CMT-RSA: Secure and Fast Encryption Technique By Using RSA with Mersenne Twister

Pavan Kumar Illa[1] , Department of Information Technology, VNRVJIET

Swathi Sambangi [2],Asst Professor, Department of Information Technology ,VNRVJIET.

Abstract

Worldwide web is the fastest growing service that allows us to share and access the information from anywhere by anyone. This factor about availability of information on the web services and its access through internet application questions to provide security of the data that we share and access. Hence there the need of building a secured cryptosystem over the data transmission lead to invention of so many public key cryptosystem in which RSA is a major one. This cryptosystem is widely used to provide security services to preserve the privacy policies of any application where as strongest RSA depends on factors of randomness of the large prime numbers used and their integer factorization to break the system. Considering the generation of random large prime numbers as a key factor, we propose a new version of a cryptographically secured RSA for public key cryptosystem. The usage of Crypt Mersenne Twister in the proposed algorithm increases the key generation time with large public key component. This paper also focuses on decreasing encryption time and decryption time when it is compared with other algorithms.

KeyWords : Public key Cryptosystem ,CMT, primality test, CRT

I.INTRODUCTION

The world has globalized in such a way that everything is digitalized and every individual has to communicate with one another through a network, even for a simple day to day activity. It is a well-known fact that the communication between humans through any network has to be safeguarded with security. Over the past years, there has been continuous research is going on to provide security through several channels. The study of algorithms that provide security over the network to maintain secret communication between involved parties is termed as Cryptography.

Every cryptographic model which is built to provide security has to maintain three main components over the communication parties: Confidentiality, Integrity and Authentication. Confidentiality makes the data to be invisible for the other parties who are not authorised over the network for communication. Integrity ensures the truthfulness of the data that is transferred from one person to other person. Authentication is a main component that checks whether the information is passed from an authorised person or not. These three factors play a major role on cryptographic model.

Here in this paper we want to work with an algorithm that provides confidentiality. It's a well known fact that confidentiality can be achieved by encrypting the data and transferring the encrypted data to other party. The information which is to be transferred is in the form of plain text turns to cipher text after the end of encryption process. The cipher text

is again converted to plain text with the process of decryption. It is clear that the encryption process and decryption process run with a combination of key pairs. One who wants to encrypt and decrypt the data has to give plaintext, type of algorithm to be used and encryption key or decryption key as inputs to the cryptographic model that is adopted for their secret communication. Here a single key can be used for both encryption and decryption or two keys can be used, one for encryption and another for decryption. The single key encryption is termed as symmetric encryption and two key encryption is known as asymmetric encryption. It's been observed that the encryption algorithm involves three steps: Key generation phase, Encryption and Decryption.

RSA is one of the major types of cryptographic algorithm which has been introduced in mid 1970s. It is widely known popular cryptosystem for its complex secure approach towards factoring modulo of N. The research related to RSA constitutes a relatively various variants of RSA so far. In this paper we are working on a RSA on its improvement with respect to security and time. In our work we are much more concerned to focus on working of algorithm with large numbers and its key generation time. We introduce a variant of RSA that uses CryptMT for random number generation as randomness of prime number in RSA plays a major role as far as the security is concerned.

The rest of the sections in this paper are organised as follows. Section gives the research study on various RSA algorithms that were introduced till now. Section 3 gives a brief explanation about the proposed algorithm and Section 4 gives the implementation details of the algorithm, Result analysis with parameters encryption time, decryption time and key generation time is analysed in section 5. In section conclusion and future work is specified in section7.

II. Literature Survey

With the strong security nature of RSA, it's a most popular algorithm used in many applications not only for the purpose of public key cryptography but also for the purpose of creating digital signatures. Usage of RSA crypto system in any application makes us to be free from sharing or exchange of private key independently.

The RSA cryptosystem works in such a way that, sender sends cipher text (obtained by the process of encryption) to receiver without sending sender's private key but by using receiver's public key. After receiving message from sender, receiver will decrypt the cipher text with his/her private key which is not disclosed in the network anywhere through the communication.

Since the time of invention of RSA, a wide research has been continuing to process RSA to increase the strength of its security level. There were several variants of RSA algorithm has invented by researchers so far and this section we are going discuss the nature of the variants of RSA and their approach to make the RSA cryptosystem to be more secure.[11]

Amos Fiat(1996) has presented a new approach of RSA cryptosystem namely Batch RSA. Batch RSA specifically works with two properties: one is smaller exponential cost for private key operation than the previous theoretical approaches. This scheme is proposed to be a fast variant of RSA as performs numerous modular operations. Another property of Batch RSA is it uses a process that detach the private key from the crypto system. And it does not depend on the system size , sites used , or the private operations performed.[12]

Thomas et al ., (1996) has filed a patent that presents the new methodology and an equipment that increase the speed of RSA relatively with computation of operations. Usually RSA cryptosystem works with two prime and calculates n .Where the proposed scheme works with three or more dissimilar prime numbers to calculate n.[13]

T.Takagi (1998) has proposed RSA cryptosystem depending on public key primitive modulo. This algorithm mainly concentrates on the speed of decryption process that uses CRT(Chinese remainder theorem) and also RSA using multi-prime factors. The proposed scheme of this algorithm uses Hensel lifting to caluculate modulous operations. They also discussed the key factors that are improved to make RSA much faster than previous variants.[14]

Krishnamurthy et al. (2003) have given a methodology with some modifications in the original RSA algorithm. This approach uses multi prime concept where as the original RSA uses only two distinct primes.Upon the change of number of primes , they also proposed an algorithm namely Montgomery reduction for the purpose of reduction in number of multiplications to be performed and squaring method to perform in an most effective way to speed up the computation [15].

Mu-En Wu et.al (2014) have proposed a new scheme of RSA to with stand over the wiener attack when the public key exponent ‘e’ is small. The difficulty in finding the factorization of n that depends on prime numbers makes the RSA to be harder. Their approach has contributed coast of exhaustive search over the mentioned attack and they named a method called EPF that reduce the cost of search[16].

Ravi Shankar Dhakar et al (2012) has presented a modified version of RSA for the improvement of RSA. The scheme proposed is based on additive homomorphic properties and their they named it as MREA(Modified RSA encryption algorithm). When it is compared with traditional RSA the results were shown that MREA is comparatively secure than traditional RSA[17].

Zulkarnain Md Ali et al. (2013) projected an RSA algorithm with EIGamal Cryptosystems that are used for encryption and decryption. The original RSA algorithm security solely depends upon the complication of compiling the factorization of large prime numbers. Whereas RSA that developed with EIGamal scheme depends on the computation problem related to discrete logarithms. They have performed comparative analysis of original RSA cryptosystem with their proposed scheme[18].

Yunfei Li et al. (2010) proposed a methodology with an objective to reduce the decryption process time and the method was named as Encrypt Assistant Multi Prime

RSA(EAMRSA) . They tried to reduce the decryption time by the reduction of private exponential values during the calculation of modular exponentiation[19].

Aayush Chhabra et al. (2011) defined one more variant of RSA cryptosystem where the modifications were done to original RSA in the phase of key computation aiming for enhanced security but this approach has not reached to the expectations in terms of results compared with original RSA[20].

Deepak Garg et al. (2009) proposed a public key cryptosystem using RSA to reduce execution time as it is running on many web applications. They have proposed a new variant of RSA by adopting the combined scheme of multi-power RSA with rebalanced RSA. The authors have significantly tried for the effective reduction with respect to execution time[21].

III. Existing Methodologies

The research review discussed above sections shows that there are lot more updations and variations of RSA to improve its performance in terms of security and efficiency. In this work we want to examine the results with respect to the classical RSA algorithm and an Enhanced and Secured RSA key generation scheme in detail. The working principle behind these two algorithms is explained with the help of numerical examples in the following section.

3.1. Algorithm 1 - Classical RSA:

This is the traditional algorithm[23] to be followed by everyone in the world for their web applications to provide security to maintain a secret communication between two parties. Here is the example that shows the working principle of this algorithm.

Example for RSA algorithm (numerical values represented in Hex) :

Select prime numbers p= 2CF and q= 3A1

Calculate n by multiplying selected prime numbers: n= A312F

Calculate r by multiplying diminished values of two primes by unit1 , r= 9F800

Select e value as a relative prime of r and gcd(e,r) has to be 1 , e= 1F007

Calculate d value by finding inverse of e modulo r , d=41177

Public key pair : [e,n] = [1F007, A312F]

Private key pair : [d,n]= [41177, A312F]

Encryption : (message) e mod n

(2306) 1F007 mod A312F

Cipher text obtained: 75884 8B056 37C95 7CAD1

Decryption : (Cipher)d mod n

(75884 8B056 37C95 7CAD1) 41177 mod A312F

Decrypted Message : 2306

3.2. ESRKGS:

This algorithm[7] uses four prime numbers for the purpose of key generation phase to increase the security of the previous algorithm. The security of RSA algorithm mainly depends specifically on integer factorization of prime numbers .If modulus n can be broken into factors easily, then with help of public key component ‘e’, the decryption key pair can be easily calculated and that breaks the cryptosystem. This algorithm concentrates to increase the key generation time by using four prime numbers so that it takes much time to break the system by an attacker. The working principle of this algorithm is explained with the help of following example.

Select four prime numbers : p= 4F ,q=65, r=6D and s=59

Set n: = multiply(p,q) ; n= 1F2B

m: = multiply(r,s) ; m= 25E5

N: = multiply (n,m) ; N= 49D1877

Compute

r(n) = (4F-1)*(65-1) ; r(n) = 1E78

r(m)= (6D-1)*(59-1) ;r(m)= 2520

r(N)= n*m = 46B2700

Select a random number ‘e1 and e2 ’ such that gcd(r(N), e1)=1 and gcd(r(N),e2)=1

Selected e1 =AC9 and e2 =24B

Compute E1 = e1e2 mod N ; E1=469AE43

Select a random number E, such that gcd(E, r(n)*e1) =1 ; E=FB924C2ACC225

Compute‘d’ with Inverse of ‘e’ modulus (r*e1) : d = 10B914E8E502AD

Public key pair: [E,n] = [469AE43, 49D1877]

Private key pair: [d,n]= [10B914E8E502AD, 49D1877]

Encryption: (message) e mod n

(3B) 1F007 mod A312F

Cipher text obtained: B43

Decryption: (Cipher)d mod n

(75884 8B056 37C95 7CAD1) 41177 mod A312F

Decrypted Message: 3B

IV .Proposed Algorithm

The motivation behind the proposed algorithm is to increase the performance of RSA with respect to security and efficiency. An algorithm can be treated as a secure cryptosystem if the key generation time is high and it is said to be faster algorithm if it takes comparatively less time for encryption and decryption. This section explains the working nature of proposed algorithm in detail. Encryption using the proposed algorithm undergoes three phases: Key generation phase, Encryption phase, and Decryption phase. The following sections explain working of each phase clearly.

4.1. Key generation Phase:

The proposed algorithm follows the base of classical RSA with some enhancements to make it stronger to with stand attacks. The key generation phase of proposed algorithm undergoes the following steps:

1. Generating a Random Number
2. Primality test for finding Mersenne prime number
3. Set selected two mersenne prime number for Mp,Mq
4. Calculate N and Euler's totient value.
5. Select public key component : Pk

Following the above steps will result us two pair of keys that we can use for encryption and decryption.

4.1.1. Generating a Random Number:

Generating a number sequence with randomness[1], plays a vital role in the field of cryptography to strengthen the objectives involved in sharing the digital information secretly. The cryptographic keys generated by using concept of randomness will end up the attackers with a challenge of unpredictability of information about keys.

As the keys used in sharing the digital information have to be unpredictable for the attacker, keys has to follow certain property of diversity in data generation. The measure of diversity can be referred as entropy value of the keys used in cryptographic algorithms. If the entropy value of data is high then we can say that the data follows a high randomness otherwise there is a high probability of predicting the randomly generated values.

There are many efficient techniques are available for suggesting a sequence of random numbers for the purpose of practical cryptography. The classification of random

numbers generation mainly depends on the collected source of entropy values. There are multiple random generators among which PRNGs is one type.

The security of PRNGs depends on the factors like correlation between the sequence, time period of generator, repetition of sequences length. There are three types of algorithms for PRNGs. They are Linear congruential generators (LCGs) ,Lagged Fibonacci generators (LFGs) and Messene Twister .Messene twister has a long period of sequence, equal distribution with high ordered distance, generation of number in less time with effective usage of memory. In our algorithm we use Crypt Messene Twister PRNGs for generating random numbers[6].

4.1.3. Primality test for finding Mersenne prime number:

As a strong generated Prime numbers plays a major role in improving the security of RSA there is a necessity to test the random number generated to be a prime. Even though the definition of a prime numbers seems to be easy for one to remember but some class of prime numbers behave in non trivial nature when there is a random generation of numbers. There are several algorithms were discovered to check the primality of a random generated numbers. These algorithms are also classified into two types: Test for composite number and Test for randomized algorithms. Examples of some algorithms to test the primality of a number are Fermat test, Miller-Rabin test, Euler test, Lucas Lehemer test etc., Here in our algorithm we use Lucas Lehemer test to check the primality test of randomly generated number[4].

4.2 .Encryption and Decryption:

The encryption and decryption of the proposed algorithm follows the same principle of traditional RSA but with slight modifications to decrease the encryption and decryption time. To make encryption more secure we use public key component a larger value as RSA with small public key component can be vulnerable to attacks[2]. And as we used large prime numbers with a random generator of Crypt Mersenne twister it is difficult to factorize modulus. And we used Chinese remainder theorem[9] for the decryption process to decrease the decryption process.

4.3. CMT-RSA Algorithm: /* Steps in CMT-RSA Algorithm

```
        /*Key generation */
Step 1: Call for procedure CM_Twister ()
                                    Return Mp
Step 2: Call for procedure CM_Twister ()
                                    Return Mq
Step 3:     Check for Randomness test and
          Primality test for Numbers
Step 4:     If (Mp && Mq )
                    Set Pi = Mp and Pj = Mq
            else
                    Repeat Step1 followed by Step 2
Step 5:     Calculate N= Pi * Pj
Step6: Calculate r= Mul((subtract(Pi,1)          ,subtract(Pj,1))
Step 7: Set Pk = 65537
Step 8 :  set Pr = Pk . ModulusInverse(r)
Step 9:  Print Publickey pair(Pk,n)
Step 10: Print PrivateKeypair(Pr,n)
```

/*Encryption*/
Step11: Read Message to encrypt
Step 12 : set Cipher= message.modPow(PK,n)
Step 13: Return Cipher

/*Decryption*/
Step14: Set decipher = CT (Cipher, p,q);
Step15: Return decipher

V Experimental Setup

For the experimental purpose we used Java Programming with its inbuilt library functions for mathematical calculations by importing mathematical library functions using Big Integer. The implementation is done in such a way that prime numbers will be generated automatically according to the size specified by the user. Random large prime numbers are generated and rest of the calculations were done accordingly with respect to the specified algorithm.

VI Performance Analysis

We have tested our algorithm by varying the bit sizes for input values and we compared our algorithm with the above mentioned pre existing algorithms. The sample values of outputs were depicted in the table 1 .We also calculated time for generating key value pairs , time taken for converting plain text to cipher text and time taken taken to convert cipher text to plain text. Comparison of key generation time, encryption time and decryption time of all the three algorithms is depicted in table 2 ,table 3 and table 4 respectively. The graphical representation is of comparision is shown in fig1,fig2 and fig3 respectively. From the results obtained we observe that key generation time for proposed algorithm is more than the previous algorithms and that increases the security of the CMT-RSA comparatively high. We also observed that it takes less time than ESRKGS with respect to encryption and decryption time stating that less time for converting text from plaintext to cipher text and vice versa.

Size	Mp	Mq	n	Pk	Pr	Message	Encryption	Decryption
10	17	31	527	65537	353	420	114	420
20	797	773	616081	65537	464065	413342	39562	413342
30	31517	29581	932304377	65537	168434513	508184083	891659988	508184083
40	678329	750353	5089862001 37	65537	1.194E+11	1.2681E+11	4.28812E+11	1.26809E+11
50	30495379	19306643	5.88763E+14	65537	7.06297E+13	1.6163E+14	1.26943E+14	1.61629E+14
60	931802089	789814241	7.35951E+17	65537	3.46357E+17	3.4636E+17	4.65073E+17	3.46357E+17
70	1.9752E+10	3.0235E+10	5.97196E+20	65537	2.8877E+19	4.8372E+20	4.40256E+19	4.83719E+20

80	9.564E+11	6.4736E+11	6.19138E+23	65537	5.41908E+23	3.7613E+23	7.04427E+22	3.76134E+23
90	3.2946E+13	3.2826E+13	1.08148E+27	65537	8.13849E+25	8.1385E+25	1.88433E+26	8.13849E+25
100	9.4921E+14	1.063E+15	1.00904E+30	65537	8.43093E+29	3.5491E+29	6.42214E+29	3.54913E+29

Table 1 : Sample output values obtained by CMT-RSA

Key Generation Time(ms)			
Size (in bits)	RSA	ESRKGS	CMTRSA
100	72	113	282
128	92	165	403
256	133	237	619
512	352	389	657
1024	889	1168	1267
2048	4315	11164	11367
4096	91,542	181811	22300

Table 2 : Comparison of keygeneration time of CMT-RSA with existing algorithms

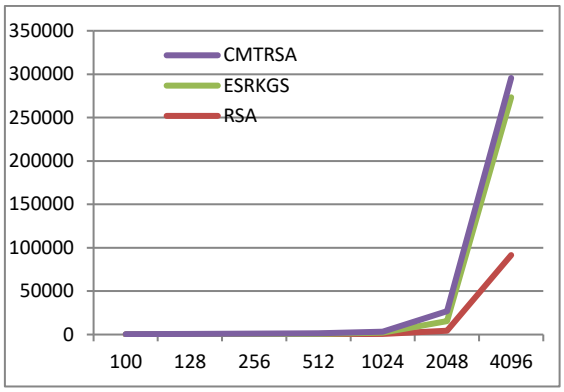

Fig 1: Graphical Representation of Key generation time

Encrytpion(ms)			
Size (in bits)	RSA	ESRKGS	CMTRSA
100	2	1.5	1
128	2.5	2	1
256	4	3	2
512	21	16	12
1024	170	105	80
2048	1393	784	690
4096	10,907	6620	5560

Table 3 : Comparison of encryption time of CMT-RSA with existing algorithms

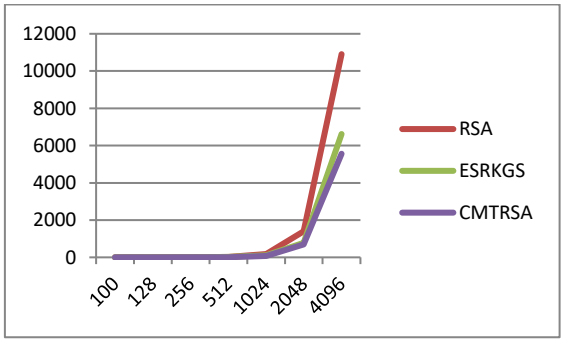

Fig2 :Graphical representation of Encryption time

DecryptionTime(ms)			
Size (in bits)	RSA	ESRKGS	CMTRSA
100	1	1.3	2
128	1.1	2	2
256	1.1	2	2
512	3	16	14
1024	22	106	30
2048	169	745	111
4096	1,381	6647	1038

Table 4 : Comparison of decryption time of CMT-RSA with existing algorithms

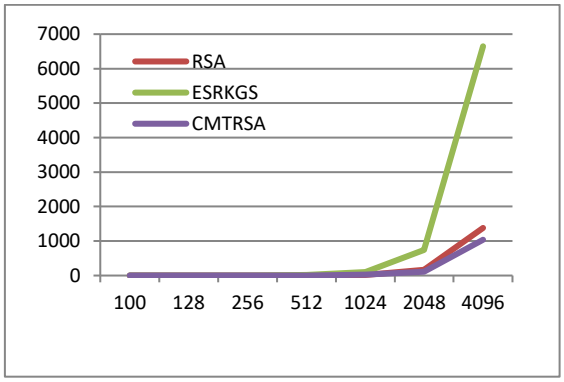

Fig 3 : Graphical representation of Decryption time

VII .Conclusion

The proposed algorithm mainly concentrates on generation and randomness of the large prime numbers used in the RSA public key cryptosystem. Due the usage of a security based random number generator we conclude that our algorithm builds the security in a promising way with respect to an increase in the key generation time which is slightly have an increasing factor when compared to previous two algorithms. As RSA with small public key component are vulnerable to attacks we used large public key component and to make the algorithm faster, we used Chinese remainder theorem concept to decrease the decryption time. The results obtained give us an idea about decrease in decryption time.

References:

[1] Fast generation of random, strong rsa primes , Robert D. Silverman ,RSA Laboratories May 17, 1997

[2] A study of public key ‘e’ in RSA algorithm, C Intila, B Gerardo, R Medina ,IOP Conf. Series: Materials Science and Engineering 482 (2019) 012016 IOP Publishing ,doi:10.1088/1757-899X/482/1/012016

[3] Implementing the RSA Cryptosystem Achim Jung Technische Hochschule Darmstndt, Fb. 4, 6100 Darmstadt, West Germany .

[4] Notes on Primality Testing And Public Key Cryptography ,Part 1: Randomized Algorithms ,Miller{Rabin and Solovay{Strassen Tests
Jean Gallier and Jocelyn Quaintance ,Department of Computer and Information Science ,University of Pennsylvania ,Philadelphia, PA 19104, USA.February 27, 2019
[5] The RSA Public Key Cryptosystem ,William P. Wardlaw Mathematics Department, U. S. Naval Academy, Annapolis, MD, 21146.
[6] Statistical analysis of random number generators luigi accardi , DOI: 10.1142/9789814343763_000
[7] An Enhanced and Secured RSA Key Generation Scheme (ESRKGS) M. Thangavel*, P. Varalakshmi, Mukund Murrali, K. Nithya
Department of Information Technology, MIT Campus, Anna University, Chromepet, Chennai, 600044, Tamilnadu, India
[8] P. L'Ecuyer. Uniform random number generation. In S. G. Henderson and B. L. Nelson, editors, Simulation, Handbooks in Operations Research and Management Science, chapter Chapter 3, pages 55–81. Elsevier, Amsterdam, The Netherlands, 2006
[9] Wu CH, Hong JH, Wu CW. RSA cryptosystem design based on the Chinese remainder theorem. In: Design Automation Conference, Proceedings of the ASP-DAC, Yokohama; 2001.p. 391
[10] Ivy PU, Mandiwa P, Kumar M. A modified RSA cryptosystem based on 'n' prime numbers. Int J Eng Computer Sci 2012;1(2):63e6.
Jamgekar RS, Joshi GS. File encryption and decryption using secure RSA. Int J Emerg Sci Eng (IJESE) 2013;1(4):11e4.
[11] https://www.di-mgt.com.au/rsa_alg.html#RIVE78
[12] Published: March 1997,Batch RSA,Amos Fiat, Journal of Cryptology volume 10, pages75–88(1997)
[13] United States Patent (19) 11 Patent Number: 5,848,159 Collins et al. (45) Date of Patent: Dec. 8, 1998
[14] T. Takagi. "Fast RSA-type Cryptosystem Modulo p k q." In H. Krawczyk, ed., Proceedings of Crypto '98, vol. 1462 of LNCS, pp. 318–326. Springer-Verlag, 1998
[15] An Efficient Implementation of Multi-Prime RSA on DSP Processor Conference Paper · May 2003
DOI: 10.1109/ICASSP.2003.1202387 · Source: IEEE Xplore

[16] On the Improvement of Wiener Attack on RSA with Small Private Exponent Mu-En Wu, Chien-Ming Chen, Yue-Hsun Lin, and Hung-Min Sun, Hindawi Publishing Corporation e Scientific World Journal Volume 2014, Article ID 650537, 9 pages http://dx.doi.org/10.1155/2014/650537
[17] Modified RSA Encryption Algorithm (MREA),Ravi Shankar Dhakar, Amit Kumar Gupta, Prashant Sharma ,Publication: ACCT '12: Proceedings of the 2012 Second International Conference on Advanced Computing & Communication TechnologiesJanuary 2012 Pages 426–429https://doi.org/10.1109/ACCT.2012.74.
[18] New computation technique for encryption and decryption based on RSA and ElGamal cryptosystems, Zulkarnain Md Ali, Jassim Mohammed Ahmed, Selangor Darul Ehsan, Published 2013, Computer Science, Journal of theoretical and applied information technology.
[19] A comparison between RSA and ElGamal based untraceable blind signature schemes, Khairul Alam ; Kazi Rokibul Alam ; Omar Faruq ; Yasuhiko Morimoto, 2016 International Conference on Networking Systems and Security (NSysS), 7-9 Jan. 2016,DOI: 10.1109/NSysS.2016.7400705, IEEE, Dhaka, Bangladesh.

[20] Design and implementation of an improved RSA algorithm, Yunfei Li ; Qing Liu ; Tong Li, 2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT),17-18 April 2010,DOI: 10.1109/EDT.2010.5496553 ,IEEE.
[21] Improvement in RSA Cryptosystem ,Seema Verma ,Dr Deepak Garg ,Journal of advances in information technology, vol. 2, no. 3, august 2011
[22] C. K. Koc, "High-Speed RSA Implementation," RSA Publications, ver. 2 1994
[23] R. L. Rivest, A. Shamir and L. Adleman, "A Method of obtaining Digital Signatures and Public Key Cryptosystems," Comm. of ACM, vol. 21, no.2, pp. 120-126, Feb 1978.
[24] D. Boneh and H. Shacham, "Fast variants of RSA," CryptoBytes, vol.5, no.1, pp. 1-9, 2002