Three-pass protocol scheme for bitmap image security by using vernam cipher algorithm

D Rachmawati*, M A Budiman, L Aulya
Departemen Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, Jl. Universitas No. 9-A, Medan 20155, Indonesia.

*E-mail: dian.rachmawati@usu.ac.id

Abstract. Confidentiality, integrity, and efficiency are the crucial aspects of data security. Among the other digital data, image data is too prone to abuse of operation like duplication, modification, etc. There are some data security techniques, one of them is cryptography. The security of Vernam Cipher cryptography algorithm is very dependent on the key exchange process. If the key is leaked, security of this algorithm will collapse. Therefore, a method that minimizes key leakage during the exchange of messages is required. The method which is used, is known as Three-Pass Protocol. This protocol enables message delivery process without the key exchange. Therefore, the sending messages process can reach the receiver safely without fear of key leakage. The system is built by using Java programming language. The materials which are used for system testing are image in size 200x200 pixel, 300x300 pixel, 500x500 pixel, 800x800 pixel and 1000x1000 pixel. The result of experiments showed that Vernam Cipher algorithm in Three-Pass Protocol scheme could restore the original image.

1. Introduction
Nowadays, sending images via internet network is very susceptible to the technological crimes such as interruptions, intercepts, and modifications. Without a security guarantee, intruder can get the message easily.

Image is one of multimedia in the form of visual information. Images are created from pixels. These pixels consist of colors namely: red, green and blue referred to as RGB [3]. Image security is indispensable because there have been many cases related to images modified by intruder. For that reason, the image is very necessary to be secured and kept confidential in order not to be abused by unauthorized persons.

Various security techniques have been developed to protect and maintain the confidentiality of messages, one of them is cryptographic techniques. Cryptography is both the science and arts of maintaining the message’s security by encoding it into a form that can no longer be understood.

Many algorithms can be used for cryptographic process, but in this research the author will discuss about Vernam Cipher algorithm. The Vernam Cipher algorithm uses a stream cipher where the cipher is derived from the XOR operation between the plaintext bits and the key bits [1]. The security of the Vernam Cipher algorithm relies heavily on the key exchange process. If the key is leaked, the security of this algorithm will collapse. Therefore, a method that minimizes the key leakage during the exchange of messages is required. The method used is known as Three-Pass Protocol. This protocol enables message delivery without key exchange [4]. So the messages sent can reach the receiver safely
without fear of key leakage. By using a fast internet connection, Three-Pass Protocol scheme is very applicable.

In this modern era, smartphones have grown rapidly around the world. The public adaptation of smartphones is very fast, especially with the presence of the first version of the operating system Android which was officially released in September 2008 [5]. Android gets attention from people around the world including Indonesia. A survey in April-May 2013 found that Android is the most popular platform that make it the most widely used smartphone operating system in the world.

2. Method

This research used Vernam Cipher algorithm in Three-Pass Protocol scheme for securing bitmap image.

2.1. Vernam Cipher Algorithm

Vernam Cipher is a cryptographic algorithm invented by Major J. Maugborne and G. Vernam in 1920’s [1]. The Vernam Cipher algorithm is adopted from One Time Pad Cipher, in which case characters are replaced by bits (0,1). In other words, Vernam Cipher is another version of One Time Pad Cipher [1].

Vernam Cipher algorithm is a symmetric of the key type cryptography algorithm. The key used to perform encryption and decryption use same key [7]. For binary messages, OTP encryption uses the XOR operation between each bit of the message with bits corresponding to the private key [1]. In this method, plaintext is converted into ASCII code and then subjected to XOR operation against keys that have been converted into the ASCII code.

Using One Time Pad, each key letter is used once for a single message and it is not reused. The key character stream length is equal to the message length. One time pad contains rows of randomly generated key characters. One pad is only used once to encrypt the message, after which the pad that has been used is destroyed (hence called one time). One encryption is only possible if both sender and receiver are in the same ownership of the key. Therefore, both parties must exchange their keys before the sending process begins. This means that safe communication is expected and planned within a certain time frame [6].

In general, the Vernam Cipher algorithm scenario is as follows:
1. Alice and Bob agree to use a key to encrypt and decrypt the message.
2. Then the key is used to encrypt messages from Alice to Bob.
3. Bob decrypts the message from Alice using the key that they had agreed on earlier, so that the original message can be read.

Here is the encryption and decryption formula of Vernam Cipher:

\[
C \equiv P \oplus K \tag{1}
\]

\[
P \equiv C \oplus K \tag{2}
\]

where C is ciphertext, P is plaintext, and K is key.

2.2. Three-Pass Protocol

Three-Pass Protocol is a framework that allows the sender to send encrypted messages to the recipient without need to distribute the sender's key to the receiver [4]. This process is referred to as Three-Pass Protocol because the sender and receiver perform three exchange stages to encrypt the message.

The protocol was developed by Adi Shamir in 1980’s [2]. The basic concept of Three-Pass Protocol is that each party has a private encryption key and a personal decryption key. Both parties independently use the key to encrypt messages and to decrypt messages [2]. The scheme of the Three-Pass Protocol can be seen in figure 1.
Three-Pass Protocol has several stages to deliver messages from the sender to the recipient. Here are the steps:

a. Alice (the message sender) has a $K_A$ key. Alice encrypts the message with her own key and sends an encrypted message ($C_1$) to Bob (the recipient of the message).

b. Bob (recipient of the message) has a $K_B$ key. Bob encrypts $C_1$ message with his private key and sends the encrypted message again ($C_2$) to Alice.

c. Alice who has received $C_2$ decrypts the message by using his private key and sends the decrypted message ($C_3$) to Bob.

d. Bob who has received $C_3$ also decrypts the message with his own private key to get the original message ($P$) sent by Alice.

Example:
- The value of RGB pixel (3.3) is (26,8,95) with sender key ($K_A$) : 00010101 and recipient key ($K_B$) : 00011101, the calculation process of encryption and decryption can be seen in figure 2.

**Figure 1. Scheme of Three-Pass Protocol**

**Figure 2. 3x3 pixels**

1. First change value of RGB from decimal to binary:

|   | R   | G   | B   |
|---|-----|-----|-----|
| Decimal | 26  | 8   | 95  |
| Binary  | 00011010 | 00001000 | 0101111 |

Then obtained “00011010 00001000 01011111” as plaintext ($P$).

2. Then encrypt plaintext with sender key ($K_A$) as follows:

| Table 1. First stage pixel encryption calculation process |
|---------------------------------|-----------------|-----------------|---------------|
| Plaintext ($P$)                 | Key ($K_A$)     | $C_1 = (P \oplus K_A)$ | Decimal       |
| 00011010                        | 00010101        | 00001111         | 15            |
| 00001000                        | 00010101        | 00011101         | 29            |
| 01011111                        | 00010101        | 01001010         | 74            |

3. After $C_1$ is obtained, encrypt $C_1$ with recipient key ($K_B$) as follows:
Table 2. Second stage pixel encryption calculation process

| Ciphertext ($C_1$) | Key ($K_B$)       | $C_2 = (C_1 \oplus K_B)$ | Decimal |
|------------------|------------------|-------------------------|---------|
| 00001111         | 00011101         | 00010010                | 18      |
| 00011101         | 00011101         | 00000000                | 0       |
| 01001010         | 00011101         | 01010111                | 87      |

4. After $C_2$ is obtained, encrypt $C_2$ with sender key ($K_A$) as follows:

Table 3. Third stage pixel encryption calculation process

| Ciphertext ($C_2$) | Key ($K_A$)       | $C_3 = (C_2 \oplus K_A)$ | Decimal |
|------------------|------------------|-------------------------|---------|
| 00010010         | 00010101         | 00000111                | 7       |
| 00000000         | 00010101         | 00010101                | 21      |
| 01010111         | 00010101         | 01000010                | 66      |

5. After $C_3$ is obtained, encrypt $C_3$ with recipient key ($K_B$) as follows:

Table 4. Calculation original pixel decryption

| Ciphertext ($C_3$) | Key ($K_B$)       | $P = (C_3 \oplus K_B)$ | Decimal |
|------------------|------------------|-------------------------|---------|
| 00000111         | 00011101         | 00011010                | 26      |
| 00010101         | 00011101         | 00001000                | 8       |
| 01000010         | 00011101         | 01011111                | 95      |

Table 4 shows that results of the pixel decryption is suitable to the pixels before encryption that is "26,8,95". It means, the use of Vernam Cipher algorithm in Three-Pass Protocol met parameters of data integrity.

3. Results and Discussions
The experiments were conducted on Windows 7 Notebook with Intel inside processor, 32-bit architecture, and 2048MB RAM. The development environment used for coding Java programming language is IDE Eclipse. And to run the execution, we used android on Xiaomi Redmi 4A with Qualcomm Snapdragon 425 quad-core 1.4 GHz processor and 2048MB RAM. The results of the experiments are presented in tables 5 and 6.

Table 5. The results of algorithm testing with process time.

| Image 200 x 200 pixels | Original image | First Ciphertext ($C_1$) | Second Ciphertext ($C_2$) | Third Ciphertext ($C_3$) | Decryption |
|------------------------|----------------|--------------------------|---------------------------|--------------------------|------------|
|                        | ![Image](image1.png) | ![Image](image2.png)     | ![Image](image3.png)     | ![Image](image4.png)     |            |
| Time Process (second)  | 0.285          | 0.254                    | 0.244                     | 0.249                    |            |
| Average Time (second)  |                |                          |                           |                          | 0.258      |

| Image 300 x 300 pixels | Original image | First Ciphertext ($C_1$) | Second Ciphertext ($C_2$) | Third Ciphertext ($C_3$) | Decryption |
|------------------------|----------------|--------------------------|---------------------------|--------------------------|------------|
|                        | ![Image](image1.png) | ![Image](image2.png)     | ![Image](image3.png)     | ![Image](image4.png)     |            |
| Time Process (second)  | 0.385          | 0.354                    | 0.344                     | 0.349                    |            |
| Average Time (second)  |                |                          |                           |                          | 0.358      |
In Table 5, there are five columns. The first column is the original image with a variety of sizes, the second column is the result of first stage image encryption, the third column is the result of second stage image encryption, the fourth column is the result of third stage image encryption, and the fifth column is the result of the original image decryption. From the testing of encryption and decryption process using image size 200x200 pixels, 300x300 pixels, 500x500 pixels, 800x800 pixels, and 1000x1000 pixels, the average time for each execution is 0.258 seconds, 0.53075 seconds, 1.431 seconds, 3.68725 seconds, and 5.66125 seconds. Figure 3 illustrates the acquired time relationship of encryption process to the size of an image.

**Figure 3.** Graph of the relationship between the size of image with the time
Figure 3 shows that the time required to encrypt image 200x200 pixel is 0.3 seconds, 300x300 pixel is 0.5 seconds, 500x500 pixel is 1.4 seconds, 800x800 pixel is 3.7 seconds, and the 1000x1000 pixel is 5.7 seconds. In Figure 3, it could be seen that the relationship between the pixel size of an image is linearly straight to the time of the encryption process. That is, the larger the pixel size, the time spent on program execution will also be longer.

| Resolution (pixel) | MSE | PSNR  |
|--------------------|-----|-------|
| 200 x 200          | 0   | Infinity |
| 300 x 300          | 0   | Infinity |
| 500 x 500          | 0   | Infinity |
| 800 x 800          | 0   | Infinity |
| 1000 x 1000        | 0   | Infinity |

In table 6, it could be seen that original image and reconstruction image are exactly the same.

4. Conclusion
In conclusion, encryption and decryption process on image security using Vernam Cipher algorithm in Three-Pass Protocol scheme has fulfilled the integrity of data. The pixel size of the image is linearly straight with time. The larger the image pixel size, the longer it takes for encryption process. Furthermore, the image of the reconstruction is the same as the original image before encryption with the value of MSE and PSNR 0 and infinity.

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] Nagaraj N 2012 Commun Nonlinear Sci Numer Simulat 17 pp 4029-4036.
[2] Oktaviania B and Siahaan A P U 2016 IOSR Journal of Computer Engineering (IOSR-JCE) 18:4 pp 26–29
[3] Rachmawati D, Amalia, Surya J 2017 IOP Conf. Series: Journal of Physics: Conf. Series 801
[4] Rahim, R and Ikhwana A 2016 International Journal of Science and Research (IJSR) 5:11 pp 102–104.
[5] Saparkhojayev N, Shakhov E, Mailybayev Y 2016 Journal of Physics:Conference Series 710
[6] Sharma S and Gupta V 2013 Int. Journal of Innovative Research in Science, Engineering and Technology 2:6 pp 2248-2250
[7] Zaeniah and Purnama B E 2015 Int. Journal of Advanced Computer Science and Applications 6:9 pp 292–297