

Phase 1: Key Production

STEP 1: Obtain any two big prime figures P and Q., Also obtain any two casual figures R1 and R2.
STEP 2: calculate N by means of the specified method
N1=P*Q*R1*R2, N2=P*Q
STEP 3: calculate Eular's totient occupation Ø 1
Ø 1 = (P-1) * (Q-1)*(R1-1)*(R2-1)
STEP 4: Choose the public key exponent E such that
1 < E < Ø 1 and, E and Ø 1 are co-prime
Which revenue that GCD (E, Ø 1) = 1
STEP 5: steadfastness private key supporter D from side to side the specified method:
D *E= 1*mod (Ø 1)
This way that D is the multiplicative opposite of E*mod ((Ø).

Now, here the public key includes public key supporter E and N. And private key consists of private key supporter D & N.

Public Key: (E, N1) Private Key: (D, N2)

Phase 2: Encryption

To encrypt any communication, the algorithm converts the specified communication keen on a numeral digit by means of a appropriate stuffing system. Then subsequent method is used to produce encrypted communication **C**:
C = M ^ E mod (N1)

Phase 3: Decryption

Subsequent method is worn to decrypt the encrypted communication: M = C ^ D mod (N2)

Example of RSA Algorithm with two random numbers with two different N Values:

Let's think about that, we contain to post a communication whose worth is 10 i.e. m=10. Similar as preceding instance.

Phase 1: Key Phase 1: Key Production

STEP 1: Obtain any two prime figures x and y. P=5 and Q=3 obtain any two random numbers X and Y. X=4 and Y=6
STEP 2: N1 = (5*3*4*6) => N1 = 360 | N2 = (5*3) => N2 = 15

STEP 3: ϕ (N) = (5-1)*(3-1)*(4-1)*(6-1) =4*2*3*5 => ϕ (N) = 120
STEP 4: GCD (E, 120) = 1 Thus, E= 7 because its co-prime to 120 STEP 5: 7 * D = 1 mod (120) => D = 103
Public Key = (7,360)
Private Key = (103, 15)

Phase 2: Encryption

Encryption of simple text by means of public key mechanism (7,360).
$C=M^E$ mod (N1)
$C=10^7$mod (360) = 280

Phase 3: Decryption

Decryption of communication by means of Private Key mechanism (103, 15)
$M=C^D$ mod (N2) = C = 280 ^ 103 mod (15) = 10

V. IMPLEMENTATION

The projected algorithm is implemented using PHP coding speech. The obtained results are as follows.

Table 1 Encryption and Decryption Time using Standard RSA Algorithm

P	Q	N	Phi(N)	E	D	E.T. (µs)	D.T. (µs)
1359833.000	1359833	1849145787889	1849143068224	3649134810816461	952143487749	0.112	0.094
2190971.000	2275067	4984605820057	4984601354020	3649134810816461	2230804116541	0.115	0.113
3306151.000	3399131	11238040354781	11238033649500	3649134810816461	1150182813641	0.137	0.104
4276829.000	4999307	21381181157503	21381171881368	3649134810816461	4746947729389	0.134	0.081
5788127.000	5792651	33528599654677	33528588073900	3649134810816461	14600829753941	0.112	0.094
6613231.000	6998177	46280561079887	46280547468480	3649134810816461	26767682055941	0.122	0.123
7002257.000	7414711	51919712002727	51919697585760	3649134810816461	20218262294981	0.100	0.094
8462089.000	8964113	75855122012057	75855104585856	3649134810816461	6051306844805	0.111	0.093
9850081.000	9865267	97173679036627	97173659321280	3649134810816461	24676700472581	0.102	0.107
10757543.000	11129113	119721911649359	119721889762704	3649134810816461	38674667625845	0.121	0.135

Table 2 Encryption and Decryption Time using Secure RSA Algorithm

P	Q	R1	R2	N	Phi(N)	E	D	E.T. (µs)	D.T. (µs)
1359833	1350403	45	90	3718553189465470	7191028542249020	3649134810816460	1116482470298110	0.108	0.165
2190971	2275067	88	36	38600787470521400	15178111122990900	3649134810816460	15080649900027000	0.128	0.124
3306151	3399131	74	92	1123804035478100	11238040354781	3649134810816461	2788182527889641	0.133	0.273
4276829	4999307	83	23	123497702365737328	21381181157503	3649134810816461	2976729839239541	0.134	0.271
5788127	5792651	10	84	45900652927252813	33528599654677	3649134810816461	51916855168151141	0.116	0.216
6613231	6998177	61	63	303646761245138607	46280561079887	3649134810816461	127761078695060741	0.103	0.319
7002257	7414711	95	95	284312342926933052	51919712002727	3649134810816461	95500542122507621	0.116	0.205
8462089	8964113	10	84	522565935541060673	75855122012057	3649134810816461	49539434601408773	0.121	0.211
9850081	9865267	76	44	242934197591567000	97173679036627	3649134810816460	166385981458503000	0.110	0.209
10757543	11129113	37	48	52797363037367319	119721911649359	3649134810816461	53793803171079941	0.115	0.211

Table 3 Encryption and Decryption Time using High Speed and Secure RSA Algorithm

P	Q	R1	R2	N1	N2	Phi(N)	E	D	E.T. (µs)	D.T. (µs)
1359833	1350403	61	63	6832956255802979	6831109851166080	1836322562699	3649134810816461	2027297117120261	0.106	0.0857
2190971	2275067	95	95	44986067526014425	44043937564120720	4984605820057	3649134810816461	21266540180365861	0.118	0.1023
3306151	3399131	10	84	1123804035478100	8394811136176500	11238040354781	3649134810816461	2788182527889641	0.121	0.0973
4276829	4999307	76	44	123497702365737328	68954279317411800	21381181157503	3649134810816461	2976729839239541	0.118	0.0971
5788127	5792651	37	48	45900652927252813	56730371021038800	33528599654677	3649134810816461	51916855168151141	0.108	0.1159
6613231	6998177	81	49	303646761245138607	177717302278963200	46280561079887	3649134810816461	127761078695060741	0.116	0.119
7002257	7414711	74	92	284312342926933052	344902551062203680	51919712002727	3649134810816461	95500542122507621	0.108	0.105
8462089	8964113	83	23	522565935541060673	136842608672884224	75855122012057	3649134810816461	49539434601408773	0.116	0.111
9850081	9865267	50	49	242934197591567000	228552446723650000	97173679036627	3649134810816460	166385981458503000	0.103	0.109
10757543	11129113	21	27	52797363037367319	62255382676606080	119721911649359	3649134810816461	53793803171079941	0.116	0.131

Table 4 Average Encryption and Decryption Time

Average Time	Encryption Time	Decryption Time
Standard RSA	0.117(µs)	0.104
Secure RSA	0.118(µs)	0.220
High Speed and Secure RSA	0.113(µs)	0.107

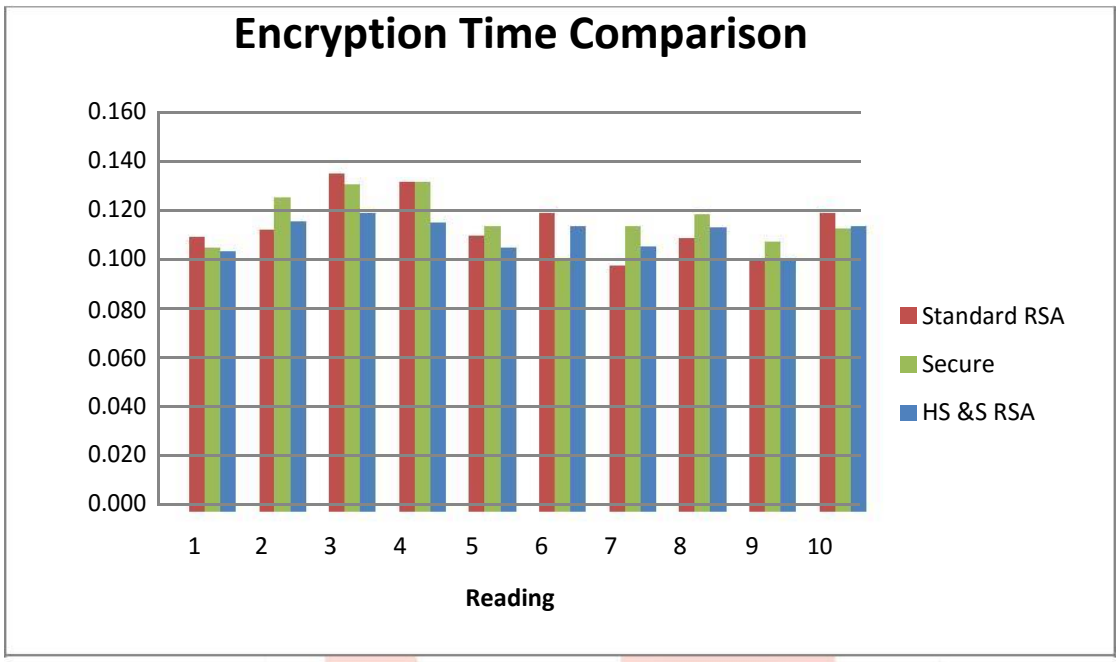

Figure 1: Graph: Encryption Time Comparisons

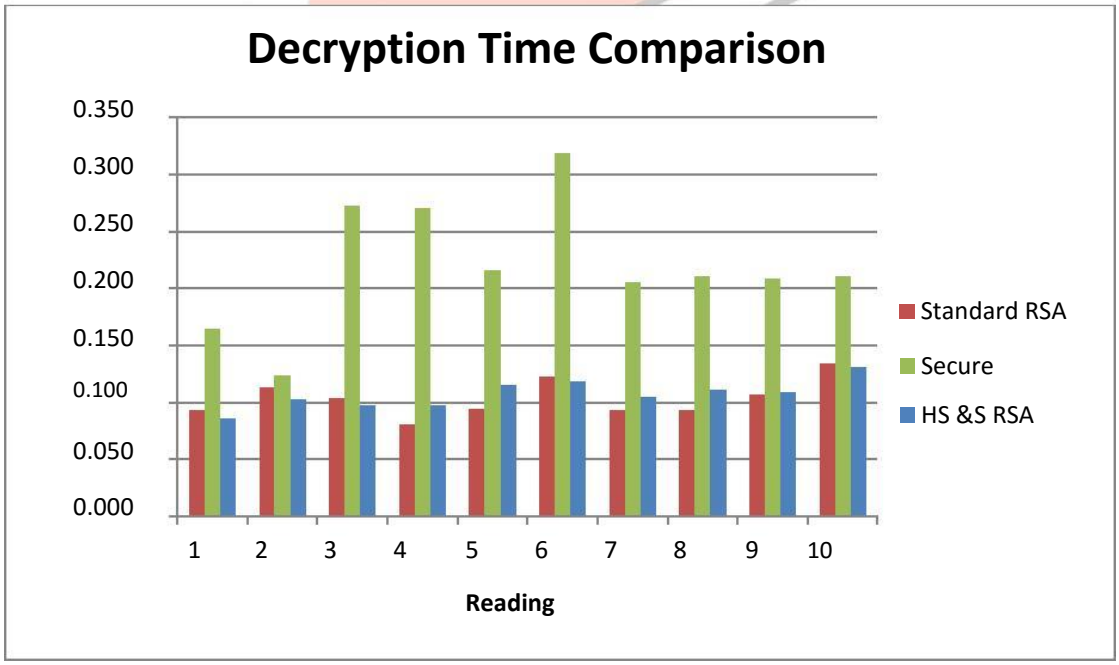

Figure 2: Graph: Decryption Time Comparisons

VI. Analysis

How the safety is amplified?

A Standard RSA Algorithm is protected sufficient, except for the factorizing procedure of N; it gives P, Q as productivity. While in the container of customized RSA algorithm, the N (in public key module) is a mixture of P, Q, R1, R2, attractive the most excellent case, allowing for R1 and R2 prime figures, we can contain N as a masterpiece of four prime integer. Now if we are applying factoring assault and at take it simple talented to locate all the issue of N, we would have 4 factors, which will effect in six single duos of them, which will amplify the difficulty of RSA algorithm six periods. If N=P*Q next we willpower get solitary pair of prime figure which will create aggressive task simple. But in trunk of N=P*Q*R1*R2, we will encompass all probable grouping of RSA prime pairs such as: P*Q, P*R1, P*R2, Q*R1, Q*R2, R1*R2. This is straight away the holder if we encompass R1 and R2 as prime, but as we include selected R1 and R2 as casual figures, it will augment the safety of RSA algorithm at enhanced stage.

How the speed is maintained?

As here we preserve observe that the protected RSA algorithm increases the hurry to smallest amount six periods than the unique RSA algorithm, but it also causes reduce in decryption hurry. Therefore it can be augmented by means of the N price generated by increase of P and Q, which also gives accurate production and accurate decrypted manuscript.

VII. Conclusion

Here in this research paper, we surveyed diverse methods customized by a variety of researchers and scholars for quicker execution of RSA algorithm. They had worn diverse techniques and methodologies in instruct to accomplish high speed execution of RSA algorithm. As we can notice that the protected RSA algorithm increases the speediness to least six periods than the inventive RSA algorithm, but it also causes decline in decryption speediness. So it can be improved by means of the N price generated by development of P and Q, which also gives accurate harvest and accurate decrypted manuscript.

References

[1] William Stallings, "Cryptography and Network Security", ISBN 81-7758-011-6, Pearson Education, Third Edition

[2] H. Ren-Junn, S. Feng-Fu, Y. Yi-Shiung and C. Chia-Yao "An efficient decryption method for RSA cryptosystem" Advanced Information Networking and Applications, 2005 (AINA 2005). 19th International Conference on, 2005, pp. 585-590 vol.1.

[3] Nagar, S.A.; Alshamma, S., "High speed implementation of RSA algorithm with modified keys exchange," Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2012 6th International Conference on , vol., no., pp.639,642, 21-24 March 2012

[4] Selby, A.; Mitchell, C., "Algorithms for software implementations of RSA," Computers and Digital Techniques, IEE Proceedings E , vol.136, no.3, pp.166,170, May 1989

[5] Dhananjay Pugila, Harsh Chitrala, Salpesh Lunawat, P.M.Durai Raj Vincent "An efficeient encrpytion algorithm based on public key cryptography",IJET ,Vol 5 No 3 Jun-Jul 2013, pp. 3064-3067

[6] Atul Kahate, Cryptography and Network Security, ISBN-10:0-07-064823-9, Tata McGraw Hill Publishing Company Limited, India, Second Edition, pages 38-62,152-165,205-240.

[7] R. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signature and Public-Key Cryptosystems, Communication of the ACM, Vol.21, No.2, 1978, pp. 120-126.

[8] Wang Rui; Chen Ju; Duan Guangwen, "A k-RSA algorithm," Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on , vol., no., pp.21,24, 27-29 May 2011

[9] Hung-Min Sun and Mu-En Wu, "Design of Rebalanced RSA-CRT for Fast Encryption," In Proceedings of Information Security Conference 2005 (ISC 2005), pages 16-27, June 2005, (Best Paper Award, the only one out of 58 papers)

[10] Yadav, Prasant Singh, Pankaj Sharma, and Dr KP Yadav. "Implementation of RSA algorithm using Elliptic curve algorithm for security and performance enhancement" International Journal of Scientific & Technology Research Vol 1.

[11] Sharma, Sonal, Jitendra Singh Yadav, and Prashant Sharma. "Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm." International Journal 2.8 (2012).

[12] J. J. Quisquater and C. Couvreur, "Fast decipherment algorithm for RSA public key cryptosystem," Electronic Letters, vol. 18, pp.905-907, 1982.

[13] M. J. Wiener, "Cryptanalysis of RSA with short secret exponents," IEEE Transactions on information Theory, IT-36, pp.553-558, 1990.

[14] P. Kornerup, "A systolic, linear-array multiplier for a class of right-shift algorithms", IEEE Transactions on Computers, vol. 43, no. 8, pp. 892–898, Aug. 1994.

[15] C. D. Walter, "Systolic modular multiplication", IEEE Transactions on Computers, vol. 42, no. 3, pp. 376–378, Mar. 1993.

[16] M. Shand and J. Vuillemin, "Fast implementation of RSA cryptography", in Proc. 11th IEEE Symp. Computer Arithmetic, Windsor, Ontario, June 1993, pp. 252–259.

[17] P.-S. Chen, S.-A. Hwang, and C.-W. Wu, "A systolic RSA public key cryptosystem", in Proc. IEEE Int. Symp. Circuits and Systems (ISCAS), Atlanta, May 1996, pp. 408–411.

[18] C.-Y. Su, S.-A. Hwang, P.-S. Chen, and C.-W. Wu, "An improved Montgomery algorithm for high-speed RSA public-key cryptosystem", IEEE Trans. VLSI Systems, vol. 7, no. 2, pp. 280–284, June 1999.

[19] J.-H. Hong, P.-Y. Tsai, and C.-W.Wu, "Interleaving schemes for a systolic RSA public-key cryptosystem based on an improved Montgomery's algorithm", in *Proc. 11th VLSI Design/CAD Symp.*, Pingtung, Aug. 2000, pp. 163–166.

[20] P. L. Montgomery, "Modular multiplication without trial division", Math. Computation, vol. 44, pp. 519–521, 1985.

[21] [illegible]