International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 06 | June 2021 www.irjet.net p-ISSN: 2395-0072

Secure Data Transmission using Upgraded Versions of RSA Algorithm

Bhoomika Wavhal[1], Samruddhi More[1], Aishwarya Chiluka[1], Santoshi Pote[2]

[1]Student, Electronics and Communication, Usha Mittal Institute of Technology, SNDT Women's University, Mumbai, Maharashtra, India
[2]Professor, Electronics and Communication, Usha Mittal Institute of Technology, SNDT Women's University, Mumbai, Maharashtra, India

---

Abstract - *Security and privacy are very important aspect of an application in any domain. These applications require data confidentiality, authenticity, integrity, and access control within the network. Security protocols and encryption prevents an attacker from tapping into the air and reading data as it passes by. Based on the security vulnerability of different applications, we aim to study the problems and issues that applications face in a real-time world. This paper discusses several variants of RSA algorithm which are best on its own and proposed a better and faster RSA based software for secure data transmission. The time required for encryption and decryption plays a very important role in this modern fast world, this paper aims at speeding up the performance of RSA algorithm by studying and executing its enhanced versions. The scope for improvement along with the efficiency of the existing new encryption algorithm and security protocols is also tested.*

Key Words: **Cryptography, Cryptography, Encryption, Decryption, RSA, Public-Key Cryptography, Modified RSA (M- RSA), Regulated RSA (R-RSA)**

1.INTRODUCTION

In this modern age, security is very important. With the advancement of technology, many threats have raised. It is important to have a proper security system. When private information is passed from one user to other there are chances of attacks to derive the information by a third party. In order to protect the information, one must convert this information into a secret text known as cipher-text, and the receiver must be able to convert cipher text into plain text i.e., the original information.

Secure Communication refers to the scenario where the message or data shared between two parties can't be accessed by an adversary. In Cryptography, an adversary is a malicious entity, which aims to retrieve precious information or data thereby undermining the principles of information security.

Cryptography is a technology used to achieve security for different applications. It basically consists of algorithms to convert plain text into cipher-text and vice versa. Cryptography typically consists of Encryption and Decryption algorithms. Encryption is used to convert plain text into cipher text whereas decryption is used to convert cipher text into plain text.

Cryptography can roughly be divided into Symmetric Key and Asymmetric Key cryptography. The symmetric key cryptography uses the same secret key for encryption and decryption whereas asymmetric key uses 2 different keys for encryption and decryption, respectively.

Classic cryptography solves only two tasks: the first task is key distribution where when we use symmetric encryption with both the sides having the common key, and the second task is even more important than the authentication which is creation of such implementations. [8]

The RSA algorithm is an asymmetric cryptography algorithm; this means that it uses a public key and a private key (i.e., two different, mathematically linked keys). RSA encryption is a type of public-key cryptography widely used for data encryption of e-mail and other digital transactions over the Internet.

This paper provides a study of different types of RSA algorithm and their comparison with conventional protocols based upon different parameters along with actual implementation.

1.1 RSA Algorithm

RSA-Key-Generation ():

Input: Two random distinct prime numbers p and q.
Output: Public Key (e), Private Key (d) and Modulus (n).
Begin
1) $n \leftarrow p*q$
2) Euler Totient Function $\Phi(n) \leftarrow (p-1) * (q-1)$
3) Public key e, such that, $\gcd(e, \Phi(n)) = 1$
4) Private key d, such that, $e * d \equiv 1 \bmod (\Phi(n))$
End

RSA-Encryption ():

Input: Plain text(T1), Public key(e).
Output: Cipher text (C1).
Begin
1) $C \equiv M^e \bmod n$
End

RSA-Decryption ():

Input: Cipher text(C1), Private key(d)

Output: Plain text (M).
Begin
1) $M \equiv C^d \bmod n$
End

2. LITERATURE SURVEY

RSA is a public-key cryptosystem that is widely used for secure data transmission. It is also one of the oldest which was first published in 1977. The acronym RSA comes from the surnames of its inventors - Ron Rivest, Adi Shamir, and Leonard Adleman. [6]

Out of many algorithms proposed for public cryptography, in RSA algorithm it is assumed that it is difficult to find the factor of large integers and hard to find the decryption key. But as decryption was based directly on variable x, it was easy to factorize and derive the key.

So, to improve the security, Enhanced RSA (ERSA) was proposed by Jayraj Gondaliya et a in [3] by considering the third prime number to increase complexity; but the original message can be acquired directly. Thus, direct attack may harm both RSA and ERSA. To overcome the problem a new algorithm with some new factors were computed to increase the complexity in encryption as well as decryption process. But use of many parameters overloaded the system.

So, to overcome the new problem another algorithm, Hybrid RSA (HRSA) [3] was developed by Jayraj Gondaliya et a which resulted in overcoming the limitations of above algorithm based on four prime numbers, which helped to reduce direct attacks as private key and public key are not dependent directly on component n, where n is the product of two prime numbers.

Yunfei Li et in [4] reviewed another new variant of RSA is also reviewed known as EAMRSA (Encrypt Assistant Multi-Prime RSA). This variant effectively combines Multi-Prime RSA [2] and [5]. It can obtain a higher speed than the basic RSA and the above two RSA variants. The variant also has obvious parallel characteristics and is easy to be implemented in parallel.

Dan Boneh and Hovav Shacham in [1] proposes a variant of RSA cryptosystem by reducing modules and private exponents in modular exponentiation. The experimental result shows that the speed of the decryption and signature has been substan- tially improved and the variant can be efficiently implemented in parallel. This paper also reviews some more variants of RSA like - CRT-RSA, Rebalanced RSA, Batch RSA, Multi Prime RSA, etc.

Multi-prime RSA technique by T.Collins, D. Hopkins, S. Langford, and M. Sabin (1997) [2] was introduced by Collins who modified the RSA algorithm so that it consists of k primes p1, p2, p3,.., pk instead of the traditional two primes p and q. Multi-Power RSA algorithm by T. Takagi (2009) [7] has generated a new variant cryptosystem by enhancing the speed of Multi Prime RSA decryption. [7]

3. PROBLEM STATEMENT

It is said that most deadly war that can happen on the internet will be caused by a hacker by attacking our personal, professional, and confidential data. Currently the standard is 2,048-bit RSA keys, up from 1,024, which was allowable until just a few years ago. Some organizations use, large bit keys, but as RSA key sizes grow, the amount of security provided by them is not commensurate to the amount of computational power that will be required to use them. This paper is to focuses on comparing different variants of RSA algorithm and conduct detailed analysis for proposing a more efficient variant of RSA Algorithm.

4. PROPOSED WORK

Cryptographic algorithms are intended to provide confidentiality and preserve the integrity of our data. But small size, limited computational capability, insufficient memory and power resources of the devices limits us to use them on a larger scale. In this paper, we have proposed two more variants RSA called **Modified RSA M-RSA** and **Regulated RSA (R-RSA).**

These algorithms perform encryption on various input bits and decrypt the cipher-text back to plain-text. M-RSA uses 6 prime numbers for key generation making algorithm more complex whereas R-RSA uses only 3 prime numbers for key generation and makes algorithm more complex than M-RSA.

4.1 Modified RSA Algorithm

Modified RSA or M-RSA algorithm calculation is planned so that there will be six indivisible numbers - p, q, r, s, t & u. Multiplication of combinations of these will be considered as n. Euler Totient Function will be determined by taking away 1 from each indivisible number and multiplying it with one another. The public key (e) comprises two numbers where greatest common divisor of the number should be 1. The private key is calculated such that $e*d = 1 \bmod (\Phi(n))$.

Modified RSA-Key-Generation ():

Input: Six random distinct prime numbers p, q, r, s, t & u.
Output: Public Key (e), Private Key (d).
Begin
1) $n \leftarrow p * q * r * s * t * u$
2) Euler Totient Function,
$\Phi(n) \leftarrow (p - 1) * (q - 1) * (r - 1) * (s- 1) * (t - 1) * (u - 1)$
3) Public key e, such that, $\gcd(e, \Phi(n)) = 1$
4) Private key d, such that, $e * d \equiv 1 \bmod (\Phi(n))$
End

Modified RSA-Encryption ():

Input: Plain text (M), Public key (e).
Output: Cipher text (C).
Begin
1) $C \leftarrow M^e \bmod n$
End

Modified RSA-Decryption ():

Input: Cipher text (C), Private key (d).
Output: Plain text (M).
Begin
1) $M \leftarrow C^d$ mon n
d mod n
End

4.2 Regulated RSA Algorithm

A regulated RSA or R-RSA algorithm is designed to accomplish high stability using lesser bit numbers. Input to this algorithm will be 3 large bit prime numbers - p, q & r and 3 small two to three-digit prime numbers - a, b & c. Multiplication of 3 large bit prime numbers will be store in n. A concept of the Regulated Euler Totient function is presented in this algorithm. Regulated Euler Totient function $\Phi_r(n)$ is alteration of Euler Totient function $\Phi(n)$. Euler Totient function is calculated by subtracting 1 from each prime number and then multiplying it together. Regulated Euler Totient function is calculated using the formula:

$$\Phi_r(n) = (p-1)^a * (q-1)^b * (r-1)^c$$

Regulated RSA-Key-Generation ():

Input: Three distinct large prime numbers p, q & r.
Three random distinct 3-digit prime numbers a, b & c.
Output: Public Key (e), Private Key (d).
Begin
1) $n \leftarrow p * q * r$
2) Calculate Regulated Euler Totient function,
$\Phi_r(n) \leftarrow (p-1)^a * (q-1)^b * (r-1)^c$
3) Generate a public key e, such that, gcd $(e, \Phi_r(n)) = 1$
4) Calculate the private key d, such that, $e * d \equiv 1 \bmod (\Phi_r(n))$
End

Regulated RSA-Encryption ():

Input: Plain text (M), Public key (e).
Output: Cipher text (C).
Begin
1) $C \leftarrow M^e \bmod n$
End

Regulated RSA-Decryption ():

Input: Cipher text (C), Private key (d).
Output: Plain text (M).
Begin
1) $M \leftarrow C^d \bmod n$
End

5. ANALYSIS

All the implementations are done on HP Pavilion Laptop, with Intel Core i5, 8.00 GB. The implementation is done using Python in SageMath 9.2 Jupyter Notebook. A comparative analysis of Traditional RSA [6], Enhanced RSA [3], Modified RSA, and Regulated RSA is presented based on encryption and decryption times for the data – "Usha Mittal Institute of Technology" for a key size of 80, 112, 128 and 144 bits, respectively.

Table -1: Total Time (in sec) taken by different RSA variants for Encryption and Decryption.

ANALYSIS TABLE				
KEY SIZE	**RSA**	**ERSA**	**MRSA**	**RRSA**
80	0.0214	0.0435	0.0188	0.2522
112	0.0232	0.0769	0.0391	0.3245
128	0.0327	0.1393	0.0409	0.3719
144	0.0488	0.2449	0.1101	0.4719

Few observations from the Table I:

- Computational time increases with increase in key size.
- Computational time for Enhanced RSA increases abruptly.
- compared to Modified RSA and Regulated RSA when key size is increased.
- Regulated RSA takes slightly more computational time than Modified RSA.
- Regulated RSA is more secured than other RSA algorithms.

6. CONCLUSIONS

This paper is about RSA algorithm and its application in secure data transmission. This paper also highlights how RSA is the most popular method for encryption and decryption along with the merits and demerits of different variants of RSA. This paper also discusses about the changes in efficiency resulted by modification of RSA algorithm where, Efficiency of an algorithm is a factor which is measured by three important parameters: key generation time, encryption time and decryption time. [3] In this paper we made a comprehensive survey on different RSA algorithms discussing about the advantages and disadvantages. This paper proposes two variants of RSA i.e., Modified RSA, and Regulated RSA to decrease the encryption and decryption time and speed up the process.

7. FUTURE APPROACH

In this paper, we have studied many variants of RSA like traditional RSA, Enhanced RSA, Hybrid RSA and have focused on improving the computational time and security by proposing two more variants of RSA. In future, we plan to better the performance of the proposed algorithms by changing their parameters to overcome all the drawbacks of a traditional algorithm without compromising on factors like CPU time, memory, and battery power etc. We also plan to increase the efficiency of these algorithm by noting down the varying values of the parameters during the process of encryption and decryption.

REFERENCES

[1] Dan Boneh and Hovav Shacham. "Fast variants of RSA". In: 5 (Aug. 2002) M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[2] Thomas Collins et al. Public key cryptographic apparatus and method. US Patent 5,848,159. Dec. 1998.

[3] Jayraj Gondaliya et al. "Hybrid Security RSA Algorithm in Application of Web Service". In: 2018 1st International Conference on Data Intelligence and Security (ICDIS). 2018, pp. 149–152. DOI: 10.1109/ICDIS.2018. 00032.

[4] Yunfei Li, Qing Liu, and Tong Li. "Design and implementation of an improved RSA algorithm". In: 2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT). Vol. 1. 2010, pp. 390–393. DOI: 10.1109/EDT.2010.5496553.

[5] Tsutomu Matsumoto, Koki Kato, and Hideki Imai. "Speeding Up Secret Computations with Insecure Auxiliary Devices". In: Advances in Cryptology — CRYPTO' 88. Ed. by Shafi Goldwasser. New York, NY: Springer New York, 1990, pp. 497–506. ISBN: 978-0-387-34799- 8.

[6] Ronald L Rivest, Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: Communications of the ACM 21.2 (1978), pp. 120–126.

[7] Tsuyoshi Takagi. "Fast RSA-type cryptosystem modulo pkq". In: Advances in Cryptology — CRYPTO '98. Ed. by Hugo Krawczyk. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 318–326.

[8] Vitaly Yakovyna et al. "The Performance Testing of RSA Algorithm Software Realization". In: 2007 9th International Conference - The Experience of Designing and Applications of CAD Systems in Microelectronics. 2007, pp. 390–392. DOI: 10.1109/CADSM.2007.4297593.