Study on Improvements in RSA Algorithm

[1]Sarthak R Patel, [2]Prof. Khushbu Shah, [3]Gaurav R Patel
[1,3]PG Scholar, [2]Assistant Professor
Computer Engineering Department, LJIET, Gujarat Technological University, Ahmedabad
[1]patelsarthakr@gmail.com , [2]khushburana1@gmail.com , [3]gauravpatel9092@gmail.com

Abstract: **In Public key cryptography two different keys (a pair of keys) are used, one for encryption and other for decryption. Main advantage of this technique is that no other one can decrypt the text/message without this key pair. This paper surveys various Improvements done on RSA algorithm by applying various modifications in order to enhance it. RSA is highly secure algorithm but have high computation time, so many researchers applied various techniques to enhance the speed of an RSA algorithm by applying various logic. This paper does the detailed study about various techniques and represents the summarized results.**

Key Terms: RSA, CRT, SRNN, K-RSA, GCD, Modular Arithmetic, Cryptography, Cryptosystem, private-key, public-key.

I. INTRODUCTION

Cryptography is a technique to hide the data over communication channel. It is an art to hide the data to strangers. As the technology grows day by day the need of data security over communication channel is increased to high extent. For securing the knowledge cryptography is used, and this cryptosystem can be distinguished in two major types: Secrete-Key Cryptography and Public Key Cryptography.

Secrete-Key Cryptography (also known as symmetric key, shared-key, single-key, and private-key or one-key encryption.) uses the same key for both encryption and decryption. This technique have lower computational cost but at the same time have many drawbacks like distribution of keys, requirement of a shared secret key, non-repudiation, authentication.

The Public key cryptography (also known as public key encryption) uses two different keys to encrypt and decrypt the message: a public key and a private key [1]. The public key is made publicly available and can be used to encrypt messages. The private key is kept secret and can be used to decrypt received messages. RSA is one of most used asymmetric key encryption algorithm [9]. RSA uses multiple keys for encryption and decryption leading to secure transmission of messages. RSA works better if value of the key is long, as it becomes difficult to figure out the factors of n. RSA algorithm involves three different phases [6]:

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

Phase 1: Key Generation

RSA involves two keys public key and private key. Public key is used for encryption and private key is used for decryption of message. The key generation takes places as follows:

STEP 1: Take any two large prime numbers P and Q.
STEP 2: Compute N by using the given formula
$N= P * Q$
STEP 3: Compute Eular's totient function $\emptyset(N)$
$\emptyset(N)= (P-1) * (Q-1)$
STEP 4: Choose the public key exponent E such that
$1 < E < \emptyset(N)$ and, E and $\emptyset(N)$ are co-prime
Which means that GCD (E, $\emptyset(N)$) = 1
STEP 5: Determine private key exponent D through the given formula:
$D *E= 1*\text{mod} (\emptyset(N))$
This means that D is the multiplicative inverse of
$E*\text{mod} ((\emptyset(N))$.

Now, the public key consists of public key exponent E and N. And private key consists of private key exponent D & N.
Public Key: (N, E) **Private Key: (N, D)**

Phase 2: Encryption

For encrypting any message, the algorithm converts the given message into an integer number by using a suitable padding scheme. Then following formula is used to generate encrypted message **C**:
$C = M ^\wedge E \text{ mod } (N)$

Phase 3: Decryption

Following formula is used to decrypt the encrypted message:
$M = C ^\wedge D \text{ mod } (N)$

II. Related Work

A. *Use of Crypto-Coprocessor and a True Random Number Generator*

Bahadori M. implemented novel approach for secure and high speed implementation of RSA algorithm, by implementing on a typical Smartcard equipped with a crypto-coprocessor and a true random number generator. An efficient method for generating the large random prime numbers significantly reduces the total time required for generating a key pair. Algorithm achieved up to 50% reduction in total generation time compared to the latest reported methods. [2]

B. *Decryption method base on CRT and strong prime of RSA criterion*

Ren-Junn Hwang and Yi-Shiung Yeh proposed an efficient method to employ RSA decryption algorithm. RSA cryptosystem has to perform modular exponentiation with large exponent and modulus for security concern. The RSA cryptosystem takes great computational cost. In many RSA applications, user uses a small public key to speed up the encryption operation. However, the decryption operation has to take more computational cost to perform modular exponentiation by this case. Ren-Junn proposed an efficient decryption method not only based on Chinese Remainder Theorem (CRT) but also the strong prime of RSA criterion. The proposed decryption method was taking 10% computational costs of the conventional decryption method. It also reduces around 66% computational costs than that of decryption methods based on CRT only. In a word, the speed of proposed method is almost 2.9 times faster than the decryption method based on CRT only. The proposed method enhances the performance of the RSA decryption operation [3].

C. *RSA algorithm with modified keys exchange*

Sami A. Nagar and Saad Alshamma speedup the RSA algorithm through a new generation keys method called *RSA-Key Generations Offline* to generate and save all keys values in tables within database. They proposed four security levels, in which each level has its own database and numbers of sets, these levels identified according to the *e* values and key length, before using the RSA algorithm between gateways must get a *Ready Acknowledgment* from RSA Handshake Database protocol, this protocol is responsible for creation or update the identical gateways database, level selections and establishment the algorithm between gateways. Nagar and Alshamma proposed a new method of keys exchange to increase the difficulty for any one knows the exchanged values between gateways, and then try to get the n, e and d values, This approach was known as Concept of Keys Exchange, where algo exchanges the indexes Nid, Eid, Did instead of n, e, d values.

D. *A systolic RSA Based on a Modified Montgomery's algorithm and CRT Technique*

Hsien Wu, Hong, and Wen Wu presented implementation of a systolic RSA cryptosystem [22] based on a modified Montgomery's algorithm and the Chinese Remainder Theorem (CRT) technique. Montgomery's algorithm [23] needs n iterations in each modular multiplication and two additions per iteration, where n is the word length. Cellular arrays based on Montgomery's algorithm can be found in [16–18]. In this algorithm, odd modulus is assumed. It calculates the modular multiplication without performing division. Some improved algorithms were later proposed, either by reducing the number of iterations or by removing the final modular reduction step. A modified Montgomery's algorithm was reported in [19], where the multiplication and modular reduction steps in Montgomery's algorithm are separated such that only one addition is required in each iteration. However, the number of iterations in the modified algorithm is two times that of Montgomery's, hence the overall computation time is not reduced. In [20, 21], the algorithm was further modified to reduce the number of iterations, doubling the speed of modular multiplication. Another way to reduce the computation time is to use the Chinese Remainder Theorem (CRT) technique, since CRT is known to reduce the RSA computation by a divide-and-conquer method. The CRT technique improves the throughput rate up to four times in the best case. The processing unit of the systolic array has 100% utilization because of the proposed block interleaving technique for multiplication and square operations in the modular exponentiation algorithm. For 512-bit inputs, the number of clock cycles needed for a modular exponentiation is about 0.13M to 0.24M. The critical path delay is 6.13ns using a 0.6μm CMOS technology. With a 150 MHz clock, can achieve an encryption/decryption rate of about 328 to 578 Kb/s. [8]

E. *Concept of K^{th} Residue*

Wang Rui ,Chen Ju ,Duan Guangwen developed k-RSA algorithm in which the idea of k^{th} power residue theory and RSA algorithm were combined. This algorithm not only inherits the advantage of RSA, whose security depends on the difficulties of factoring large integers and finding discrete logarithms, but also had high flexibility of parameters. It is designed for improved security and had agreed balance between speed and space. At the same time, it can realize functions like hierarchical system management, secret sharing and so on. The result shows that, in the case of e equality, d in k-RSA algorithm is smaller. And that's why new algorithm can largely reduce the computation time of decryption.[10]

F. *Designing two variants of Rebalanced RSA-CRT*

Based on the Chinese Remainder Theorem (CRT), Quisquater and Couvreur proposed an RSA variation [14], RSA-CRT, to speed up RSA decryption. Then, Wiener suggested another RSA variant, Rebalanced RSA-CRT[15], to further accelerate RSA-CRT decryption by shifting decryption cost to encryption cost. However, such an approach makes RSA encryption time-consuming because the public exponent e in Rebalanced RSA-CRT is of the same order of magnitude as φ(N). There was a need to find a secure variant of RSA-CRT algorithm whose public exponent *e* is much shorter than the function φ(N). Hung-Min Sun and Mu-En Solved this problem by designing two variants of Rebalanced RSA-CRT, *Scheme A* and *Scheme B*. In *Scheme A*, they focused on designing a variant in which dp and dq are of 160 bits, and e = 2567 +1. Thus the encryption time is reduced to about 2.7 of the time required by the original Rebalanced RSA-CRT. While *Scheme B* is a variant in which dp and dq are of 198 bits, and e = 2511 +1. Thus its encryption is about 3 times faster than that of Rebalanced RSA-CRT, but the decryption is a little slower than that of Rebalanced RSA-CRT [11].

G. Using Short Range Natural Number(SRNN) Algorithm

The Short Range Natural Number (SRNN) algorithm [13] is similar to RSA algorithm with some modification, with enhanced security of the cryptosystem. In this algorithm we have an extremely large number that has two prime factors (similar to RSA). Moreover Algorithm uses 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" that used short range in both algorithms they 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, both algorithms are giving almost same time. From decryption point of view SRNN algorithm is much slower then RSA algorithm. Overall performance of SRNN algorithm is better in security but slower in speed. 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.

III. SUMMARIZED RESULTS

Sr#	Approach/Method Used	Remarks	Scope of Improvements
A	Use of Crypto-Coprocessor and a True Random Number Generator[2]	• 50% reduction in total of Key Generation time. • Less than 10% of total time for generating a key pair is assigned to perform the main processor tasks and the other 90% is for the crypto-coprocessor tasks.	
B	Decryption method based on CRT and strong prime of RSA criterion[3]	• Only takes 10% computational costs • Reduces 66% computational costs than that of decryption methods based on CRT only • 2.9 times faster than the decryption method based on CRT only.	• Method can be applied to signing phase of digital signature
C	RSA algorithm with modified keys exchange[4]	• Storing Key Indexes in Database depending on security level. • Exchange of the indexes Nid, Eid, Did instead of n, e, d	
D	A systolic RSA cryptosystem based on a modified Montgomery's algorithm and CRT Technique[8]	• Can achieve an encryption/ decryption rate of about 328 to 578 Kb/s.	
E	Concept of Kth Residue.[10]	• This algorithm not only inherits the advantage of RSA, whose security depends on the difficulties of factoring large integers and finding discrete logarithms, but also has high flexibility of parameters.	
F	Designing two variants of Rebalanced RSA-CRT[11]	• In Scheme A : Encryption time is reduced to About 2.7 of the time required by the original Rebalanced RSA-CRT • In Scheme B: Encryption is about 3 times faster than that of Rebalanced RSA-CRT, but the decryption is a little slower than that of Rebalanced RSA-CRT	• Key generation in this scheme is slow (289 seconds in avg) due to factorization. Can be improved by Algorithm optimization.
G	Using Short Range Natural Number(SRNN) Algorithm[13]	• We have an extremely large number that has two prime factors (similar to RSA). • Moreover Algorithm uses two natural numbers in pair of keys (public, private) .These natural numbers increase the security of the cryptosystem	

IV. CONCLUSION

In this paper, we surveyed different methods modified by various researchers and scholars for faster implementation of RSA algorithm. They used various techniques and methodologies in order to achieve high speed implementation of RSA algorithm.

V. ACKNOWLEDGEMENT

The authors would like to thank anonymous reviewers for their valuable comments and suggestions on the part of security analysis, which certainly led to improvements of this correspondence.

REFERENCES

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

[2] M. Bahadori, M. R. Mali, O. Sarbishei, M. Atarodi and M. Sharifkhani “A novel approach for secure and fast generation of RSA public and private keys on Smartcard” NEWCAS Conference (NEWCAS), 2010 8th IEEE International, 2010, pp. 265-268.

[3] 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.

[4] 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

[5] 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

[6] 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

[7] 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.

[8] Chung-Hsien Wu; Jin-Hua Hong; Cheng-Wen Wu, "RSA cryptosystem design based on the Chinese remainder theorem," Design Automation Conference, 2001. Proceedings of the ASP-DAC 2001. Asia and South Pacific , vol., no., pp.391,395, 2001

[9] 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.

[10] 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

[11] 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)

[12] 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.

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

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

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

[16] 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.

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

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

[19] 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.

[20] C.-C. Yang, T.-S. Chang, and C.-W. Jen, “A new RSA cryptosystem hardware design based on Montgomery’s algorithm”, IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, no. 7, pp. 908–913, July 1998.

[21] 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.

[22] 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.

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