Comparison Of RSA Algorithm & Hybrid RSA-OTP Algorithm For Image Security

Dian Rachmawati*, Mohammad Andri Budiman*, and William Andrei*

1Departemen Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, Jl. Universitas No. 9-A, Medan 20155, Indonesia

*Email: dian.rachmawati@usu.ac.id, mandrib@usu.ac.id, william.andrei1010@gmail.com

Abstract. Transferring an image is one of the common processes. The image transferring, the sender wants to send a picture by maintaining the security and integrity and even the confidentiality of an image. One of the techniques to manage that aspect is cryptography – hybrid cryptosystem and Rivest-Shamir-Adleman algorithm. Hybrid cryptosystem uses RSA-OTP algorithm for its combination. Hybrid cryptosystem has the strength to secure an image because of its combination of the algorithm. A hybrid use symmetric algorithm to encrypt the image and asymmetric algorithm to encrypt the key from a symmetric algorithm. Rivest-Shamir-Adleman algorithm is an asymmetric algorithm which is hard to guess because of its different key. Rivest-Shamir Adleman Algorithm has a public key to encrypt and private key to decrypt; also prime test was implemented using the Agrawal-Kayal-Saxena algorithm. One-Time-Pad algorithm is a symmetric algorithm that is too hard to guess because this algorithm must have a key which length must be similar to plain's length, the more length of the plain, the harder key to guess.

1. Introduction
Cryptography is the science that using mathematics technique to transform the information to be secure and also immune to attack [10] — the reasons why we use cryptography, to guarantee the confidentiality, integrity, authentication, and non-repudiation. The data previously was converted to cipher before it was sent so that if the eavesdropper tries to steal the data, they would have only found abstract data that cannot be used [4]. The cryptographic algorithms to be used is the Rivest Shamir Adleman (RSA) which utilizes two prime numbers to generate a public key and private key [8]. The advantages of the RSA algorithm is located on its difficulties to factor a large number into a prime factor which is used to get a private key[2].

RSA is also useful to use in the hybrid cryptosystem. Hybrid cryptosystem scheme is used to improve the security of data. In the hybrid cryptosystem, a file is secured by using the symmetric algorithm and symmetric key is secured by using the asymmetric algorithm [7]. Hybrid makes prime factor more complicated to make the data more safety, and if the information is attempted to be broken by the eavesdropper, then the cipher and also the RSA will be broken[3].

One-time-pad (OTP) as a stream cipher where the plain and the key has an equal length, and the key never reused for any plain. This cipher is easily proven mathematically unbreakable, even in unlimited computer power access[1]. Third parties can decrypt cipher if only when they get the sequences of the key, and if they do not, it is impossible to decipher the ciphertext[5]. That makes OTP is suitable as the symmetric algorithm for the hybrid cryptosystem.
2. Method

In this comparison, there are three main menus: Key Generator to generate the public and private key, encryption/decryption for hybrid cryptosystem, and encryption/decryption for RSA Algorithm

2.1 Steps for generating the key:
   1. Randomize a large prime number p with Agrawal Biswas test prime Algorithm. The test as follows:
      a. Input n where \( n \in \mathbb{N} \)
      b. If \( n \) is a perfect power, if \( n = a^b \) for integers \( a > 1 \) and \( b > 1 \), \( n \) is not a prime
      c. Find the smallest \( r \) such that \( \text{ord}_r(n) > \log_2 n \)
      d. If \( 1 < \gcd(a, n) < n \) for some \( a \leq r \), \( n \) is not a prime
      e. For \( a = 1 \) to \( \lfloor \sqrt{\phi(r)} \log_2(n) \rfloor \)
      f. If \((x + a)n \equiv xn + a \pmod{2r - 1, n}, \( n \) is not a prime
      g. \( n \) is prime, output \( n \).
   2. Randomize value p and q from the large prime number
   3. Get the value of \( n \) as the public key from \( n = p \times q \)
   4. Get the value of \( \phi(n) \) from \( \phi(n) = (p-1)(q-1) \)
   5. Get the value of \( e \) as a public key from large prime number with condition \( 1 < e < \phi(n), \gcd(e, \phi(n)) = 1 \)
   6. Get the value of \( d \) from \( d = e^{-1} \pmod{\phi(n)} \)

Then on the Hybrid Encryption menu, at first image converted to Base-64 String then the result is encrypted by using the OTP algorithm.

2.2 Steps for encrypting the plaintext:
   1. Convert an image to Base-64 String as a plaintext
   2. Randomize character for the key of OTP algorithm as plainkey using secure random which similar length to plaintext length
   3. Encryption the plaintext using the One-Time-Pad algorithm, which formula is \( c_i = (p_i + k_i) \pmod{256} \)
   4. Save the result value of ciphertext.

2.3 Steps for encrypting the plainkey:
   1. Get the public key (e,n).
   2. Encryption the plainkey using RSA algorithm, which formula is \( c = m^e \pmod{n} \)
   3. Save the result value of cipher key.

Then on the Hybrid Decryption menu, get the result of the encrypted file. Then, the cipher key is decrypted using the RSA algorithm and the ciphertext is decrypted using the OTP algorithm to get the original image.

2.4 Steps for decrypting the cipher key:
   1. Get the private key (d).
   2. Decryption the cipher key using RSA algorithm, which formula is \( m = c^d \pmod{n} \)
   3. Save the result value of decryption.

2.5 Steps for decrypting the ciphertext:
   1. Get the plainkey as the result of cipher key decryption.
   2. Decrypt the ciphertext using OTP algorithm, which formula is \( p_i = (c_i - k_i) \pmod{256} \)
   3. Convert the result into an image.

On the RSA Algorithm encryption menu, at first, the image converted to Base-64 String then the result is encrypted by using the RSA algorithm.

2.6 Steps for encrypting the ciphertext:
   1. Convert an image to Base-64 String as a plaintext
   2. Get the public key (e,n).
   3. Encryption the plain key using RSA algorithm, which formula is \( c = m^e \pmod{n} \)
   4. Save the result value of ciphertext.

Then on the RSA Algorithm Decryption menu, get the result of the encrypted file. Then, the ciphertext is decrypted using the RSA algorithm to get the original image.

2.7 Steps for decrypting the cipher key:
   1. Get the private key (d).
2. Decrypt the ciphertext using RSA algorithm, which formula is \( m = c^d \mod n \).

3. Convert the result into an image.

Figure 1 shows the plain image that converted to plaintext by Base-64 String.

Figure 1. The plain image that converted to plaintext

Figure 2 shows the encryption/decryption of (a) Hybrid RSA-OTP and (b) RSA Algorithm.

Figure 2. (a) Encryption/Decryption Hybrid RSA-OTP  
Figure 2. (b) Encryption/Decryption RSA Algorithm

3. Results and Discussions

The experiments were created on Windows 10 with Intel Core i5 processor, 64-bit architecture, and 4096MB RAM using Android Studio for coding java and performed on Android with Octa-Core processor, Android 7.1.1 Nougat System Operation, and 3072MB RAM. The results of the experiments of each set are presented in Tables 1 and two as follows.

| Resolution (pixel) | OTP Encryption plaintext (ms) | RSA Encryption plainkey (ms) | RSA Decryption cipherkey (ms) | OTP Decryption ciphertext (ms) |
|--------------------|-------------------------------|-----------------------------|-------------------------------|-------------------------------|
| 64 x 64            | 135                           | 4158                        | 2482                          | 115                           |
| 128x128            | 136                           | 10332                       | 5113                          | 128                           |
| 256x256            | 163                           | 27423                       | 11205                         | 143                           |

Table 1. Time Process For Hybrid RSA-OTP.
Table 2. Time Process For RSA Algorithm.

| Resolution (pixel) | RSA Encryption plaintext (ms) | RSA Decryption ciphertext (ms) |
|--------------------|-------------------------------|-------------------------------|
| 64 x 64           | 3990                          | 2419                          |
| 128x128           | 10290                         | 4989                          |
| 256x256           | 27195                         | 10519                         |
| 512x512           | 74764                         | 28001                         |

4. Conclusions
In conclusion, the integrity of the image is maintained, with the result showed that running time complexity for Rivest-Shamir-Adleman for encryption and decryption is $\Theta(n \log(m)^2 p)$, where $n$ means plainkey’s length and $n$ means cipher key's length. The running time complexity for One-Time-Pad for encryption and decryption is $\Theta(m)$, where $m$ means plaintext’s length and $m$ means ciphertext’s length.

5. Acknowledgments
The authors gratefully acknowledge that the present research is supported by Fund Dissemination IPTEKS Research Results for Lecturers / Researchers Universitas Sumatera Utara.

References
[1] Abel, M.W and Chung, S M 2015 Defending One-Time Pad Cryptosystems from Denial-of-Services Attacks
[2] Aiswarya P. M., Archana Raj, Dona John, Liya Martin, Sreenu G 2016 Binary RSA Encryption Algorithm
[3] Panda Prabhat K., Chattopadhyay Sudipta 2017 A Hybrid Security Algorithm for RSA Cryptosystem
[4] Rivest, A. Shamir, and L. Adleman 1978 A Method for Obtaining Digital Signatures and Public-Key Cryptosystem
[5] Tang Songsheng and Liu Fuqiang 2012 A one-time pad encryption algorithm based on one-way hash and conventional block cipher
[6] Thakkar, Jaydip 2015 An Encryption and Decryption More Secure Elgamal Cryptosystem International Journal of Science Technology & Engineering 1 (12)
[7] Dian Rachmawati et al 2018 IOP Conf. Ser.: Mater. Sci. Eng. 300 012042
[8] Dian Rachmawati and Yeni Rosalin Munthe 2018 J. Phys.: Conf. Ser. 1090 012062