Implementation of hybrid cryptosystem using Rabin-p algorithm and One Time Pad to secure images

M A Budiman, M Zarlis, and Hafrizah

Department of Computer Science, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Jl. Alumni No. 9-A, Kampus USU, Medan 20155, Indonesia

mandrib@usu.ac.id, m.zarlis@usu.ac.id, and hafrizah088@gmail.com

Abstract. Cryptography is a study of mathematical techniques to secure messages by encoding the messages so it cannot be read by a third party. There are two types of algorithms in cryptography, symmetric algorithm and asymmetric algorithm. Asymmetric algorithms, such as Rabin-p cryptosystem, are known to be secure if the key sizes are long enough. However, asymmetric algorithms are ineffective to encrypt large data since the calculations consume CPU time and the resulting ciphertext is multiple times bigger than the original data. In this research, we combine Rabin-p cryptosystem and One Time Pad (OTP) in a hybrid scheme to solve this problem. The OTP is used to encrypt the data (images) and Rabin-p algorithm is used to encrypt the OTP key. In this research, we demonstrate how to secure a *.bmp file with this hybrid scheme.

1. Introduction

The issue of message security becomes an issue that if discussed will not be endless until now. Confidential and sensitive messages are so security that people of no interest cannot access them. In keeping messages secure we can use cryptography [1]. Cryptography is the science and art used in keeping messages safe by turning them into encoded messages so that they cannot understand their meaning [2]. Cryptographic algorithms are divided into three parts based on the keys it uses, namely symmetric algorithms, asymmetric algorithms and Hash functions [3]. Without us noticing, it turns out that cryptography already exists in our lives today, such as transactions on cash machines, e-commerce transactions to pay-TV channel transactions also using cryptography[ 1].

Rabin-p is an asymmetric algorithm named after the rabin cryptosystem and an additional p that symbolizes that the proposed scheme uses only one prime p number used as its decryption key. [4]. The keys used in the Rabin-p algorithm are two types of keys namely public keys for encryption and private keys for decryption. The advantage of Rabin-p algorithm is that it has a good level of security, but the disadvantage of Rabin-p algorithm lies in the resulting ciphertext longer than plaintext.

Rabin Algorithm is an asymmetric algorithm that is a variant of the Rivest Shamir Adleman (RSA) algorithm. Rabin's algorithm uses two keys in the encryption and decryption process: a public key and a private key [5]. The RSA algorithm was the first to use the asimtris concept of Diffie-Hellmen. RSA's security depends on strength of factoring a large integer into its prime factors [6].

One Time Pad (OTP) is a symmetrical algorithm and the One Time Pad algorithm also includes a perfect algorithm that cannot be solved so it is named Unbreakable Cipher [1]. The key that this algorithm uses is only one key. One key is only used for one message and for the next message is a randomize process. The key length should be equal to the length of the message.

Hybrid cryptosystem is a combination between symmetrical algorithms and asymmetric algorithms by using the advantages of each algorithm [7]. In this study the authors will combine the One Time Pad (OTP) and Rabin-p algorithms using hybrid cryptosystems.
2. Method
In this section the author will explain how the Rabin-p algorithm works and how the One Time Pad algorithm works. Plainimage encryption uses the One Time Pad algorithm and plainkey encryption using the Rabin-p algorithm. Then decrypt the cipherkey using the Rabin-p algorithm and decrypt the cipherimage using the One Time Pad algorithm.

2.1. Rabin-p
2.1.1. Key Generator. In this process will be generate public keys and private keys that will be used for encryption and decryption.
   a. Select two prime random numbers for the \( p \) and \( q \) values where \( 2^k < p, q < 2^{k+1} \) satisfy \( p \equiv q \equiv 3 \mod 4 \).
   b. Compute \( N \equiv p^2q \)
   c. \( N \) as public key and \( p \) as private key.

2.1.2. Encryption. In this process encrypt the keys that have been generate.
   a. Chose plaintext \( 0 < m < 2^{2^k-1} \) where \( \gcd(m, N) = 1 \).
   b. Compute \( C \equiv m^2 \mod N \)

2.1.3. Decryption. This decrypts the encryption key to be the original key.
   a. Compute \( w \equiv c \mod p \)
   b. Compute \( m_p \equiv w^{p^{-1}} \mod p \)
   c. Compute \( i \equiv \frac{c-m_p^2}{p} \mod p \)
   d. Compute \( j \equiv \frac{1}{2m_p} \mod p \)
   e. Compute \( m_i \equiv m_p + j \)
   f. If \( m_i < 2^{2^k-1} \) then \( m = m_i \)
   g. Else return \( m = p^2 - m_i \)

2.2. One Time Pad
2.2.1. Encryption. This process encrypts the image that has been.
   \[ C_i \equiv P_i + K_i \mod N \]

2.2.2. Decryption. In this process it will decrypt the encrypted image.
   \[ P_i \equiv C_i - K_i \mod N \]

3. Results and Discussion
In this section will be explained the process of image encryption and key encryption, as well as image decryption and key decryption. First generate the key to be mastered for the encryption and decryption process, then test for some imagery size, the test results are displayed in graphic form.

3.2. Rabin-p
3.2.1. Key Generator
   ● Specify security parameters \( k \). Suppose \( k = 5 \)
   ● Specifies the prime number \( p \) and \( q \) where \( 2^k < p, q < 2^{k+1} \)
     satisfy \( p \equiv q \equiv 3 \mod 4 \).
   \[ p=151 \text{ dan } q=59 \]
   \[ 2^k < p \]
   \[ 2^5 < 151 \]
   \[ 32 < 151 \]
   \[ 151 \equiv 3 \mod 4 \] (satisfy)
   ● Compute \( N \equiv p^2q \)
   \[ N = 151^2 \cdot 59 \]
\[ N = 22801.59 \]
\[ N = 1345259 \]

### 3.2.2. Encryption

- Chose plainkey \( m \) where \( 0 < m < 2^{2k-1} \) and \( GCD (m, N) = 1 \). Suppose \( m \) we take the value from one of the keys that is 121. \( 0 < m < 2^{9} \)
  \( 0 < 121 < 2^{9} \)
  \( 0 < 121 < 512 \)

\[ GCD(m, N) = 121 \mod 1345259 = 225 \]
\[ 1345259 \mod 121 = 19 \]
\[ 102 \mod 19 = 7 \]
\[ 19 \mod 7 = 5 \]
\[ 7 \mod 5 = 2 \]
\[ 5 \mod 2 = 1 \]
\[ 2 \mod 1 = 0 \]

Because it qualifies, then \( m=121 \) can be used.

- Compute \( c \equiv m^2 \mod N \).
  \( c \equiv 121^2 \mod 1345259 \).
  \( c \equiv 14641 \mod 1345259 \).
  Then \( c \equiv 14641 \)

### 3.2.3. Decryption

- Compute \( w \equiv c \mod p \)
  \( w \equiv 14641 \mod 151 \)
  \( w \equiv 145 \mod 103 \)

- Compute \( m_p \equiv w^{p+1} \mod p \)
  \( m_p \equiv 145^{151} \mod 151 \)
  \( m_p \equiv 121 \mod 151 \)

- Compute \( i \equiv \frac{c-m_p^2}{p} \)
  \( i \equiv \frac{14641-121^2}{151} \)
  \( i \equiv \frac{14641-14641}{151} \)
  \( i \equiv 0 \)

- Compute \( j \equiv \frac{i}{2m_p} \mod p \)
  \( j \equiv \frac{0}{2(121)} \mod 151 \)
  \( j \equiv 0 \mod 151 \)

- Compute \( m_1 = m_p + jp \)
  \( m_1 = 121 + 0(151) \)
  \( m_1 = 121 \)

- Specifies \( m \). If \( m_1 < 2^{2k-1} \) then \( m = m_1 \)
  \( 121 < 2^9 \)
  \( 255 < 512 \)
  Then came the results \( m = 121 \).

### 3.3. One Time Pad (OTP)

#### 3.3.1. Encryption
- Compute $C_i \equiv P_i + K_i \ (mod \ N)$
  $C \equiv 121 + 151 \ (mod \ 256)$
  $C \equiv 16 \ (mod \ 256)$

3.3.2. Decryption
- Compute $P_i \equiv C_i - K_i \ (mod \ N)$
  $P \equiv 16 - 151 \ (mod \ 256)$
  $P_i \equiv 121 \ (mod \ 256)$

In this study, we tested rabin-p algorithms and One Time Pad algorithms with multiple imagery sizes for their encryption and decryption processes. The test results can be seen in figure 1 and figure 2.

**Table 1.** Result of encryption process

| Plainimage | Cipherimage | Time of encryption process (second) |
|------------|-------------|-------------------------------------|
| Plainimage1.bmp | Cipherimage1.bmp | 9.7130 |
| 100 x 100 pixel | 100 x 100 pixel | 29.3 KB | 29.3 KB |
| Plainimage2.bmp | Cipherimage2.bmp | 12.0807 |
| 100 x 200 pixel | 200 x 220 pixel | 58.6 KB | 56.8 KB |
| Plainimage3.bmp | Cipherimage3.bmp | 28.0513 |
| 100 x 300 pixel | 300 x 178 pixel | 87.9 KB | 87.9 KB |
| Plainimage4.bmp | Cipherimage4.bmp | 50.4977 |
| 100 x 400 pixel | 276 x 400 pixel | 117 KB | 117 KB |
| Plainimage5.bmp | Cipherimage5.bmp | 79.1628 |
| 100 x 500 pixel | 100 x 500 pixel | 146 KB | 146 KB |
Figure 1. Test results in encryption process

Table 2. Result of decryption process

| Cipherimage | Plainimage | Time of encryption process (second) |
|-------------|------------|------------------------------------|
| Cipherimage1.bmp 100 x 100 piksel 29,3 KB | Plainimage1.bmp 100 x 100 piksel 29,3 KB | 28,7314 |
| Cipherimage2.bmp 100 x 200 piksel 58,6 KB | Plainimage2.bmp 100 x 200 piksel 58,6 KB | 66,6513 |
| Cipherimage3.bmp 100 x 300 piksel 87,9 KB | Plainimage3.bmp 100 x 300 piksel 87,9 KB | 106,4756 |
4. Conclusion
The conclusion of this study is that firstly, the One Time Pad (OTP) algorithm can be used to secure image messages, and the Rabin-p algorithm can be used to secure keys. Secondly, in the one time pad (OTP) key randomizing process the larger the size of the image message the greater the time used. Thirdly, the larger the size of the image used the longer the process takes in the encryption and decryption process, and the decryption process takes longer than the encryption process time. Fourthly, the size of the original image with cipherimage does not change when performing the encryption and decryption process.

References
[1] Munir R 2019 Kriptografi (Bandung: Informatika).
[2] Meyer C H and Matyas S M 1982 Cryptography: a new dimension in computer data security; a guide for the design and implementation of secure systems.
[3] Ariyus D 2018 Kriptografi Keamanan Data dan Komunikasi, (Yogyakarta: Penerbit Graha Ilmu).
[4] Asbullah M A and Ariffin M R K 2014 Rabin-p cryptosystem: Practical and efficient method for Rabin based encryption scheme. arXiv preprint arXiv:1411.4398.
[5] Budiman M A, Rachmawati D and Lydia S 2019 The implementation of the Pollard factorization method in the cryptanalysis of the Rabin public key algorithm *Journal of Physics: Conference Series* **1235** 012086.

[6] Budiman M A and Rachmawati D 2017 On factoring RSA modulus using random restart hill-climbing algorithm and Pollard’s rho algorithm *Journal of Physics: Conference Series* **943** 012057.

[7] Schneier B 1996 Applied Cryptography 2nd Ed (New York: John Wiley & Sons, Inc)