Secure Data Encryption Through Combination of RSA Cryptography Random Key Algorithm and Quadratic Congruential Generator

N Khairina¹, M K Harahap², A M Husein³, Muhathir¹, R Muliono¹

¹ Department of Informatics Engineering, Universitas Medan Area, Medan, Indonesia
² Department of Informatics Engineering, Politeknik Ganesha, Medan, Indonesia
³ Department of Informatics Engineering, Universitas Prima Indonesia, Medan, Indonesia

E-mails: nurulkhairina27@gmail.com, choir.harahap@yahoo.com, amirmahmud@unprimdn.ac.id, muhathirbangdes@gmail.com, rizkimuliono@gmail.com

Abstract. A cryptographic algorithm is used to enhance the security of the data. Algorithms whose security level is not too strong will pose a risk to data security. To safeguard the security of data, the key on the cryptographic algorithm is made as strong as possible, in order to avoid from irresponsibility parties. This paper uses the random keys for the Rivest Shamir Adleman Algorithm. The random keys that used to encrypt each plaintext letter are always different. The random key was originally formed from the first value of iteration, and it was called as seed. The seed for forming the value of p and q which generated by the Quadratic Congruential Generator Algorithm. The combination of the RSA Random Key Algorithm and the Quadratic Congruential Generator algorithm shows that each plaintext letter has own unique value of p, q, public key, and private key. This value is really always different for each encryption process. The result from this research can measure from the resistance of the cryptanalyst attack. The resistance of the RSA Random Key Algorithm toward the cryptanalyst attacks is 64 %.

1. Introduction

Technological advances that continue to evolve make data security an important priority. Research in cryptography continues to be explored by researchers. Cryptography is one of the arts [1] used in protecting data messages or secrets to be safe from irresponsible parties [2] [3]. Rivest Shamir Adleman (RSA) algorithm is the most popular modern cryptographic algorithm and has a fairly good level of security. RSA was introduced by Ron Rivest, Adi Shamir, and Len Adleman in 1970 [4]. The uniqueness of RSA lies in using two different keys, the encryption key is opened to the public (public key) and the decryption key is kept secret from the public (private key) [5] [6].

Previous research on RSA that combined with The Sieve of Eratosthenes was carried out by Dicky et al [7], this algorithm is able to produce prime numbers on a large scale and have better security due to the absence of the same RSA key as the data sent before.

Research conducted by Budi et al [8], the private key on the RSA algorithm can be solved by factoring the public key. The Kraitchik method is one algorithm capable of solving the RSA private key algorithm. Shikha et al [9] modifying the RSA algorithm using four prime numbers and multiple public keys with the k-nearest neighbor algorithm. This algorithm is not easily to attacked by cryptanalysts because two public keys sent separately.
Aarushi et al [10] combines the RSA algorithm with the Chinese Remainder Theorem algorithm and uses four primes and two public keys. The results showed an increase in speed in the decryption process so that it could overcome some attacks from cryptanalysts.

Nur Atiqah et al [11] modify the RSA algorithm and generate the Multi-Prime RSA (MPRSA) algorithm. Comparing the RSA algorithm to the MPRSA algorithm in time complexity, it can be seen that the MPRSA algorithm requires a shorter time than the RSA algorithm.

Kohei et al [12] perform cryptographic key generation by combining two types of binary sequences, namely M-Sequence and Quadratic Congruential Sequence so that a random number is generated that is quite good.

In 1954, Linear Congruential Generator (LCG) random number generator algorithm [13] was introduced by Lehmer [14] and then continued to develop into Quadratic Congruential Generator (QCG) algorithm and Cubic Congruential Generator (CCG) algorithm. Dicky et al [15] perform a Linear Congruential Generator (LCG) and Quadratic Congruential Generator (QCG) random key generation comparisons in One Time Pad (OTP) cryptographic algorithms. In this study, the combination of OTP and LCG algorithms has a faster processing time than the combination of OTP and QCG algorithms.

In this research, the researchers will modify the generating keys of RSA by forming a random key for each p and q. The process for generating this keys using Quadratic Congruential Generator (QCG) algorithm. So that, for each plaintext letter that will be encrypted will have its own pairing p, q, public key, and private key. In this research, the QCG algorithm will be combined with a prime number testing algorithm that using Fermat's method. Generate the random keys aim to strengthen the security of the RSA algorithm. In addition, the researchers also compared the complexity of encryption result between the original RSA algorithm and the RSA random key algorithm.

2. Literature Review

2.1 Prime Numbers
Prime numbers are a unique number. A prime number is an integer number greater than 1 and does not have a dividing factor other than number 1 and its own number [16]. In terms of determining the priority of a number, it is necessary to test the priority. Many formulas have been formulated in determining the priority of a number. In this discussion, we use Fermat's Theory as stated in theorem's little fermat theory.

2.2 Little Fermat Theorem
Fermat algorithm is used to test a prime number. A “n” number is a prime when:
\[ a^{n-1} \equiv 1 \mod n \] (1)

where : “ a " is a random number that 1 < a < n-1 [17]

“n” still as a prime probability. To get accuracy needs to be tested many times by changing the value of a, in this case, the test is done 10 times so that it can be concluded that n is a prime number. Numbers that do not meet the requirements of the formula above are a composite number.

2.3 Rivest Shamir Adleman
Rivest Shamir Adleman (RSA) algorithm is part of the asymmetric key cryptography algorithm, where the keys used in the process of encryption and decryption are different keys. RSA is also called the public key cryptography [18].
Here are the important things in the RSA algorithm:
1. p and q is a prime number,
2. n is the multiplication of p and q,
3. \( \phi (n) \) is the multiplication of (p-1) and (q-1),
4. e is the encryption key,
5. d is the decryption key,
6. $m$ is the plaintext, and
7. $c$ is the ciphertext

From the several important terms above, there are several variables which are confidential and should be kept, so that hackers cannot easily be known. The confidential variables are:
1. value of $p$ and $q$,
2. $\Phi(n)$,
3. decryption key ($d$) and
4. plaintext ($m$) [8]

2.3.1 Generate Key
The process of generating the key:
- a. Generate two different random numbers $p$ and $q$
- b. Test two numbers $p$ and $q$ with whether the numbers are primes or composite numbers.
- c. Calculate $n = p \times q$, make sure $p$ is not equal to $q$
- d. Calculate $\Phi(n) = (p-1) \times (q-1)$
- e. For the public key, select the value of $e$ which is relatively primed to $\Phi(n)$ where $1 < e < \Phi(n)$ and $gcd(e, \Phi(n)) = 1$
- f. For the private key, $d$ obtained by equation $d \equiv 1 \pmod{\Phi(n)}$, where $(0 \leq d \leq n)$
- g. We obtain the public key of the couple from $e$ and $n$
- h. We obtain the private key of the couple from $d$ and $n$

2.3.2 Encryption:
The process of Encryption:
- a. Take the public key $e$ and $n$
- b. Convert each plaintext letter to ASCII Code
- c. Perform the encryption process with $c_1 = m^e \mod n$, so obtained the first ciphertext
- d. To encrypt the second plaintext letter, do another key generation process by generating random numbers $p$ and $q$, and repeat this step until all the plaintext letters are encrypted.

2.3.3 Decryption:
The process of Decryption:
- a. Take the private key $d$ and $n$
- b. Convert each ciphertext letter to ASCII Code
- c. Perform the encryption process with $m_1 = c_1^d \mod n$ so obtained the first plaintext
- d. Perform the next decryption process using the private key that has been generated in the previous encryption process

2.4 Quadratic Congruential Generator
This is the formula of Quadratic Congruential Generator [15]:

$$X_n = (a \times X_{n-1}^2 + b \times X_{n-1} + c) \mod m \quad (2)$$

Description of formula:
- $X_n$ = Random number to $n$ of the series
- $X_{n-1}$ = Previous random number
- $a, b$ = Multiplier
- $c$ = Increment
- $m$ = Modulus
3. Proposed Method

In this study, the researchers will combine the RSA algorithm with Quadratic Congruential Generator (QCG) algorithm. The combination of the two algorithms is carried out in the process of forming RSA keys which depend on random numbers which will be tested for their prime. Random numbers will be generated with the QCG algorithm, where each letter on the plaintext will be a seed ($Z_0$). Then random numbers will be tested for their prime with the Theorem Little Fermat algorithm. In this modified RSA algorithm, p, q, the public key will be generated, different private keys for each plaintext letter that will be encrypted.

Here is the process of RSA encryption using RSA random key, for example, we take “icosnikom” as plaintext.

a. Generate random keys of prime numbers:

Let $a = 3; b = 17; c = 19$, and $m = 256$ for generate the random number.

Base on formula:

$$X_n = (a X_{n-1}^2 + b X_{n-1} + c) \mod m$$

Take a value $X_{n-1}$ as seed or the first value from iteration. To get a random number, the seed will be taken from the ASCII code plaintext that will be encrypted. For example we have “I” = 73,

$$X_n = (3 \cdot 73^2 + 17 \cdot 73 + 19) \mod 256$$

$$X_n = 95$$

We test the value of $X_n$, if it's a composite number, then repeat the steps above. If the number is prime, then the value of $X_n$ will be a value of $p$.

b. Look for the $q$ value with the same method above, and look at the table below. We have $a = 3, b = 17, c = 19$ and $m = 256$

| Plaintext | ASCII | P   | q   | Public Key (e, n) | Private Key (d, n) |
|-----------|-------|-----|-----|------------------|-------------------|
| i         | 105   | 197 | 11  | 3, 2167          | 1307, 2167        |
| c         | 99    | 167 | 251 | 3, 41917         | 27667, 41917      |
| o         | 111   | 227 | 137 | 3, 31099         | 20491, 31099      |
| s         | 115   | 109 | 139 | 5, 15151         | 2981, 15151       |
| n         | 110   | 149 | 193 | 5, 28757         | 22733, 28757      |
| i         | 105   | 197 | 11  | 3, 2167          | 1307, 2167        |
| k         | 107   | 89  | 157 | 5, 13973         | 8237, 13973       |
| o         | 111   | 227 | 137 | 3, 31099         | 20491, 31099      |
| m         | 109   | 139 | 227 | 5, 31553         | 18713, 31553      |

4. Result and Discussion

After we get the public key and private key, we can see the results of the encryption and their complexity in the table below:
Table 2. The Result of the Encryption Process:

| Plaintext | Ciphertext |
|-----------|------------|
| i         | ć          |
| c         | @          |
| o         | )          |
| s         | Å          |
| n         |          |
| i         | i          |
| k         | U          |
| o         | )          |
| m         | ®          |

After going through simple testing of ciphertext, the result shows that the resistance of the RSA Random Key for attack by cryptanalyst is 64% (fairly good) and testing of the brute force attacks can take a long time to solve the ciphertext.

![Strength of Ciphertext](image)

**Figure 1. Strength of Ciphertext**

![Brute Force Attack Cracking Time Estimate](image)

**Figure 2. Brute Force Attack Cracking Time Estimate**

5. Conclusion

The conclusion for this research are:

a. With a combination of the RSA random key algorithm and QCG algorithm, we can see that the uniqueness of this RSA algorithm has different p, q, public key, and private keys for each plaintext letter.

b. The resistance of the RSA Random Key Algorithm towards the cryptanalyst attacks is 64%.

The suggestion for this research in the future is the need for additional visualization for the distribution of random numbers that will be used in public keys and private keys.

References

[1] N. C. Ashioba and R. Yoro, “RSA Cryptosystem using Object-Oriented Modeling Technique,” *International Journal of Information and Communication Technology Research*, vol. 4, no. 2, pp. 57-61, 2014.

[2] Aryanti and I. Mekongga, “Implementation of Rivest Shamir Adleman Algorithm (RSA) and Vigenere Cipher In Web Based Information System,” *ICENIST*, vol. 31, no. 10007, pp. 1-5, 2017.
[3] S. Y. Tarigan, D. S. Ginting, M. L. Gaol and K. L. Sitompul, “The Combination of RSA and Block Chiper Algorithms to Maintain Message Authentication,” Journal of Physics: Conf. Series, pp. 1-7, 2017.

[4] M. A. Albahar, O. Olawumi, and P. T. Keijo Haataja, “Novel Hybrid Encryption Algorithm Based on AES, RSA, and Twofish for Bluetooth Encryption,” Journal of Information Security, pp. 1-10, 2018.

[5] K. Zhu, Z. Lin and Y. Ding, “A New RSA Image Encryption Algorithm Based on Singular Value Decomposition,” International Journal of Pattern Recognition and Artificial Intelligence, pp. 1-16, 2018.

[6] M. Kumar, “Advanced RSA Cryptographic Algorithm for improving Data Security,” Proceedings of the 50th Golden Jubilee Annual Convention CSI-2015 on “Digital Life”, Dec, 2015, pp. 1-7, 2015.

[7] D. Apdillah, M. K. Harahap, and N. Khairina, “Generating Mersenne Prime Number Using Rabin Miller Primality Probability Test to Get Big Prime Number in RSA Cryptography,” vol. 1, no. 1, 2017.

[8] B. S. Muchlis, M. A. Budiman and D. Rachmawati, “Teknik Pemecahan Kunci Algoritma Rivest Shamir Adleman (RSA) dengan Metode Kraitchik,” vol. 2, no. 2, 2017.

[9] S. Mathur, V. Goar, D. Gupta, and M. Kuri, “ANALYSIS AND DESIGN OF ENHANCED RSA ALGORITHM TO IMPROVE THE SECURITY,” in IEEE International Conference on "Computational Intelligence and Communication Technology, 2017.

[10] A. Rai and S. Jain, “Encryption and Decryption through RSA Cryptosystem using Two Public Keys and Chinese Remainder Theorem,” vol. 170, no. 1, 2017.

[11] N. A. A. Sani and H. Kamarulhaili, “RSA Cryptography and Multi-Prime RSA Cryptography,” in Proceedings of the 24th National Symposium on Mathematical Sciences, Penang, Malaysia, 2017.

[12] T. M. S. A. Kohei Kawase, “Randomness of Binary Sequences with Long Period by Combining m-Sequence and Knuth’s Quadratic Congruential Sequence,” in IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Taiwan, 2018.

[13] Imamah, A. Djunaidy, A. Rachmad, and F. Damayanti, “Comparison of Password Generator between Coupled Linear Congruential Generator (CLCG) and Linear Congruential Generator (LCG),” IOP Conf. Series: Journal of Physics: Conf. Series, pp. 2-6, 2017.

[14] R. T. H. F. A. E. Tchahou Tchendjeu, “FPGA Implementation of Linear Congruential Generator Based on Block Reduction Technique,” vol. 27, no. 10, 2018.

[15] D. Apdilah, M. K. Harahap, A. M. H. Nurul Khairina and Mawaddah, “A Comparison of One Time Pad Random Key Generation using Linear Congruential Generator and Quadratic Congruential Generator,” in International Conference on Mechanical, Electronics, Computer, and Industrial Technology, Medan, 2017.

[16] N. A. Pande, “Improved Prime Generating Algorithms by Skipping Composite Divisors and Even Numbers (Other Than 2),” vol. 31, no. 2, 2015.

[17] M. K. Harahap, “Membangkitkan Bilangan Prima Mersenne di atas 512 Digit Menggunakan Kombinasi Eratosthenes dan Fermat Little Theorem Untuk Pendukung Kunci Publik RSA,” in Seminar Nasional Teknologi Informatika dan Komunikasi (SNASTIKOM), Medan, 2015.

[18] D. S. R., Y. P., S. G. Kripa N Bangera, “Multilayer Security Using RSA Cryptography and Dual Audio Steganography,” in IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT), India, 2017.