Scientific Research Publishing

Journal of Computer and Communications, 2018, 6, 78-90
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Modified and Secured RSA Public Key Cryptosystem Based on “n” Prime Numbers

Muhammad Ariful Islam[1], Md. Ashraful Islam[1], Nazrul Islam[1]*, Boishakhi Shabnam[2]

[1]Department of Information and Communication Technology (ICT), Mawlana Bhashani Science and Technology University, Tangail, Bangladesh
[2]Uttara University, Dhaka, Bangladesh
Email: arifulislam14.ict@gmail.com, ashrafulmahim@gmail.com, *nazrul.islam@ieee.org, nimme.math@gmail.com

How to cite this paper: Islam, M.A., Islam, Md.A., Islam, N. and Shabnam, B. (2018) A Modified and Secured RSA Public Key Cryptosystem Based on “n” Prime Numbers. *Journal of Computer and Communications*, **6**, 78-90.
https://doi.org/10.4236/jcc.2018.63006

Received: February 4, 2018
Accepted: March 20, 2018
Published: March 23, 2018

Abstract

Cryptography is the study that provides security service. It concerns with confidentiality, integrity, and authentication. Public key cryptography provides an enormous revolution in the field of the cryptosystem. It uses two different keys where keys are related in such a way that, the public key can use to encrypt the message and private key can be used to decrypt the message. This paper proposed an enhanced and modified approach of RSA cryptosystem based on “n” distinct prime number. This existence of “n” prime number increases the difficulty of the factoring of the variable “*N*” which increases the complexity of the algorithm. In this approach, two different public key and private key generated from the large factor of the variable “*N*” and perform a double encryption-decryption operation which affords more security. Experiment on a set of a random number provided that the key generation time, analysis of variable “*N*”, encryption and decryption will take a long time compared to traditional RSA. Thus, this approach is more efficient, highly secured and not easily breakable.

Keywords

RSA Algorithm, Encryption, Decryption, Cryptosystem, Security, Public Key, Private Key

1. Introduction

Security refers to the process of keeping information confidential by protecting it from unauthorized users. According to security goal, confidentiality means to keep a data secured. It must be hidden from unauthorized access. Integrity proposes prevention of modifications. Availability means data being available to

*Corresponding author in the paper.

authorized persons when needed the data. Confidentiality, integrity, and availability are three security goals [1]. Security goal can achieve using cryptography which is a widely used method around the world.

Cryptography is a Greek word which means secret writing. It ensures secure communication and prevents the public from reading private messages. The cryptographic system is usually characterized by three main dimensions. They are, the operations involved in the transformation of plain text to cipher text, the number of secret keys used and the method of processing the plaintext [2]. Based on that property, cryptography is classified into two broad classifications. One of them is symmetric key and another is asymmetric key. Symmetric key cryptosystem uses the same key for encrypting and decrypting a message. Asymmetric key cryptosystem uses two different keys, the Public key for encrypting the message and private key for decrypting the message.

The Public key cryptosystem is based on two related keys, one is a public key and another is a private key. The sender generates a public key and encrypts the message. Sender shares one key which is called a private key and receiver decrypts the message using that key. The encrypted message cannot be decrypted by unwanted people, who know the public key only. In public key cryptography, the private key is always link with mathematically to the public key. It is always possible to attack public key system to generate a private key. This attack can be opposed by creating problem to generate a private key from the public key. Some public key algorithms designed such that deriving the private key from public key need to factorize large number by the attacker. RSA is known for such type of algorithm [3]. It takes two large prime number "p" and "q" and multiplies them to get a big number "N". The idea behind RSA is some mathematical operation easier to do but the inverse is very difficult without any additional information.

This research presents a new modification of RSA algorithm which is Modified RSA (MRSA) based on "n" distinct prime numbers with double encryption and decryption process. The weakness of RSA algorithm is the use of two prime numbers, small encryption exponent and use the same key for encryption and signing. This schema is based on "n" distinct prime numbers instead of two prime numbers which give an opportunity to select a big encryption exponent from large product "N" to enhance the security. Numerous prime numbers and big encryption exponent increase the factoring time comparing to RSA algorithm. The double encryption and decryption process make the algorithm stronger than RSA.

The rest of the paper is structured as follow. In Section 2, the proposed scheme represents some related work. In Section 3, there is a discussion about research methodology. The RSA algorithm and proposed model for RSA modification is described in Section 4 and 5. Section 6 represents implementation and result. Conclusion and future work are discussed in Section 7.

2. Related Works

This section involves the work done by various researchers in the field of RSA cryptosystem. The discussion is based on the Modification of RSA algorithm through the recent past. M. Thangavel *et al.* [4] proposed a modified RSA key generation algorithm which uses four primes instead of two primes and thereby increasing the time needed to find these primes. While increasing the security, the key generation time of the proposed algorithm is higher than original RSA. According to [5], encryption and decryption are not only dependent on "N" but also other new factors computed. The encryption and decryption technique is very complex and several factors are introduced without clearly justified. H. M. Sun *et al.* [6] came up with an algorithm which is known as dual RSA having two aspects from RSA with decreasing the needs of the storage for keys. It has two applications, first one is known as blind signature and other is known as security. Segar's [7] introduced a new idea to determine the private key without using the factoring approach using Pell's equation. Pell's RSA increases the strength that taking the private key "d" above the Wiener's possible range [8].

A Modified RSA introduced in the paper [9] that usage of three prime numbers instead of two prime numbers. The encryption and decryption process is the same as original RSA algorithm. Paper [10] ensures higher security by reducing modulus and private exponent in modular exponentiation. The performance of the decryption and signature generation are improved by this approach. A modification in encryption and decryption which involves magic rectangle and randomness of ciphertext which is calculated from the plain text was proposed in paper [11]. The main drawback of this approach is the need of additional time for constructing a magic rectangle. Chhabra and *et al.* [12] introduced an approach that eliminated the requirement of transferring "N" making it difficult for hackers to derive at the prime numbers used. Though the system can be cracked with the value of "k_p" and "d" unknown. Security is ensured by using dual modulus based double encryption and decryption with the use of Jordan function in the paper [13]. A new concept of adding a third prime number in the paper [14] increases the divisible time of modulo "N". The speed of encryption and decryption is increased compared to traditional RSA. Kong *et al.* [15] came up with an attack on the Encinas-Masqu'e-Dios algorithm. This algorithm uses two prime number "s", "t" and "N" represents the multiplication of these two prime numbers. The algorithm is insecure when the "public key" e satisfies the conditions $e > s + t$ and $e > N^{1/4}$. Rui's paper [16] developed a method which surpasses the original RSA algorithm in term of encryption and decryption speed. This algorithm is known as k-RSA algorithm combined k^{th} power residue theory and RSA algorithm.

This research focuses on to modify RSA algorithm and enhance the security. The schema is based on "n" prime numbers instead of two prime numbers with double encryption and decryption process. Multiple prime numbers increase the factoring time to get the private key. The encryption and decryption process is considered a pair of a random number and their modular multiplicative inverse to increase the security of RSA to a great extent.

3. Research Methodology

A modify RSA algorithm is proposed using "n" distinct prime numbers. A pair of a random number and their modular multiplicative inverse is used to increase the security of the RSA algorithm. Key generation, encryption and decryption time of Modified RSA (MRSA) algorithm to break the system is significantly higher. An observation has been performed using JAVA programming language. Modified RSA (MRSA) is implemented using JAVA Big Integer library functions for simulation purpose [17]. It is possible for the user to enter a prime number or set the bit length of prime numbers to generate automatically using the secure random function. Big Integer library provides various functions such as modular arithmetic, GCD calculation, primality testing, prime generation, bit manipulation and some other operations. Various bit length prime numbers are generated by using the prime generator function of Java Big Integer library to calculate key generation, encryption and decryption time. The Modified RSA (MRSA) model performs a comparison for key generation, encryption and decryption time between RSA and Modified RSA (MRSA) based on these calculated times for specific bit length.

4. RSA Cryptosystem

RSA is a public key algorithm based on the assumption that some mathematical operation easier to do in one direction but the inverse is very difficult. The idea behind RSA is that it is easier to multiply but difficult to factor the large number. Multiplication can be computed in polynomial time whereas factoring time can grow exponentially proportional to the size of the number.

```
RSA KEY GENERATION
RSA_key_gen()
Input:
Two prime numbers $p$ and $q$
Output:
Public key exponent: {$e$, $N$}
Private Key exponent: {$d$, $N$}
Procedure:
                    $N \leftarrow p * q$
Compute Euler phi value of $N$
                    $\Phi(N) \leftarrow (p-1)*(q-1)$
Find a random variable e, satisfying $1 < e < \Phi(N)$ and $\gcd(e, \Phi(N)) = 1$
Compute a random number d, such that $d * e \equiv 1 \bmod \Phi(N)$
RSA ENCRYPTION AND DECRYPTION
```

The encryption is done with the help of public key exponent and the decryption is done with the help of private key exponent.

```
RSA_Encryption()
Input:
```

```
Plain text message, $M$ ( <$N$)
Public key exponent: {$e$, $N$}
Output:
Cipher text, $X$
Procedure:
        $X \leftarrow \left(M^{e} \bmod N\right)$
RSA_Decryption()
Input:
Ciphertext message, $X$
Private key exponent: {$d$, $N$}
Output:
Decrypted plain text, $Y$
Procedure:
        $Y \leftarrow \left(X^{d} \bmod N\right)$
```

5. Proposed Model

The proposed Modified RSA（MRSA）scheme focuses on mitigating the major issues of RSA system. Most cases the major issue is, it is breakable because of easily computation of keys based on "*N*". Since it is the only product of two prime numbers, "*N*" can be easily traceable [18]. Once the value of "*N*" is obtained, a hacker can find the keys and break the system. The major modifications which make the system efficient and secure are discussed in the below section.

MRSA KEY GENERATION

The proposed model involves "*n*" distinct prime numbers. In this paper, all calculation and performance analysis is performed using four large prime numbers. The public and private key exponent consists of three components. "*N*" is the product of four prime numbers "*w*", "*x*", "*y*", and "*z*". The public key exponent consists of three components (e, f, N) where "*e*" and "*f*" are randomly taken. It further adds more complexity along with the factoring of "*N*". Since only the value of "*N*" is kept as a public and private component, an attacker with the knowledge of "*N*" cannot determine the value of four prime numbers which are the basis for finding the value of "*N*" and subsequently "*e*" and "*f*". The private key exponent consists of three components (d, g, N). For security purpose, the bit length of all four-chosen prime is of the same length as in case of traditional RSA. The algorithm is presented below.

```
MRSA_key_gen()
Input:
Four prime numbers $w$, $x$, $y$, and $z$
Output:
Public key exponent: {$e$, $f$, $N$}
Private Key exponent: {$d$, $g$, $N$}
Procedure:
```

$$N \leftarrow w * x * y * z$$

Compute Euler phi value of N

$$\Phi(N) \leftarrow (w-1)*(x-1)*(y-1)*(z-1)$$

Find a random variable *e*, satisfying $1 < e < \Phi(N)$ and $\gcd(e, \Phi(N)) = 1$

Find another random variable *f*, satisfying $1 < f < \Phi(N)$ and $\gcd(f, \Phi(N)) = 1$

Compute a random number *d*, such that $d * e \equiv 1 \bmod \Phi(N)$

Compute another random number *g*, such that $f * g \equiv 1 \bmod \Phi(N)$

MRSA ENCRYPTION AND DECRYPTION

The encryption is done with the help of public key exponent and the decryption is done with the help of private key exponent. The encryption and decryption are not only related to "N" which consists of four large prime numbers making it difficult to factor but also four random components "*e*", "*f*", "*d*", "*g*" are involved to make it even more difficult to break the system.

MRSA_Encryption()

Input:

Plain text message, M (<N)

Public key exponent: {e, f, N}

Output:

Cipher text, X

Procedure:

$$X \leftarrow \left(M^e \bmod N\right)^f \bmod N$$

MRSA_Decryption()

Input:

Ciphertext message, X

Private key exponent: {d, g, N}

Output:

Decrypted plain text, Y

Procedure:

$$Y \leftarrow \left(X^g \bmod N\right)^d \bmod N$$

Figure 1 present flow chart representation of Modified RSA (MRSA) algorithm. Four distinct prime number is taken as input to calculate N and $\Phi(N)$. A Pair of the random number (e, f) is selected from the range $1 < e < \Phi(N)$ as the public key exponent. The modular multiplicative inverse of those random numbers (d, g) is calculated to use as the private key exponent. Encryption and decryption is done by using those private key and public key exponents.

Let us discuss an example problem using the proposed Modified RSA (MRSA) algorithm.

MRSA example

- Take four distinct prime numbers $w = 53, x = 41, y = 43, z = 47$.

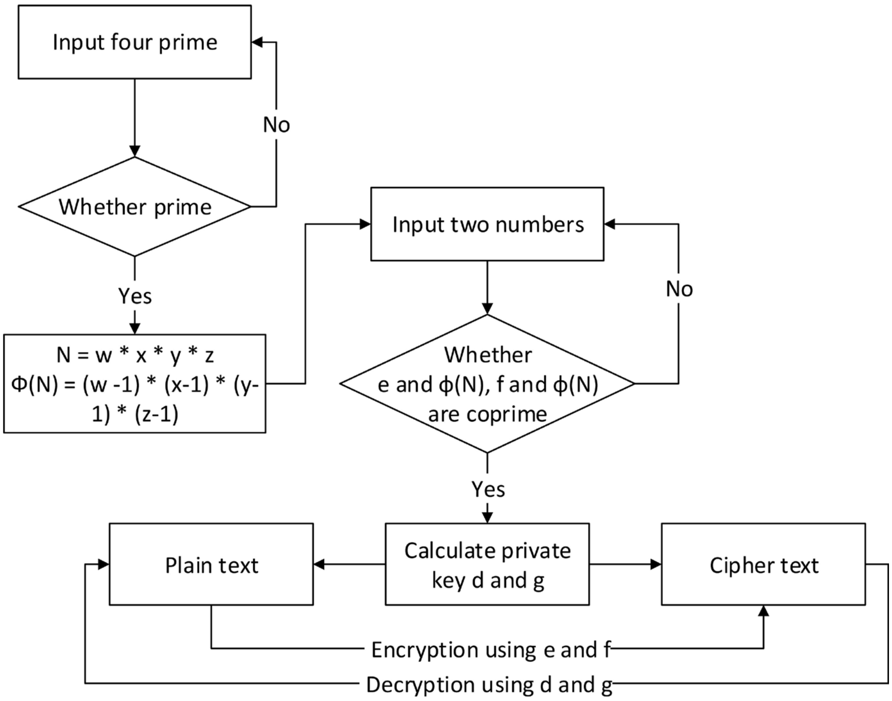

Figure 1. Flow chart of Modified RSA (MRSA) Algorithm.

- Compute $N = w*x*y*z$.
- $N = 53*41*43*47 = 4391633$.
- Compute Euler phi value of N

$$\Phi(N) = (w-1)*(x-1)*(y-1)*(z-1)$$

$$\Phi(N) = (53-1)*(41-1)*(43-1)*(47-1) = 4018560$$

- Find a random number *e*, satisfying $1 < e < \Phi(N)$ and $\gcd(e, \Phi(N)) = 1$.

$$e = 41$$

- Compute a random number *d*, such that, $d*e \equiv 1 \bmod \Phi(N)$.

$$d = 294041$$

- Find a random number *f*, satisfying $1 < f < \Phi(N)$ and $\gcd(f, \Phi(N)) = 1$.

$$f = 97$$

- Compute a random number *g*, such that, $g*f \equiv 1 \bmod \Phi(N)$.

$$g = 455713$$

- Input message, $M = 12321$
- Encryption, $X = (M^e \bmod N)^f \bmod N = 1081588$
- Decryption, $Y = (X^g \bmod N)^d \bmod N = 12321$

6. Implementation and Result

The algorithm is implemented in JAVA 8 running on an Intel (R) Core (TM) i5-3230M CPU @ 2.60 GHz machine and 4.00 GB RAM. The algorithm (RSA and MRSA) have different important parameter affecting its level of security and speed. Increasing the modulus length invoke complexity of decomposing it into factor. Thus, also increase the length of the private key and hence difficult to detect the key. The RSA and Modified RSA (MRSA) parameter changes depend on time and others remain fixed to study the relative emphasis.

6.1. Performance Analysis

The proposed Modified RSA (MRSA) algorithm is examined on varying bit sizes of input.

Performance of original RSA algorithm by Rivest, Shamir, and Adleman [3] are depicted in Table 1. Also the performance of Modified RSA (MRSA) scheme in terms of key generation, encryption time and decryption is shown in Table 2.

Comparing the above tables, it can be concluded that the time of key generation of Modified RSA (MRSA) is higher than that of RSA. The higher key generation time of Modified RSA (MRSA) can be seen as an advantage by the fact that the time to break the system is high because of the extra complexity added.

Figure 2 depicts the encryption time comparison between RSA and proposed Modified RSA (MRSA) scheme. It illustrates that, for the lower bit length of prime numbers, two algorithms consume the almost identical amount of time. But with the increase of bit length, the difference between curves rises rapidly.

Table 1. Performance of RSA.

Length of *w*, *x* (in bits)	Analyzing time for RSA algorithm		
	Key generation time (in ms)	Encryption time (in ms)	Decryption time (in ms)
100	76.63	0.16	0.25
128	90.46	0.17	0.28
256	94.96	0.35	0.96
512	177.47	0.56	5.2
1024	570.90	1.69	26.18
2048	4201.47	3.32	130.83
4096	54,368	11.17	1116.24

Table 2. Performance of Modified RSA (MRSA).

Length of *w*, *x*, and *z* (in bits)	Analyzing time for Modified RSA (MRSA) algorithm		
	Key generation time (in ms)	Encryption time (in ms)	Decryption time (in ms)
100	244	0.28	1.59
128	252.33	0.66	2.89
256	257.8	1.46	14.26
512	386.8	3.00	87.94
1024	1268.6	7.79	446.32
2048	7098.6	21.90	2472.70
4096	161,913	56.87	19,983.37

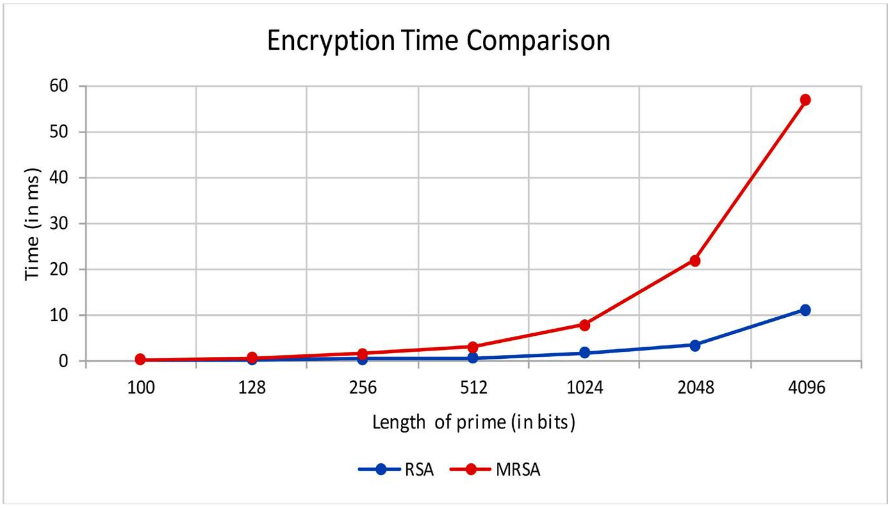

Figure 2. Encryption time comparison.

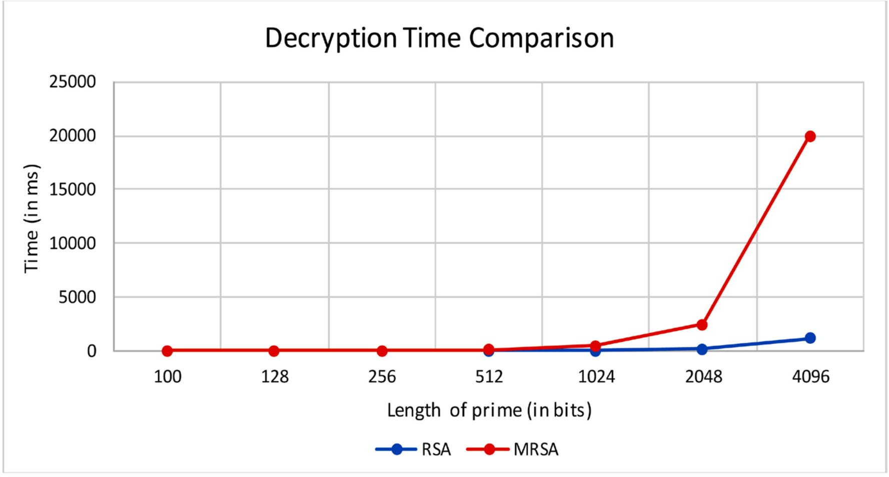

Figure 3. Decryption time comparison.

For example, input primes of bit length 2048, encryption time in Modified RSA (MRSA) is 21.9 ms and RSA takes 3.32 ms.

Figure 3 shows the decryption time comparison between RSA and proposed Modified RSA (MRSA) scheme. It demonstrates the almost identical amount of time consumed by RSA and Modified RSA (MRSA) for the lower bit length of prime numbers. With the increase of bit length, the difference between curves elevates rapidly. For example, input primes of bit length 4096, decryption time in Modified RSA (MRSA) is 19,983.37 ms and original RSA takes 1116.24 ms

From the above graphs, it can be easily seen that encryption and decryption times are higher than RSA. The increase in time is adaptable because it increases the security to a great extent in the proposed Modified RSA (MRSA) method.

6.2. Complexity Analysis

Comparison between complexity analysis of RSA algorithm and Modified RSA (MRSA) algorithm is discussing in below.

6.2.1. Complexity of RSA Algorithm

Complexity for Random selected two prime numbers:

- The complexity of MILLER-RABIN gives the above-mentioned complexities for finding a prime number is $O\left(s*(\log 2p)^2*\ln p\right)$ [19].
- Similarly, the complexity of the second number is $O\left(s*(\log 2q)^2*\ln q\right)$.

Complexity for calculation of *N*:

- The complexity of computation of *N* is $O(\log 2p*\log 2q)$.

Complexity of Computing Euler phi value of *N*

- By MODULAR-EXPONENTIATION, the complexity for the second part is (*N*) −1. The complexity of compute Euler phi value of *N* is [19]: $O\left((\log 2(p-1)*(q-1))^2*((p-1)*(q-1)-1)\right)$.

Complexity for random variables e:

- The complexity of finding the random variable *e* is $O\left(\log 2(p-1)*(q-1)+\gcd\left(e,(p-1)*(q-1)\right)\right)$, as it is known that *e* and (*N*) are coprime to each other so $\gcd\left(e,(p-1)*(q-1)\right)=1$, and so complexity is $O\left((\log 2(\log 2p-1)*(\log 2q-1))^2+1\right)$.

6.2.2. Complexity of Modified RSA (MRSA) Algorithm

The complexity will increase based on the number of the prime numbers conceded for the proposed algorithm.

Complexity for Random Selected prime numbers:

- Complexities for finding first prime number is $O\left(s*(\log 2w)^4*\ln w\right)$.
- Similarly, the complexity of the second number is $O\left(s*(\log 2x)^4*\ln x\right)$.
- The complexity of the third number is $O\left(s*(\log 2y)^4*\ln y\right)$.
- Similarly, the complexity of the fourth number is $O\left(s*(\log 2z)^4*\ln z\right)$.

Complexity for calculation of *N*:

- The complexity of computation of *n* is $O(\log 2w*\log 2x*\log 2y*\log 2z)$.

Complexity of Computing Euler phi value of *N*

- Complexity of compute Euler phi value of *N* is: $O\left((\log 2(w-1)*(x-1)*(y-1)*(z-1))^4*((w-1)*(x-1)*(y-1)*(z-1)-1)\right)$.

Complexity for random variables e and f:

- The complexity of finding the random variable *e* is $O\left(\log 2(w-1)*(x-1)*(y-1)*(z-1)+\gcd\left(e,(w-1)*(x-1)*(y-1)*(z-1)\right)\right)$, as it is known that *e* and (*N*) are coprime to each other so $\gcd\left(e,(w-1)*(x-1)*(y-1)*(z-1)\right)=1$, and so complexity is $O\left((\log 2(\log 2w-1)*(\log 2x-1)*(\log 2y-1)*(\log 2z-1))^4+1\right)$.
- Similarly, Complexity of finding the random variable *f* is $O\left((\log 2(\log 2w-1)*(\log 2x-1)*(\log 2y-1)*(\log 2z-1))^4+1\right)$.

Comparing the above complexity it depicts that Modified RSA (MRSA) is more complex than RSA algorithm. The complexity will increase depending on the number of primes considered for the algorithm.

6.3. Security Analysis

A wide variety of attacks are possible on RSA which includes brute force attack, timing attack etc. [20]. The time needed to break an RSA system is equivalent to the time needed for finding the prime numbers used.

This introduces the requirement of factoring the product "*N*". Elliptic Curve factorization Method (ECM) [18] and General Number Field Sieves (GNFS) [21] is used commonly for factoring "*N*". These are the fastest known factoring methods. Even though an attacker can factorize "*N*" by using those methods but it is still not sufficient enough in the computation of two arbitrary component "*e*" and "*f*". Above factorization technique can be used to find "*w*", "*x*", "*y*", "*z*" but "*e*" and "*f*" can only be found by an exhaustive brute force attack. In other words,

$$\Omega_{\text{system}} = \Omega_{w,x,y,z} + \Omega_{\text{brute force}}$$

Here,

Ω_{system} = Time needed to break the system

$\Omega_{w,x,y,z}$ = Time needed to find *w, x, y, z* using GNFS or ECM

$\Omega_{\text{brute force}}$ = Time needed for brute force attack for finding *e*, *f*

The important observation is, Modified RSA (MRSA) involves four primes "*w*", "*x*", "*y*", "*z*" and two random numbers "*e*", "*f*" for encryption whereas the original RSA involves only two prime numbers "*w*", "*x*" and only one random number "e" for encryption. So, the time needed to break Modified RSA (MRSA) algorithm will be greater than the time needed to break the original RSA at least by a factor of 2. And this will make the proposed Modified RSA (MRSA) algorithm more secure than the original RSA algorithm.

7. Conclusion and Future Work

The security of RSA depends on factoring the large number. This research works based on "n" distinct prime numbers instead of two prime numbers and it increases the attacking time to find the large prime number. The key generation of Modified RSA (MRSA) depends on large factor value "N" thus it needs higher key generation time. The higher the key generation time increases the time need to break the system which makes the system stronger. The double encryption and decryption procedure of Modified RSA (MRSA) is simple compared to the RSA algorithm thus it is not overhead on the system. Encryption and decryption also take more time than RSA algorithm. The accomplishment of the algorithm is measured with reference to time taken for brute force attack. Limitation of this proposed schema is it will not work properly unless "n" distinct prime numbers are considered. To enhance the security of RSA algorithm by adding some extra factors in encryption and decryption process can be a good future work.

References

[1] Forouzan, B.A. (2010) Cryptography and Network Security. Tata McGraw Hill Education Private Limited, New York.

[2] Stallings, W. (2013) Cryptography and Network Security: Principles and Practice. 6 Edition, Pearson, Boston.

[3] Rivest, R.L., Shamir, A. and Adleman, L. (1978) A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Communications of the ACM*, **21**, 120-126. https://doi.org/10.1145/359340.359342

[4] Thangavel, M., Varalakshmi, P., Murrali, M. and Nithya, K. (2015) An Enhanced and Secured RSA Key Generation Scheme (ESRKGS). *Journal of Information Security and Applications*, **20**, 3-10. https://doi.org/10.1016/j.jisa.2014.10.004

[5] Dhakar, R.S., Gupta, A.K. and Sharma, P. (2012) Modified RSA Encryption Algorithm (MREA). 2012 *Second International Conference on Advanced Computing Communication Technologies*, Rohtak, 7-8 January 2012, 426-429. https://doi.org/10.1109/ACCT.2012.74

[6] Sun, H.M., Wu, M.E., Ting, W.C. and Hinek, M.J. (2007) Dual RSA and Its Security Analysis. *IEEE Transactions on Information Theory*, **53**, 2922-2933. https://doi.org/10.1109/TIT.2007.901248

[7] Segar, T.C. and Vijayaragavan, R. (2013) Pell's RSA Key Generation and Its Security Analysis. 2013 *Fourth International Conference on Computing, Communications and Networking Technologies* (*ICCCNT*), Tiruchengode, 4-6 July 2013, 1-5. https://doi.org/10.1109/ICCCNT.2013.6726659

[8] Wiener, M.J. (1990) Cryptanalysis of Short RSA Secret Exponents. *IEEE Transactions on Information Theory*, **36**, 553-558. https://doi.org/10.1109/18.54902

[9] Al-Hamami, A.H. and Aldariseh, I.A. (2012) Enhanced Method for RSA Cryptosystem Algorithm. 2012 *International Conference on Advanced Computer Science Applications and Technologies* (*ACSAT*), Kuala Lumpur, 26-28 November 2012, 402-408. https://doi.org/10.1109/ACSAT.2012.102

[10] Li, Y., Liu, Q. and Li, T. (2010) Design and Implementation of an Improved RSA Algorithm. 2010 *International Conference on E-Health Networking Digital Ecosystems and Technologies* (*EDT*), Shenzhen, 17-18 April 2010, **1**, 390-393.

[11] Amalarethinam, I.G. and Leena, H.M. (2017) Enhanced RSA Algorithm with Varying Key Sizes for Data Security in Cloud. 2017 *World Congress on Computing and Communication Technologies* (*WCCCT*), Tiruchirappalli, 2-4 February 2017, 172-175. https://doi.org/10.1109/WCCCT.2016.50

[12] Chhabra, A. and Mathur, S. (2011) Modified RSA Algorithm: A Secure Approach. 2011 *International Conference on Computational Intelligence and Communication Networks*, Gwalior, 7-9 October 2011, 545-548. https://doi.org/10.1109/CICN.2011.117

[13] Swami, B., Singh, R. and Choudhary, S. (2016) Dual Modulus RSA Based on Jordan-totient Function. *Procedia Technology*, **24**, 1581-1586. https://doi.org/10.1016/j.protcy.2016.05.143

[14] Patidar, R. and Bhartiya, R. (2013) Modified RSA Cryptosystem Based on Offline Storage and Prime Number. 2013 *IEEE International Conference on Computational Intelligence and Computing Research*, Enathi, 26-28 December 2013, 1-6. https://doi.org/10.1109/ICCIC.2013.6724176

[15] Kong, F., Yu, J. and Wu, L. (2011) Security Analysis of an RSA Key Generation Algorithm with a Large Private Key. *ISC*, **2011**, 95-101.

[16] Rui, W., Ju, C. and Guangwen, D. (2011) A k-RSA Algorithm. 3*rd International Conference on Communication Software and Networks*, Xi'an, 27-29 May 2011, 21-24. https://doi.org/10.1109/ICCSN.2011.6013537

[17] Wagner, N.R. (2003) The Laws of Cryptography with Java Code.

[18] Ali, H. and Al-Salami, M. (2004) Timing Attack Prospect for RSA Cryptanalysis

using Genetic Algorithm Technique. *The International Arab Journal of Information Technology*, **1**, 80-85.

[19] Mohapatra, A.K., Gupta, N. and Prakash, N. (2016) Step-Wise Calculation of Performance and Complexity Analysis of Safer with RSA Algorithm. University School of Information Technology.

[20] Pomerance, C. (1996) A Tale of Two Sieves. *Notices of the American Mathematical Society*, **43**, 1473-1485.

[21] Lenstra, H.W. (1987) Factoring Integers with Elliptic Curves. *Annals of Mathematics*, **126**, 649-673. https://doi.org/10.2307/1971363