A hybrid cryptosystem approach for information security by using RC4 algorithm and LUC algorithm

D Rachmawati*, M A Budiman and D F Perangin-angin

Computer Science Department, Faculty of Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, Medan, Indonesia

*Corresponding author: dian.rachmawati@usu.ac.id

Abstract. Security is an essential aspect in the sending information. One of the data security methods is by using Cryptography. The methods used in this study, i.e., hybrid cryptosystem. The hybrid cryptosystem is one way of combining two different algorithms which are using a symmetric algorithm such as RC4 and asymmetric algorithm such as LUC. The RC4 algorithm will be used to encrypt or to decrypt files and LUC algorithm will be used to encrypt or to decrypt the key algorithm RC4. Tools that will apply to build the system are by using the Java programming language and Android Studio. Testing of the system utilizing encryption and decryption of files. The result of the system testing is the system will convert plain text into cipher text which in turn into the decimal using ASCII character code. In the algorithm LUC, the processing time required will be longer if the value of primes was a large number. Encryption and decryption time needed on both algorithms are proportional linearly with the length of the message. Based on the results of the research the average time needed to process the encryption with the RC4 algorithm and LUC are 287.06 ms and 74.86 ms whereas the average time for the process of decryption by using the RC4 algorithm and LUC was 53.43 ms and 94.26 ms.

1. Introduction

The security system is an important part that must be realized in conveying information. The development of communication tools today makes many people exchange information through the internet. Information without security makes information stolen by unauthorized people, So we need a security system for information. One way to secure data using cryptography.

Cryptography is a science and an art of securing data by converting the data into a password [1] [5]. Data encryption can use asymmetrical and symmetrical algorithms. The asymmetric algorithm is an algorithm that has two keys, a public key for encryption and a secret key for decrypting data. The advantages of this algorithm are safer than symmetrical algorithms. The weakness of this algorithm is slower than symmetrical algorithms.

The symmetrical algorithm is an algorithm that is key for data encryption and decryption using the same key. The advantage of this algorithm is that the speed of decryption encryption is faster than asymmetric algorithms. The disadvantage of this algorithm is that because the encryption and decryption keys are one, it is difficult to send keys to decrypt files without being known to others.
Many asymmetric and symmetrical algorithms are used to secure data. One of them is the LUC algorithm which is an asymmetric algorithm and the RC4 algorithm which is a symmetry algorithm.

The LUC algorithm is an asymmetric key algorithm designed in 1993 by Peter J. Smith and Michael J. J. Lennon at the University of Auckland. The LUC algorithm uses two primes number to generate public and secret keys. This LUC algorithm has many similarities with the RSA algorithm, but in the LUC algorithm, the function of the RSA function is replaced by the lucas function, whose function itself is created by F.E.A Lucas.

The RC4 algorithm is an algorithm created by Ron Rivest from the RSA laboratory in 1987. The RC4 algorithm itself is a stream cipher whose encryption and decryption processes are carried out one by one bit or bit per bit.

A hybrid cryptosystem is one way to combine two different algorithms. One uses symmetrical algorithms and one asymmetric algorithm. Hybrid cryptosystem can provide better security by operating in a faster time. So in the current study, I want to combine the LUC algorithm and the RC4 algorithm by using a hybrid cryptosystem so that the file is quickly encrypted and harder to steal by unauthorized people.

2. Methods
2.1 Hybrid Cryptosystem
Hybrid Cryptosystem is used to combine symmetrical and asymmetric algorithms so that they can cover each other's weaknesses between the two algorithms. This method combines the concept of symmetric and asymmetric, using random key data encrypted with a symmetric method. The next stage of the random key used for the data encryption process is also encrypted, called the cipher key, but uses an asymmetric method using the public key of the recipient. The cipher text generated by each method is combined and sent to the recipient. The recipient will open the cipher key to getting a random key by using the private key it has to open the text cipher with a random key.

2.2 Algorithm RC4
The RC4 algorithm is an algorithm built by Ron Rivest at the RSA Security Laboratory. This RC4 algorithm is included in the stream cipher algorithm. Initially, the RC4 method was kept secret by RSA Security, but this was leaked on the internet in 1994 on the Cypherpunks mailing list. RSA Security has never officially released RC4. As a result, many call it ARC4 (alleged RC4 or RC4 suspect) to avoid trademark problems. RC4 contains two processes which are KSA and PRGA. [7]

Steps of RC4 Algorithm :
1. Key Scheduling Algorithm(KSA)
   First, initialize the first S-Box $S[0]$ to $S[255]$ which contains a number of 0 to 255 sequentially. After that another S-Box initialization, eg, array K with a length of 256. Fill the K array with the key repeated until the entire array K [0] to K [255] is filled. Then do the randomization step of S-Box with steps to make 2 additional variables such as i and j, then do the loop on i to 255. Look for the value of j by using the formula as follows :
   $$j = (j + S[i] + K[i]) \mod 256$$
   Then swap the values in the array $S[i]$ and $S[j]$. [4]

2. Pseudo Random Generation Algorithm(PRGA)
   Create a pseudo random byte by looking for values of i and j with the following formula:
   $$i = (i + 1) \mod 256$$
   $$j = (j + S[i]) \mod 256$$
   Swap values in arrays $S[i]$ and $S[j]$. Create a new variable to hold values such as t and k then look for the t and k values with the following formula:
   $$t = (S[i] + S[j]) \mod 256 \quad k = S[t]$$
2.3 Algoritma LUC

The LUC algorithm is an asymmetric algorithm designed by Peter J. Smith and Michael J. J. Lennon at the University of Auckland, Auckland, New Zealand. The algorithm is divided into 3 stages: key generation, encryption, and decryption. [9, 10]

1. Key Generator
   
   Select 2 primes (p and q). With the terms $\text{GCD}(p, q) = 1$ or p and q relatively prime.
   
   Find the value of n by multiplying p and q, with the value of n must be greater than the length of the encoding table.
   
   Calculate the Euler extension value (m):
   
   $$m = (p+1) \times (p-1) \times (q+1) \times (q-1)$$
   
   Take a random number that we symbolize with e whose value is greater than 1 than smaller than m ($1 < e < m$). e must be relatively prime with $\text{cd}(e, m) = 1$
   
   If you have obtained the value of e, then the value of e becomes a public key.
   
   Calculate the value of R (n):
   
   $$R(n) = \text{LCM} (p+1, p-1, q+1, q-1)$$
   
   Calculate d:
   
   $$d*e \mod R(n) = 1$$
   
   After getting the value of d, the value of d becomes a private key. After the key generator is finished processing, the encryption and decryption process is ready to be done.

2. Encryption
   
   Specify the plain text that you want to protect.
   
   Change the plain text to the ASCII encoding table.
   
   Calculate the cipher text:
   
   $$V[0] = 2$$
   
   $$V[1] = \text{plain text}$$
   
   $$V[i] = (\text{plain text} \times V[i-1] - V[i-2]) \mod n$$
   
   with the condition i is greater than 1 to V[e].
   
   $$V[e] = (\text{plain text} \times V[e-1] - V[e-2]) \mod n$$
   
   V[e] is a cipher text in the form of a number that will be changed back to the letter according to the ASCII table.

3. Decryption
   
   Specify the cipher text that you want to enter.
   
   Change the text cipher to the ASCII encoding table.
   
   Calculate plain text:
   
   $$V[0] = 2$$
   
   $$V[1] = \text{cipher text}$$
   
   $$V[i] = (\text{cipher text} \times V[i-1] - V[i-2]) \mod n$$
   
   provided that i is greater than 1 to V[d]
   
   $$V[d] = (\text{cipher text} \times V[d-1] - V[d-2]) \mod n$$
   
   V[d] is plain text in the form of a number that will be changed back to the letter according to the ASCII table.

3. Results and Discussions
   
   Based on the above method, the illustration of information security methods to be built is as follows on Figure 1:
Figure 1. The purposed hybrid cryptosystem

a. Key generation process
Choose 2 random primes which are p-value, q-value which have gcd 1.
\[ p = 179 \] and \[ q = 109 \]
After getting \( p \) and \( q \), look for the value \( n \).
\[ n = p \times q, n = 179 \times 109, n = 19511 \]
After getting \( n \), find the value \( m \).
\[ m = (p + 1) \times (p - 1) \times (q + 1) \times (q - 1) \]
\[ m = (179 + 1) \times (179 - 1) \times (109 + 1) \times (109 - 1) \]
\[ m = (180) \times (178) \times (110) \times (108) \]
\[ m = 380635200 \]
Then find the value \( e \).
\( (1 < e < m) \) and \( \text{gcd} (e, m) = 1 \) obtained \( e = 7 \).
Then look for the \( R(n) \) value
\[ R(n) = \text{LCM} (p + 1, p - 1, q + 1, q - 1) \]
\[ R(n) = \text{LCM} (180, 178, 110, 108) \]
\[ R(n) = 528660 \]
Finding \( d \) by \( d \times e \mod R(n) = 1 \) is obtained \( d = 75523 \).

b. C4 Encryption Algorithm
1. KSA Phase (Key Scheduling algorithm)
Make the initial \( S \) array and the \( k \) string to hold the key and repeat the key up to as many as 256.
After that, 256 \( S \) array arrays are randomized by formula
\[ j = (i + S[i] + K[i]) \mod 256 \]
swap \( S[i] \) and \( S[j] \)
Iteration 1 (\( i = 0 \); \( j = 0 \))
\[ j = (0 + S[0] + K[0]) \mod 256 \]
\[ j = (0 + 0 + 102) \mod 256 \]
\[ j = 102 \]
swap \( S[0] \) & \( S[102] \)
Iteration 2 (\( i = 1 \); \( j = 102 \))
\[ j = (102 + S[1] + K[1]) \mod 256 \]
\[ j = (102 + 102) \mod 256 \]
\[ j = 204 \]
swap \( S[1] \) & \( S[204] \)
Up to 256 iterations of the \( S \) array
2. PRGA (Pseudo-Random Generation Algorithm) Stage
   After doing this, then doing PRGA is done along the plain text that you want to encrypt. Plain text that you want to encrypt is "d."
   Iteration 1 ( i = 0 ; j = 0 )
   \[ i = ( i + 1 ) \mod 256 \]
   \[ i = 1 \]
   \[ i = ( j + S[0] ) \mod 256 \]
   \[ i = ( 0 + S[0] ) \mod 256 \]
   \[ i = 102 \]
   swap S[i] & S[j] swap S[1] & S[102]
   \[ t = ( S[i] + S[j] ) \mod 256 \]
   \[ t = ( 38 + 204 ) \mod 256 \]
   \[ t = 242 \]
   \[ k = S[242] \]
   \[ k = 25310 = 111111012 \]

3. XOR Process
   After getting the key of each character, xor is done to get the cipher text [6-7]. The following is the ASCII value for the character "d" = 0110010. The encryption process is shown on Table 1.

   Table 1. XOR for the encryption process
   \[
   \begin{array}{|c|c|c|}
   \hline
   \text{Plain text} & 01100100 \\
   \text{Key} & 11111101 \\
   \text{Cipher text} & 10011001 \\
   \hline
   \end{array}
   \]

   So the text cipher is obtained: "153"

4. Calculation of the Encryption of the LUC Algorithm
   Key encryption uses the public key LUC algorithm that has been generated when performing key generation. The key you want to encrypt is "f". And the public key is: e = 7 and n = 19511.
   Plain text = f = 10210
   Calculate the cipher text:
   \[ V[0] = 2 \]
   \[ V[1] = 102 \]
   \[ V[i] = ( \text{plain text} \times V[i-1] - V[i-2]) \mod n \text{ with the condition } i \text{ is greater than 1 to e} \]
   \[ V[2] = ( \text{plain text} \times V[i-1] - V[i-2]) \mod n \]
   \[ V[2] = (102 \times V[1] - V[0]) \mod 19511 \]
   \[ V[2] = 10402 \]
   Continued to V[8]
   \[ V[7] = ( \text{plain text} \times V[i-1] - V[i-2]) \mod n \]
   \[ V[7] = (102 \times V[7-1] - V[7-2]) \mod 19511 \]
   \[ V[7] = 6355 \]
   cipher key : “6355"
5. Description with the LUC algorithm
Key descriptions use the LUC private key algorithm that has been generated when performing key generation. The cipher key that you want to decrypt is "6355". And the private key is: \( d = 75523 \) and \( n = 19511 \).
Cipher key = 6355

Count Key RC4

\[
\begin{align*}
V[0] &= 2 \\
V[1] &= 6355 \\
V[i] &= (\text{cipher key} \times V[i-1] - V[i-2]) \text{ with the condition } i \text{ is greater than } 1 \text{ to } d. \\
V[2] &= (\text{cipher key} \times V[1] - V[0]) \mod n \\
V[2] &= (6355 \times 6355 - 2) \mod 19511 \\
V[2] &= 17764 \\
\end{align*}
\]
Repeated until:

\[
V[75523] = (\text{cipher key} \times V[i-1] - V[i-2]) \mod n \\
V[75523] = 102(f)
\]

6. Calculation of RC4 Decryption Algorithm
The process of KSA and the prgaPRGA process in the process of decoding the rc4 algorithm is the same at the time of the encryption process so that in the description process it will only discuss the xor process. Ciphertext is: "153". The encryption process is shown on Table 2.

| Table 2. Xor for decryption process |
|-------------------------------------|
| Ciphertext | 10011001 |
| Key        | 11111101 |
| Plain text | 01100100 |

After xor is done, the plain text will return to "d"
The time which takes to encrypt 16 characters plain text with RC4 algorithm by using a key length of 5, 10, 16 are as follows on Figure 2:

*Figure 2. Graph of plain text encryption time by using the RC4 algorithm*
When the LUC algorithm encryption process is calculated based on key lengths 5, 10, 16 and uses public key $e = 5$ and $n = 3901$ are as follows on Figure 3:

![Figure 3. Graph of key encryption time by using LUC algorithm](image)

When the decryption process of the LUC algorithm is calculated based on the key lengths 5, 10, 16 and using the private key $d = 190109$ and $n = 3901$ are as follows on Figure 4:

![Figure 4. Graph of the key decryption time by using LUC algorithm](image)

The RC4 algorithm decryption process is calculated based on the length of the cipher text and the RC4 key length that has been determined on Figure 5:
4. Conclusion
By using the concept of hybrid cryptosystem i.e. algorithm RC4 for securing messages and LUC algorithm to secure the key of RC4, the message becomes more secure than using a single algorithm. After the process of decryption, message can be returned to its original form. Cipher text and cipher key that has been encrypted will change to the form of decimal numbers. Based on experiments conducted on the RC4 algorithm and LUC algorithm, the time of encryption and decryption needed in both algorithms is linearly proportional, i.e. the longer the message and the key used, the time it takes longer. Based on the results of the research the average time needed to process the encryption with the RC4 algorithm and LUC are 287.06 ms and ms 74.86 whereas the average time for the process of decryption by using the RC4 algorithm and LUC was 53.43 ms and 94.26 ms.

Acknowledgments
We gratefully acknowledge that this research is supported by Fund Dissemination IPTEKS Research Results for Lecturers / Researchers Universitas Sumatera Utara.

References
[1] Mollin RA 2007 An Introduction to Cryptography London (New York: Chapman and Hall/CRC)
[2] Rachmawati D et al 2018 IOP Conf. Ser.: Mater. Sci. Eng. 300 012042
[3] Smart N 1999 Cryptography: An Introduction 3rd Ed (Bristol City: Mcgraw-Hill College)
[4] Zoltak B 2004 VMPC one-way function and stream cipher In International Workshop on Fast Software Encryption pp. 210-225
[5] Schneir B 1996 Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed (John Wiley & Sons)
[6] Padhi R and Pradhan P 2016 Int. Conf. Commun. Electron. Syst. (ICCES) Coimbatore pp. 1-5.
[7] Sahi A, Lai D, and Li Y 2018 3rd Int. Conf. Comput. Commun. Syst. (ICCCS) pp. 33-40. IEEE.
[8] Jindal P and Singh B 2014 Int. Conf. Recent Adv. Innov. Eng. (ICRAIE-2014) Jaipur pp. 1-5.
[9] Zeng T and Zhou Y 2014 ICINS 2014 Int. Conf. Inf. Netw. Secur. Beijing pp. 121-126.
[10] Napitupulu D, Nurdiyanto H, Kurniasih N, Hasibuan A, Daengs GA, Manurung RT, and Abdullah D 2018 IOP Conf. Ser 1114 012090