An Efficient Approach on the Implementation of RSA Algorithm in network computing environment

Reyna Royce, U.G Student, Department of Computer Science, St Joseph's University, Banglore, India, reynaroyce.12@gmail.com

Dr B.G Prasanthi, Assistant Professor, Department of Computer Science, St Joseph's University, Banglore, India, nitai2009@gmail.com

Abstract - The internet and other networking applications are gaining rapid growth with an ever-increasing need for strong information security because the internet by default is anything but secure. To protect the data transferred and to carry out safe transactions on the web, cryptographic techniques are used. These techniques play a vital role in keeping the internet as secure as possible. With constant research and developments in the field of encryption and cryptography, enhanced security is guaranteed every time. In the modern days, one of the most widely used cryptographic techniques is the RSA algorithm. This cryptographic algorithm traditionally uses 2 prime numbers for carrying out the mathematical concepts underlying the security of this technique. This paper presents a modified approach to the implementation of the same by using a combination of 4 prime numbers and also includes the OpenSSL implementation of the same with comparisons and analysis. By this alternate approach, the factoring complexity of the algorithm can be significantly increased thereby providing more security.

Keywords: Asymmetric encryption, Cryptography, Complexity, Data security, Factorization, Four prime numbers

I. INTRODUCTION

Today, the world is getting more connected day by day. We pass and receive all kinds of information and data to whomever we like and whenever we like without having to worry about what happens behind the scenes and how all of this is kept confidential. With attackers and eavesdroppers trying to get ahold of others' data, everything that happens on the web has to be concealed. That's where domains like cryptography, encryption, and cybersecurity come into the picture. Right from network security to database security, cryptography, and cryptographic algorithms play a vital role in providing data integrity. Though you may know it or not, these algorithms are the backbone of internet security and they guard us from the attackers and allows us to do anything we use the internet for. Information security and cryptography have been a hot topic since the internet existed. Researchers spent their time and resources finding new algorithms and improving existing ones to give the web the maximum security and to keep attackers away from it. Because today, the internet is a vital global resource that brings incredible benefits to everyone around the globe. And an internet without any sort of security and reliability takes away all of its gains.

In layman's terms, cryptography can be defined as the practice and study of a secure communication technique that enables only the sender and the intended recipient to view the contents of the data transmitted between them. While the message is being sent and when it is in transit, a variety of attacks and middlemen await it to take advantage of the data being sent. Cryptography is an art. The art of concealing information. It is where mathematics and computer science is at its finest. In earlier days, with restricted computational power and resources, the algorithms developed were of limited security. For that time, they were adequate and provided safety at their best for how they were designed and implemented. But with times changing and computing power advancing, more and more cryptographic algorithms were developed to guarantee enhanced protection. By tackling all the problems that can come their way, the world of cryptography forges ahead and continues to be the backbone of internet security and designs algorithms that can withstand the test of time.

II. TYPES OF ENCRYPTION ALGORITHMS

The encryption algorithms are mainly classified into 2 types namely symmetric encryption algorithms and

asymmetric encryption algorithms.

A) Symmetric Encryption Algorithms

Symmetric algorithms are the one that uses the same key for the process of encrypting data as well as decrypting the data i.e They are the family of cryptographic algorithms that uses identical keys for both of the processes. Both the sender and the receiver will have the same copies of the key which has

 DOI : 10.35291/2454-9150.2023.0113

to be kept confidential. Some of the widely used symmetric encryption algorithms are DES, AES, Twofish etc

B) Asymmetric Encryption Algorithms

The asymmetric cryptographic algorithms are the class of encryption algorithms that uses two separate but related keys for data encryption and decryption. The public key as the name implies is known to everyone and is used for encryption. The second key which is the private key, known only to the intended recipient is used for the decryption purpose. There is no doubt that asymmetric encryption algorithms provide way more security than symmetric ones because they use larger keys when compared to the latter. It removes the need for a key exchange like in symmetric algorithms. But asymmetric algorithms are resource-hogging techniques. They require bulky processing requirements and take too much computation time. Symmetric and Asymmetric encryption work hand in hand with the Public key infrastructure. It best makes use of the power of both encryption techniques. With the cons of symmetric techniques leveled up with the pros of asymmetric technique, the combination of both is so capable and powerful in PKI.

Some of the popularly known asymmetric techniques are RSA - Rivest, Shamir, Adleman, DSA - Digital signature algorithm, and Diffie-Hellman key exchange algorithm, etc.

III. RSA

RSA, which expands to Rivest Shamir and Adleman named after its founders, is one of the most widely used asymmetric cryptographic algorithms which uses two different keys that are mathematically linked together for encryption and decryption. It involves a public key as well as a private key. The public key is known to everyone and anybody can use it to encrypt the data but the private key will be known only to the one who has to decrypt the data. The algorithm has a huge practical application such as in website security, digital signatures - which are used for validating the authenticity of a message sent between two parties and other electronic transactions. RSA works by the method of prime factorization which is a very well-known mathematical concept to encrypt and decrypt the data. It uses two random prime numbers and works by factoring a large integer based on multiplication. Picking the prime numbers and getting their product might be an easy task, but finding out the numbers back from their product is what makes the process hard and the algorithm secure.

IV. PROBLEM DEFINITION

The security of the algorithm depends on the complexity of the two large prime numbers chosen. The algorithm is slow as it involves operations with large prime numbers and because of this fact, in real-world scenarios, only a small amount of data is encrypted with this technique. A solution to this problem is to implement the algorithm with more than 2 prime numbers. And as previously mentioned, RSA is an algorithm that requires intensive computing resources as it belongs to the asymmetric encryption category. But with multi-prime RSA, the largeness of the prime numbers chosen can be relatively decreased as more than 2 numbers are selected. With this, the speed of the algorithm can be enhanced and the increased need of system resources can be reduced thereby encrypting a good amount of data as compared to the original and traditional implementation of the RSA. In addition to this, various other optimized techniques can be used throughout the process for generating the prime and for carrying out modular arithmetic operations.

V. IMPLEMENTATION

While selecting numbers for multi-prime RSA, they can be chosen in such a way that they are not large to make the whole process system intensive as we are selecting more prime numbers to give it the complexity it needs to make the algorithm secure.

Algorithm:

Step 1 Select four prime numbers. Let it be p, q, r and s

Step 2: Calculate the product of the above two prime numbers and assign the value to N

Step 3: Calculating the Carmichael's totient function of N using the least common multiple. $\lambda(N) = \text{lcm}(p-1, q-1, r-1, s-1)$

Step 4: Calculate e value.

Step 5: Calculate the value of d

$$d = 1/e \bmod \lambda(N)$$

The public key pair is (n,e)

$$C = P^e \bmod N$$

The private key pair is (d, n)

$$P = C^d \bmod N$$

Example:

Step 1: Selecting four prime numbers. Let it be p, q, r and s

p = 2 q = 3 r = 5 s = 7

Step 2: Calculate the product of the above two prime numbers and assign the value to N

$$N = p \times q \times r \times s$$

$$N = 2 \times 3 \times 5 \times 77 = 210$$

Step 3: Calculating the Carmichael's totient function of N using the least common multiple.

$$\lambda(N) = \text{lcm}(p-1, q-1, r-1, s-1)$$

From the above step N = 210

Substituting N, p and q

$$\lambda(210) = \text{lcm}(1, 2, 4, 6)$$

 DOI : 10.35291/2454-9150.2023.0113

$\lambda(210) = 12$

Step 4: Taking e as 11 which is a coprime to 210

$e = 11$

Step 5: Calculate the value of d

$d = 1/e \bmod \lambda(N)$

Let d = 31

The public key pair (n,e) is (210, 11), so the encrypted ciphertext will be

$C = P^e \bmod N$

$= P^{11} \bmod 210$

The private key pair is (d, n) which is (31, 210). So, the decrypted text will be

$P = C^d \bmod N$

$= C^{31} \bmod 210$

Let p = 42, then encrypted text will be

$C = P^{11} \bmod 210$

$= 42^{11} \bmod 210$

$= 168$

For decryption,

$P = C^d \bmod N$

$= 168^{31} \bmod 210$

$= 42$

For the security of any encryption algorithm, the only thing that needs to be taken care of is its correct implementation and using a key that checks all of the security parameters. The security of the RSA algorithm depends upon the complexity of factoring the large prime numbers involved. It is generally considered that RSA is secure if N is large enough and when quantum computing is kept out of the picture. With the field of quantum computing showing immense potential, a few years from now, quantum computers can solve problems that are labeled as 'difficult' or 'impossible' today. As far as RSA is concerned, Shor's algorithm can run efficiently on such computers which is a quantum computing algorithm for finding prime factors of large integers and that is a security threat. Maybe with the advance in computing power, the algorithm may not be as resource-hogging as it is today. But with the usage of this modified approach to the algorithm with some optimized techniques, we can reduce its resource intensiveness which can then offer enhanced speed along with all of its security advantages.

VI. IMPLEMENTATION OF THE RSA ALGORITHM USING OPENSSL

Below images depicts the implementation of the RSA algorithm with 4 prime numbers using openssl in ubuntu 20.0

Step 1: Generation of the private key using 4 prime numbers and storing it as a .pem file

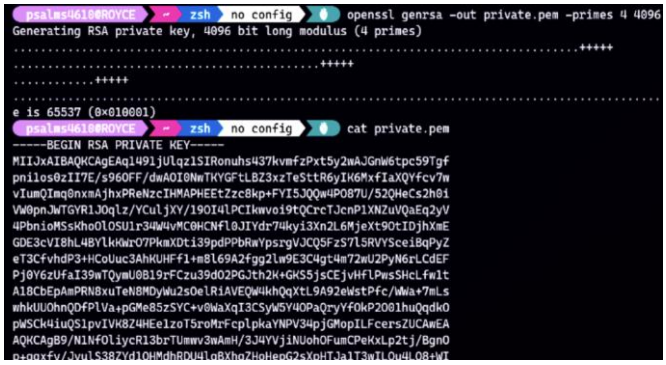

Figure 1: RSA Private key generation with 4 primes

Step 2: For 4 prime numbers the key size can be given as 4096 for maintaining the security level.

Step 3: Generate the public key using the generated private key and storing it as public.pem

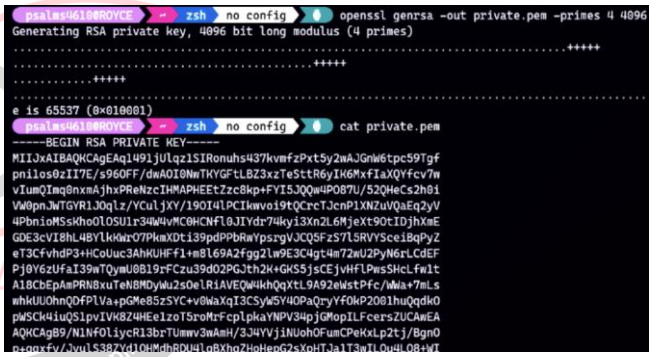

Figure 2: RSA Public key generation with 4 primes

Step 4: Create a sample file with the message that is to be encrypted.

Step 5: Encrypt the message that is stored in the sample.txt file using the public key and storing it as encrypted.txt. The ciphertext can be viewed by opening the file encrypted.txt which then shows the following.

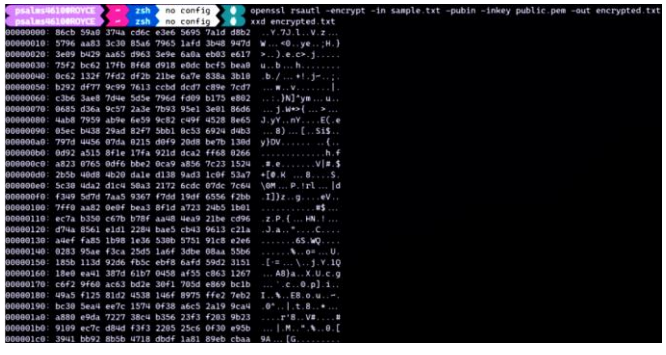

Figure 3: Encrypted text

 DOI : 10.35291/2454-9150.2023.0113

Step 6: Decrypt the encrypted file using the private key and store it as decrypted.pem. You can view the original message again after decryption by opening the file.

VII. COMPARISON AND ANALYSIS

The following table compares the various time measurements taken by the traditional approach as well as the modified approach of the RSA Algorithm. Although it is to be noted that the measurements may vary depending upon the OS, the systems' architecture and various other factors such as the memory and speed. The following measurements and tests were carried out in Ubuntu 20.0 with ARM64 system architecture.

Time Calculation for RSA with 2 prime numbers

Size (Bits)	Key generation time (In ms)	Encryption time (In ms)	Decryption time (In ms)	Total time for encryption and decryption (In ms)
256	3.82	0.16	0.73	0.89
512	14.13	0.94	15.31	16.25
1024	282.1	22.52	39.68	62.20
2048	3559.8	45.11	224.63	269.74
4096	56,231	64.56	1213.34	1277.90

TABLE 1

Time Calculation for RSA with 4 prime numbers

Size (Bits)	Key generation time (In ms)	Encryption time (In ms)	Decryption time (In ms)	Total time for encryption and decryption (In ms)
256	10.22	0.85	16.10	16.95
512	32.93	2.60	36.72	39.32
1024	624.37	46.12	582.46	628.58
2048	6801	61.31	1892.62	1953.93
4096	1,64,952	93.61	20,937	21030.61

TABLE 2

From the above tables, it can be concluded that the key generation time and the decryption time for the RSA with 4 prime numbers is larger when compared to the traditional RSA and the time taken for the key generation increases as the size increases. Higher the key generation time, higher the time required to break the key and hence more security. The modified approach takes more time for the process of decryption and can be considered more resilient to various brute force and factorization attacks.

It's also worth noting that the encryption and decryption time is also greater with the increased key generation time and the entire process can be system intensive.. This can be refined by optimizing some techniques throughout the entire implementation of the algorithm starting right from the prime generation to hardware acceleration for the algorithm implementation such as the use of cryptographic coprocessor or hardware acceleration modules which are dedicated hardware components that are optimized for cryptographic operations and can reduce the computational overhead. Also there are some efficient methods which are used for the generation of the prime numbers such as the elliptic curve method and prime number sieve which can reduce the time required for the number generation. Another factor that can be considered for reducing the overload is the usage of efficient modular arithmetic operations such as an optimized Chinese remainder theorem which can lead to easier computation and lesser complexity.

VIII. CONCLUSION AND FUTURE ENHANCEMENTS

The RSA algorithm which is one of the widely used cryptographic techniques with its strong security properties play an important role in ensuring integrity of data transmitted over networks. But in today's world where there is a need for constant improvement and modification for these techniques, optimizing the encryption algorithms is of great significance. This paper tries to present a modified approach for the traditional RSA algorithm by using 4 prime numbers instead of 2 and makes an attempt to prove the security of the improved version compared to the former. With the experimental results, it can be seen that the key generation time taken for the modified approach is greater compared to the RSA with 2 primes which means that more time will be required for breaking the system thus ensuring enhanced security.

However, the security of a system is subjective to its requirements and there is a need to balance security with the system's performance. If the benefits of the improved security is worth the implementation that can come with the increased computational overhead in the absence of any process optimizations as earlier mentioned, the modified approach this paper presents proves to be efficient as far as security is concerned. Further optimizing the algorithm by evaluating its performance and including some more concrete measures concerning efficiency can be a good future enhancement.

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.

 DOI : 10.35291/2454-9150.2023.0113

[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] Bell, E. T. "The Prince of Amateurs: Fermat.", New York: Simon and Schuster, pp. 56-72, 1986. Gabriel Vasile Iana1, Petre Anghelescu 1, Gheorghe Serban "RSAencryption algorithm implemented on FPGA" 1University of Pitesti, Department of Electronics and Computers, Romania, Arges, Pitesti, Str. Targul din Vale, No. 1, Code: 110040.

[5] Na Qi Jing Pan Qun Ding "The implementation of FPGA-based RSA public-key algorithm and its application in mobile-phone SMS encryption system" HeiLongjiang University Electronic Engineering Key Laboratory of Universities in Heilongjiang Province Harbin, China.

[6] Sonal Sharma, Prashant Sharma and Ravi Shankar Dhakar "RSA Algorithm Using Modified Subset Sum Cryptosystem" International Conference on Computer & Communication Technology (ICCCT)- 2011

[7] João Carlos Leandro da Silva, "Factoring Semiprimes and Possible Implications", IEEE in Israel, 26th Convention, pp. 182-183, Nov. 2010.

[8] Sattar J Aboud, "An efficient method for attacking RSA schemes", IEEE 2009.

[9] R. P. Brent, "An improved Monte Carlo factorization algorithm", BIT 20 (1980), 176-184. MR 82a:10007, Zbl 439.65001. Rpb051.

[10] 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.

 DOI : 10.35291/2454-9150.2023.0113