Optimizing Digital Image Steganography to Enhance the Security of Secret Message Delivery

Eko Heri Susanto
Cyber Security Engineering, Army Polytechnic, Batu-Malang, Indonesia
Dimas Pramudya Pratama
Cyber Security Engineering, Army Polytechnic, Batu-Malang, Indonesia
Ricki Septian Nurpratama
Cyber Security Engineering, Army Polytechnic, Batu-Malang, Indonesia

Abstract: Image steganography is a technique used to hide secret messages in image media. In this research, 2 method improvements are proposed, namely (1) layered message security utilizing vigenere encryption with asymmetric keys, and (2) optimizing the hiding capacity (HC) of secret messages, utilizing the bit shift operation method. From the test results it can be proven that layered message security has been successfully implemented, and the capacity of secret messages can be increased. From the initial 4 bits per pixel it increased to 8 bits per pixel. However, the steganography image quality of the proposed method is not better than the PVDMF method.

Keywords: Steganography, Least Significant Bit, Modified Bit, Asymmetric Key Encryption Generator, Information Hiding.

Introduction

Steganography is a technique used to conceal secret messages within seemingly ordinary media, such as images. The significance of images in steganography lies in their visual nature, allowing secret messages to be embedded without attracting the attention of others (Jain, 2020). Research in this field holds great importance due to the increasing need for secure communication. Through the use of image steganography, secret messages can be embedded into images, appearing as ordinary pictures without arousing suspicion of the message's presence. This is crucial in various contexts, such as data security, communication privacy, and criminal investigations (Mandal, 2019). The military's utilization of steganography is

Correspondents Author:
Eko Heri Susanto, Department of Cyber Security Engineering, Army Polytechnic, Indonesia
Email : ekoheri@gmail.com

Received January 12, 2024; Revised February 6, 2024; Accepted February 15, 2024; Published March 27, 2024

highly significant as it provides a means to communicate sensitive information without alerting the enemy (ALBaaj, 2022).

Research discussing various techniques for hiding messages in digital images is also increasingly widespread. One of them is a technique for manipulating bits whose influence is not significant, the term Least Significant Bit (LSB). The LSB-based steganography technique has the advantage of producing steganography image quality that is almost the same as the original image (Putra, 2018). To implement this LSB technique, there are many methods. One approach is to utilize data mapping and LSB substitution methods (Zakaria, 2018). However, this data mapping method has a complex level of algorithmic complexity.

A simpler method approach is Pixel Value Differentencing (PVD) and Modulus Function (MF). This PVDMF method was successfully improved with two method variants, namely pixel value differencing and modulus function variant 1 and variant 2 (PVDMF 1 and PVDMF 2). PVDMF 1 variant produces a high peak signal-to-noise ratio (PSNR), but has a low hiding capacity (HC). Meanwhile, the PVDMF 2 method variant produces low PNSR, but has a higher concealment capacity (HC) (Sahu, 2019). The use of the PVD and MF methods succeeded in increasing the insertion of secret messages by 4 bits per pixel, and was still able to maintain image quality after inserting the message (Darwis, 2020).

Basically the PVDMF 1 and PVDMF 2 variant approaches work on two consecutive pixel blocks. So to insert 1 letter, it takes 2 pixels. However, the increase in concealment capacity (HC) has not been correlated to the colored RBG channels. There is still an opportunity to improve this method by correlating colored RGB channels to increase hiding capacity (Sahu, 2019) (Darwis, 2020). Apart from that, this research has not discussed the techniques for hiding messages with layered security.

Research that aims to apply layered security methods also already exists. One of the methods applied is combining LSB-based steganography with Vigenere encryption (Purba, 2021). However, the weakness in this research lies in the Vigenere Cipher encryption, namely if the key length is not the same as the plaintext length, then the key will be repeated until it is the same as the plaintext length, this of course makes it easier for cryptanalysts to carry out the cryptanalysis process (Subandi, 2017). Therefore, in other research, the Vigenere encryption method was tried to be improved. The improvement method is to apply a three-layer key, namely by combining the random function, Euler numbers and the Blum Blum Shub algorithm (Wibowo, 2022). However, even this research still has weaknesses. Because the encryption key used is still a symmetric key, the security factor for distributing the encryption key can be a big problem.

To increase the security of this encryption key, there has been other research that uses the approach of using two keys, one in the form of a normal string and the other based on insertion time (Wijaya, 2023). However, there is still a weakness, namely that the encryption key in the encryption key distribution process is in the form of a normal string. This distribution of encryption keys in the form of normal strings can be improved by combining On time Pad (OTP) with Message Digest 5 (MD5). Where improvements to this method have been proven to be able to be implemented for classic Vigenere encryption (Putra, 2023). Even though these two studies are better than before, they still have problems, namely the encryption key distribution mechanism. The fundamental weakness of these two methods is that the encryption key is still a symmetric key.

Based on the summary of existing research, the research problem (RP) that can be described is as follows:

- **RP1**. To further secure secret messages inserted into images, classic encryption can be applied, namely Vigenere cipher. This substitution encryption technique is effective when combined with steganography, because ciphertext encryption does not change the number of characters that will be inserted into the cover image. However, for most substitution encryption, including Vigenere, the encryption key is of the symmetric key type. The fundamental weakness of symmetric keys is the security factor when distributing the encryption key itself.
- **RP2**. If you use the LSB steganography technique, the character capacity that can be inserted into the image is not large, because it only uses the last 1 bit of the vessel image. However, if the pixel value difference and modulus function (PVDMF) approach is applied, the number of bits that can be accommodated is only 4 bits per pixel (BPP).

From these two research problems, the research questions (RQ) that can be explained are as follows:

- **RQ1**. How to create asymmetric keys (private key and public key) to apply to Vigenere encryption (text substitution ecryption)?
- **RQ2**. How to develop a bit shifting operation algorithm that can maximize the capacity of characters to be inserted into the cover image?
- **RQ3**. How is the quality of the steganography image after applying the bit modification algorithm?

To answer research question number 1 (RQ1), researchers will apply a combination of several methods to produce asymmetric keys that will be applied to substitution text encryption. There are 6 types of combining methods, including (1) On Time Password which functions to

generate 4 random digit numbers, (2) Smallest Multiple Multiple (LCM), (3) Application of Secure Hash Algorithm (SHA) version 1, (4) Sengkalan Lamba, and (5) Implementing Lempel–Ziv–Welch (LZW) encoding, and (6) processing secret message encryption using the key resulting from method number 4. Henceforth the result of method number 4 or the result of the hash is called the private key. Meanwhile, the result of method number 5 is called the public key.

Meanwhile, to answer research question number 2 (RQ2), researchers will apply a bit modification method which consists of 2 steps, namely (1) bit masking operation, (2) bit injection using AND and OR operations.

Finally, to answer research question number 3 (RQ3), researchers will measure image quality by (1) analyzing Histogram Equalization, (2) comparing the Peak Signal-to-Noise Ratio (PNSR) between the cover image and the steganography image.

Research Method

In this discussion topic, researchers present two types of research contributions: (1) the implementation of text substitution (vigenere) encryption equipped with asymmetric keys, and (2) the optimization of the LSB bit shifting operation to accommodate a larger number of characters compared to the LSB with PVDMF method. The overall method flow in this research is illustrated in Figure 1 below.

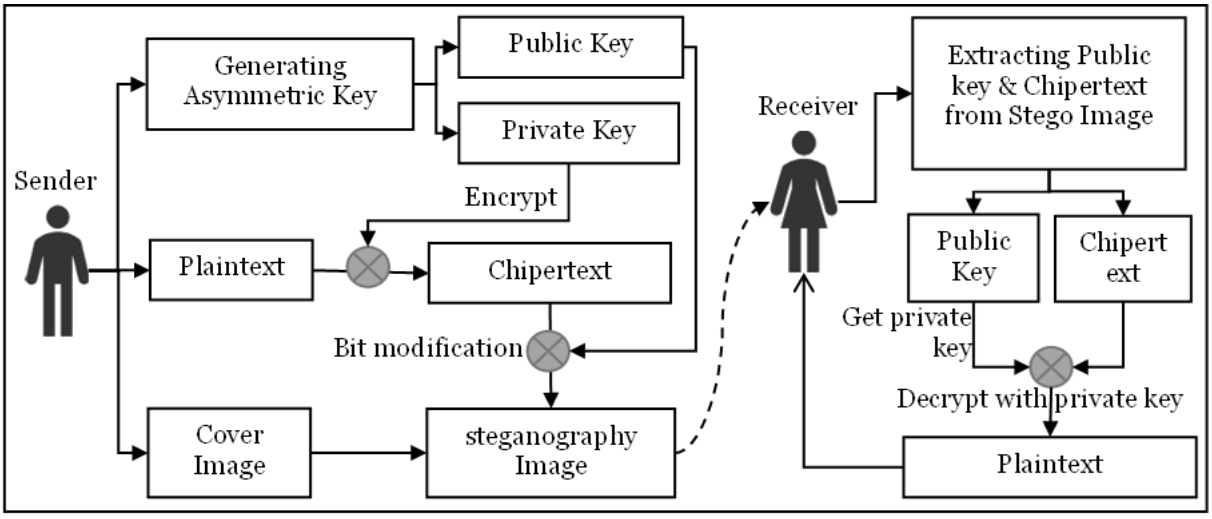


Figure 1 Model of steganography system

In accordance with the system flow as previously explained, the system design developed in this research, follows the following class diagram flow. The explanation of the class diagram can be seen in Figure 2 below.

Vigenere Encryption with Asymmetric Keys

The first research contribution is implementing vigenere encryption equipped with asymmetric keys (public key and private key). Researchers implemented it into 5 classes, namely (1). Key generator, (2). Sengkalan Encoding, (3). SHA-1 hashes, (4). Lempel–Ziv–Welch (LZW) encoding, and (5). Vigenere encryption.

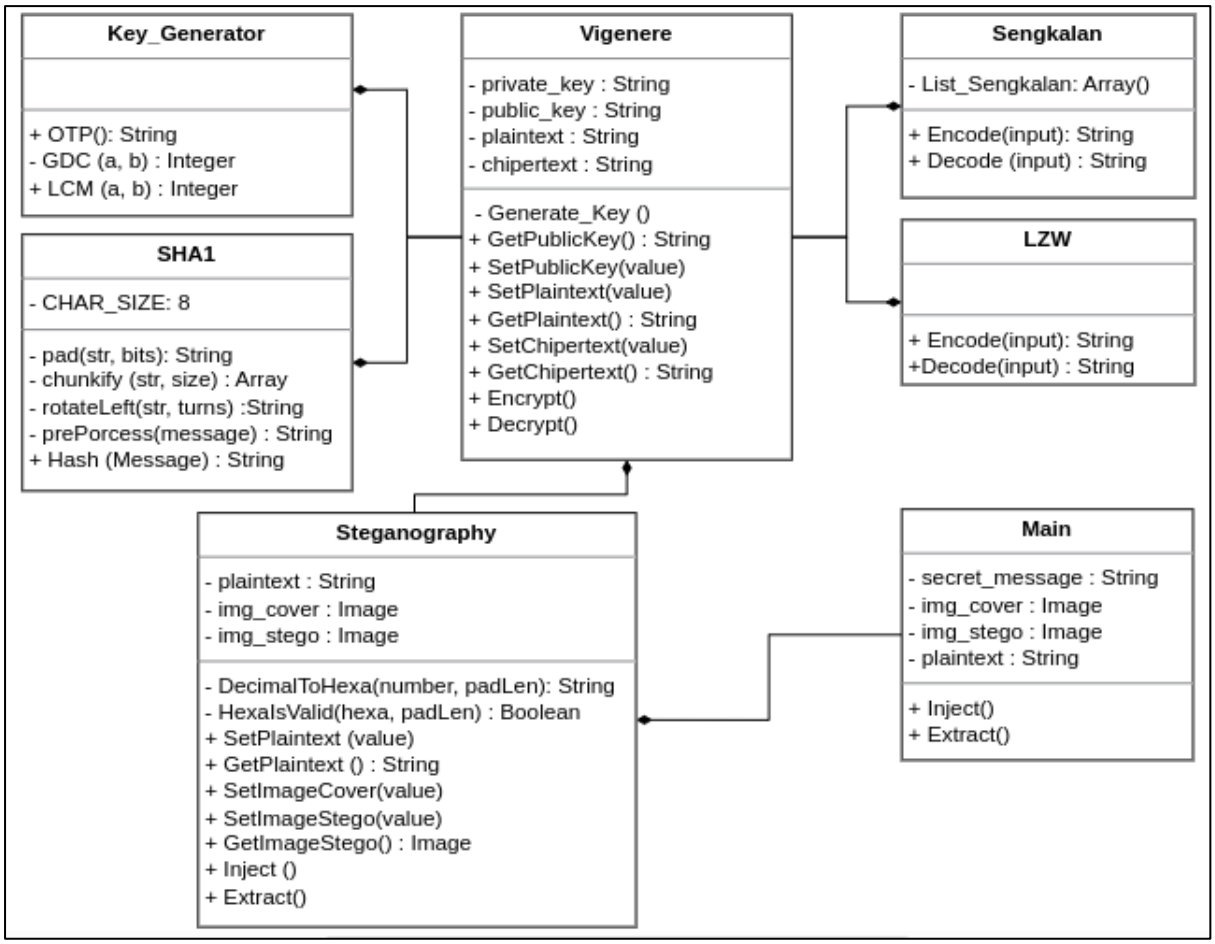


Figure 2 Class Diagram of Steganography System

A detailed explanation of each step in the first research contribution can be seen in the following discussion.

Stage 1 : Generating the On Time Password (OTP)

In stage 1, the system automatically generates 4 digit numbers randomly. The aim of implementing this method is that the encryption key always changes every time the encryption process is carried out. The mathematical equation at stage 1 looks like the formula for equation 1 below.

$$OTP = password + \sum_{i=1}^{4} IntToStr([[Math.Random() \times 10]]) \quad (1)$$

Where OTP is data in the form of a string consisting of 4 characters. This OTP is the original key, which will later be modified to become a private key and a public key.

Stage 2 : Calculate the Least Common Multiple (LCM) from OTP

The aim of stage 2 is to modify the OTP to produce numbers that are not the same as the OTP results. Next, the results of stage 2 will be used to form a private key, which will be implemented in substitution encryption. The mathematical equation at stage 2 looks like the formula for equation 2 below.

$$GCD(a,b) = \begin{cases} a & if\ b = 0 \\ GCD(a,b) & if\ b \neq 0 \end{cases} \quad (2)$$

$$LCM(a,b) = \frac{(a \times b)}{GCD(a,b)}$$

Where :
a = Lenght of String OTP number = 4
b = OTP number
gcd = greatest common division

Stage 3 : Generating a hash from the Result of Least Common Multiple (LCM)

In stage 3, this is the stage for forming the actual private key, where this private key will later be used in the encryption stage. To further secure this **private key**, the mechanism applied is to randomize (hash) the results of the process in step 2. To randomize (hash) in this research the Secure Hash Algorithm version 1 (SHA-1) method is used. The researcher did not discuss the SHA1 process flow in detail, because the researcher utilized the SHA-1 function which is available in Request For Comments (RFC) 3174 (Eastlake, 2001). The mathematical equation at stage 3 looks like the formula for equation 3 below.

(3)

1. Initialization
   a) $H_0, H_1, H_2, H_3, H_4 = are\ the\ initial\ values\ used\ to\ initialize$
2. Message padding
   a) The input message is padded to a form that can be processed, typically by adding padding and information about the message length
3. Message Block Splitting:
   a) The padded message is split into smaller blocks, usually 512 bits each.
4. Message Block Expansion:
   a) Each message block is expanded into 80 words of 32 bits.
5. Variable Initialization:
   a) Internal variables like *A,B,C,D,E* are initialized with the values H_0, H_1, H_2, H_3, H_4
6. Message Block Processing:
   a) Involves a series of bitwise logical operations, bit rotations, and non-linear functions.
7. Updating Hash Values:
   a) After processing each message block, the values of H_0, H_1, H_2, H_3, H_4 are updated with the results of the processing.
8. Iteration:
   a) The processing of message blocks is iterated until the entire message is processed.
9. Private Key (PvK) : The final hash values H_0, H_1, H_2, H_3, H_4 are taken as the SHA-1 hash output.

Stage 4: Encoding and Decoding the OTP into a Sengkalan Sentence

Sengkalan is a culture that originates from the people of the island of Java, Indonesia. Sengkalan expressions are forms of expressions of Javanese society. The definition of sengkalan is sentences and words that have the character of numbers. From these sentences or words, year numbers are composed as written on the gate of the house or cemetery (Adi, 2014). Examples of characterizing numbers into the word sengkalan are as shown in Table 1 below.

Table 1 Symbol of Sengkalan Sentence

Symbol	List of Sengkalan Sentences
0	Akasa, Awang-Awang, Barakan, Ilang, Sirna, etc.
1	Badan,Budha,Budi, Bumi,Candra,Karta, etc.
2	Apasang, Asta, Athi-athi, Buja, Bujana, etc.
3	Agni, Api, Apyu, Bahni, Benter, etc.
4	Bun, Catur, Dadya, Gawe, Karta, etc.
5	Angin, Astra, Bajra, Bana, Bayu, etc.
6	Amla, Anggana, Anggang-Anggang, Amnggas, Artati, etc.
7	Acala, Ajar, Angsa, Ardi, Arga, etc.
8	Anggusti, Astha, Bajul, Basu, Basuki, etc.
9	Ambuka, Anggangsir, Angleng, Angrong, Arum, etc.

An example of a sengkalan sentence is "Sirna Ilang Kartaning Bumi", if translated literally the meaning is "The prosperity of the earth has disappeared." But the real meaning is not that. The meaning according to the science of sengkalan is that "Sirna" is the symbol for the number 0, "ilang" is the symbol for the number 0, "kartaning" or "karta" is the symbol for the number 4, and "bumi" (earth) is the symbol for the number 1. When arranged, the numbers become 0041. If read from back to front, it becomes the number 1,400. This figure indicates the year 1,400 Saka or 1.478 AD (Adi, 2014).

In this research, the Sengkalan rule is used to show the character of the On Time Password. By adding the contribution of combining Sengkalan with On Time Password, a public key can be generated that is different from the private key. On this basis, to encode the Sengkalan method into a public key encryption security system, it can be described in a mathematical equation such as equation 4 below. For the purposes of the decryption process, this "sengkalan" sentence must be returned (decode) to the OTP. The mathematical equation for the decoding process of this Sengkalan sentence looks like equation 4 below

$$L = \begin{bmatrix} a01\ a02 \ldots a0n \\ a11\ a12 \ldots a1n \\ \ldots\ldots \\ a91\ a92 \ldots a9n \end{bmatrix} \quad (4)$$

Encode :

$$Encode(input) = Sengkalan + \sum_{k=0}^{n-1} L[ordinal(input, k)][Math.random() \times n]$$

Where :
Input = the OTP, example 1400.
L = List of Sengkalan Sentence. Example a01 = Sirna, a02 = Ilang, a11 = Bumi, etc
n = number of words for each character number "sengkalan".
Sengkalan = Sengkalan sentence. Example input = 1400 then Sengkalan sentence = Bumi Karta Sirna Ilang.

Decode :

$$Decode(input) = \sum_{i=0}^{n-1} \left(\sum_{rows=0}^{9} \sum_{cols=0}^{100} cols \cdot \delta(L[rows][cols], input[i]) \cdot \delta(found, false) \cdot \delta(rows, 9) \right) \cdot 10^{n-1-i}$$

Where :
input = Sengkalan sentences. Example "Sirna Ilang Karta Bumi"
n = number of words for each character number "sengkalan".
rows = number of rows of sengakalan sentence symbols (0..9)
cols = number of columns of sengkalan sentences for each symbol
L = List of Sengkalan Sentence. Example a01 = Sirna, a02 = Ilang, a11 = Bumi, etc

Stage 5 : Encode and Decode the Sengkalan Sentence using the LZW Algorithm

The sentence "sengkalan" can actually indicate a public key which will later be translated into a private key which is the same as the key to encrypt a secret message. However, if the sentence is shared as is, it is feared that other people will easily guess it. Moreover, if someone understands Javanese literary vocabulary, then that person will definitely know the true meaning of the sengkalan sentence. For security reasons, the distribution of the "sengkalan" sentence (public key) needs to be engineered first. For this reason, in this study the sentence "sengkalan" needs to be encoded first. Later the encoded results will be distributed. In this research, the type of encode chosen is the Lempel–Ziv–Welch (LZW) method (Krokosz, 2023). The Lempel-Ziv-Welch (LZW) encoding algorithm can be mathematically represented in equation 5 below.

(5)

Encode(s) = Encode_output

1. Initialize *D* as a dictionary with single ASCII characters (0-255) as the initial dictionary.
2. Initialize *p* with the first character of *s*.
3. For each yet-to-be-processed character *c* in *s* :
   a) If *p*+*c* is present in the dictionary:
      i. Update *p* to *p*+*c*.
   b) If not:
      i. Output the dictionary index of *p*.
      ii. Add *p*+*c* to the dictionary.
      iii. Update *p* to *c*.
4. Output the dictionary index of *p*.

The Lempel-Ziv-Welch (LZW) decoding algorithm can be represented mathematically as seen in equation 6 below.

Decode(encoded_output) = s (6)

1. Initialize D as a dictionary with single ASCII characters (0-255) as the initial dictionary.
2. Initialize s with the first character of encoded_output
3. Initialize p with the first character of encoded_output.
4. For each index k in encoded_output(starting from the second index):
   a) If k is not in the dictionary:
      i. Initialize c with the first character of p.
   b) If k is in the dictionary:
      i. Initialize c with the string associated with index k.
   c) Output c.
   d) Add p+c[0] to the dictionary.
   e) Update p to c.

Stage 6 : the Secret Message Encryption

The mathematical equation for this encryption process looks like equation 6 below.

1. **Generate Key :** (7)

$$K_{result} = K + \sum_{i=1}^{n} (K_i + index)$$

Where

K_{result}: is the generated key

K : original key

K_i : is the i-th character in the key

n : is the number of iterations required for the key length to reach the message length

$index$: is the character index in the key

2. **Encryption :**

$$C_i = (M_i + K_i)\ Modulus\ 256$$

3. **Decryption:**

$$M_i = (C_i - K_i)\ Modulus\ 256$$

Where :

C_i : is the i-th character in the ciphertext

M_i : is the i-th character in a plain text messaget

K_i : is the i-th character in the key

The final step in the secret message encryption phase is to scramble the secret message using the private key obtained from stage 4. In this research, of course the private key available is a hash code of 160 bits, or 40 hexadecimal characters. This number corresponds to the number of characters resulting from the SHA-1 algorithm.

Optimization of the LSB using Bit Shifting Operation

In the PVDMF method the number of modified bits is only 3 bit per pixel, where the modified bits are in the colors Red, Green and Blue. Meanwhile, for the improved method proposed in this research, the number of modified bits is 8 bits per pixel. The modified bits in the proposed method are in the colors Red, Gren, Blue and Alpha. To get a complete picture of the differences between these two methods, you can look at Figure 3 below.

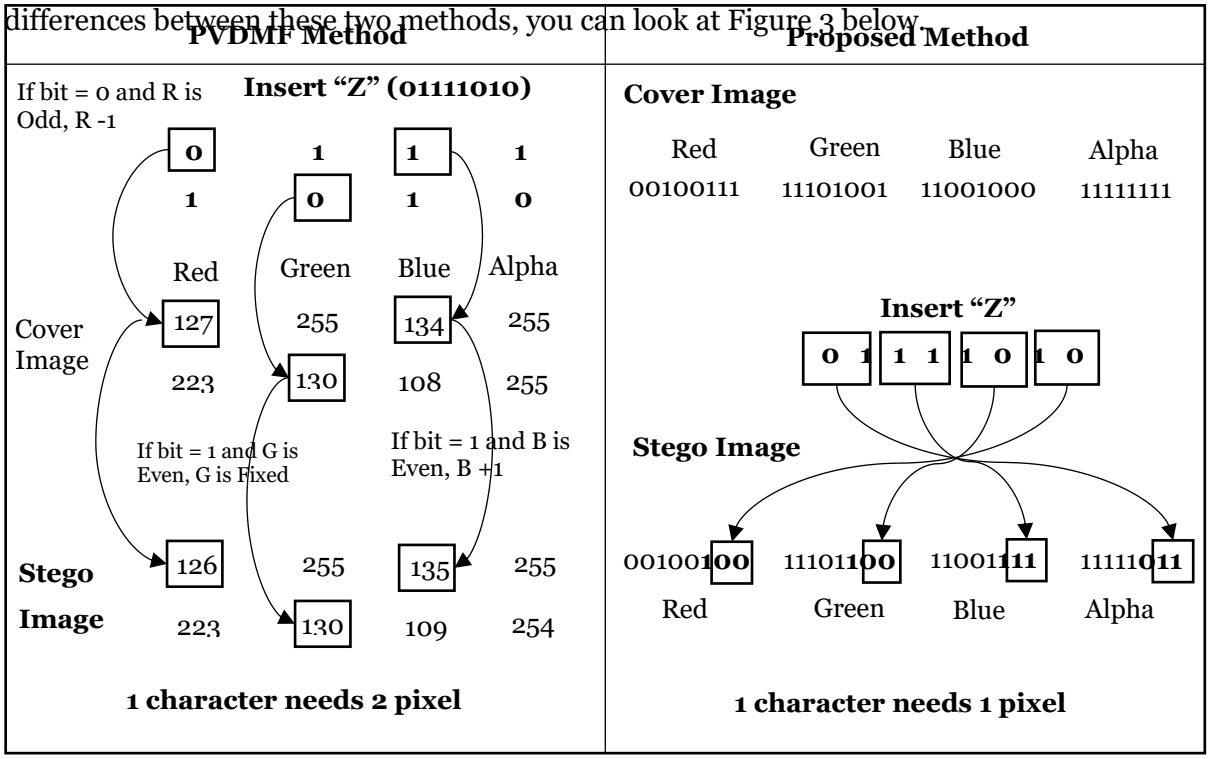


Figure 3 Comparison of PVDMF Methods with Proposed Methods

To implement this steganography method, researchers apply a bit modification method which consists of 2 steps, namely (1) bit masking operation, (2) bit injection using AND and OR operations. The first step, namely the bit shift operation, aims to shift the secret message bits so that their position matches the desired color location. The next step is to insert the secret message bit into the cover image. For this insertion, researchers utilized AND and OR logical operations. The mathematical equation for this encoding process looks like equation 8 below.

1. **Initialization :** (8)
   a) img = read image from cover image
   b) pixel = Convert from image array 3D to array 1D
2. **Convert Message Length to Hexadecimal Format :**
   a) data_length = format(len(secret_message),'06h')
   b) decimal_msg = encode_utf8(data_length + secret message)
3. **Insert Message into Pixel**:
   For i in $0 \leq i < length(decimal\_msg)$
   a) Masking bit Operation :

i. *asciiCode=decimal_msg[i].*

ii. $\_maskR = (asciiCode \wedge 3) \ggg 0$

iii. $\_maskG = (asciiCode \wedge 12) \ggg 2$

iv. $\_maskB = (asciiCode \wedge 48) \ggg 4$

v. $\_maskA = (asciiCode \wedge 192) \ggg 6$

b) Bit Injection

i. $pixel[4 \times i + 0] = (pixel[4 \times i + 0] \wedge 252) \vee$ _maskR

ii. $pixel[4 \times i + 1] = (pixel[4 \times i + 1] \wedge 252) \vee$ _maskG

iii. $pixel[4 \times i + 2] = (pixel[4 \times i + 2] \wedge 252) \vee$ _maskB

iv. $pixel[4 \times i + 3] = (pixel[4 \times i + 3] \wedge 252) \vee$ _maskA

4. **Result :**

a) img = Convert from array pixel 1D to array 3D

b) Save to file ('secret_image.png')

The hidden message data format looks like Figure 4 below

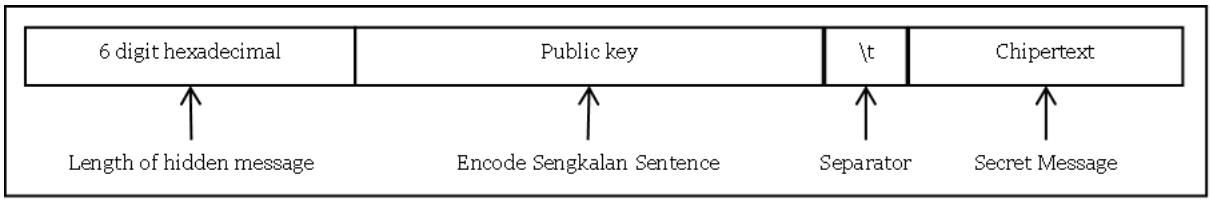


Figure 4 the Hidden Message Data Format

To extract secret messages from images, researchers applied the AND operation method and shifted the bits from the Alpha, blue, green and red pixels. The mathematical equation for this encoding process looks like equation 9 below.

1. **Initialization :** (9)

a) img = read image from stego image

b) pixel = Convert from image array 3D to array 1D

c) i = 0, completed=false, hidden_msg = empty string, exctract_bit = 0, number_of_hidden_msg = 0

2. **Extract bit into String Hidden Message**:
while $(i < length(pixel)$ *AND* $completed = false)$

a) **Get 2 digit of bit LSB from Pixel Alpha-Blue-Green-Red :**

i. $\_2bit\_A = (pixel[(4 \times i) + 0] \wedge 3) \lll 0$

ii. $\_2bit\_B = (pixel[(4 \times i) + 1] \wedge 3) \lll 2$

iii. $\_2bit\_G = (pixel[(4 \times i) + 2] \wedge 3) \lll 4$

iv. $\_2bit\_R = (pixel[(4 \times i) + 3] \wedge 3) \lll 6$

b) **Extract 8 digit of bit to ASCII :**

I. $asciiCode = \big(((\_2bit\_A \vee \_2bit\_B) \vee \_2bit\_G) \vee \_2bit\_R\big)$

c) **Assembly ASCII to String Hidden Message:**

i. $hidden\_msg \mathrel{+}= chr(asciiCode)$

ii. $extract\_bit \mathrel{+}= 1$

d) **Calculate the length of the hidden message**

i. $if\ (length\ of\ hidden\_msg = 6)\ AND\ (number\_of\_hidden\_msg\ =\ 0)$

1. $number\_of\_hidden\_msg = Convert\ to\ Integer(hidden\_msg)$

2. $hidden\_msg = empty\ string$

ii. $else\ if\ (extract\_bit \geq (number\_of\_hidden\_msg + 6))$

1. $completed = true$

3. **Completed extract of Hidden Message :**

a) *Return hidden_msg*

Result and Discussion

The proposed research is implemented in the Javascript language (ECMAScript 6). The source code can be accessed at this URL address: https://github.com/ekoheri/Optimizing_Steganography. The program in this research supports Graphical User Interface (GUI). The experimental study was carried out in four stages using a GUI. These steps are given as follows.

1. Cover Image: In this stage, the cover image is uploaded from the camera or file by using the GUI.
2. Secret message : The user uploads the message that he wants to hide in the cover image by manual.
3. Encryption and Decryption : In this phase, the secret message is encrypted using text substitution with a private key, while the decryption process uses a public key. The use of different keys is then called asymmetric encryption.
4. Hide data by shifting the LSB bit: In this phase, the encrypted secret message is inserted into the cover image.
5. Analysis: This section provides data analysis about the stego image. The PSNR value (Hore, 2010) is calculated to measure the quality of the stego image compared to the cover image. The PSNR value calculation is given in Equation 10.

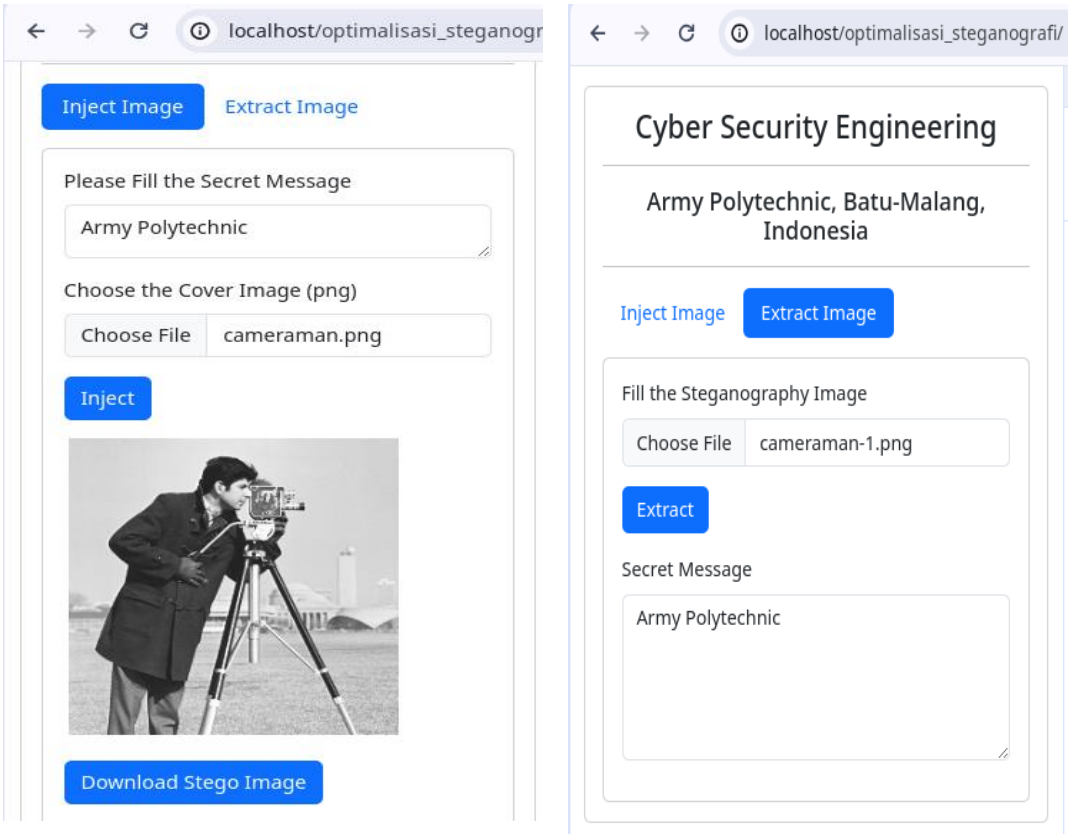
$$PNSR = 10 \times log_{10}\left(\frac{255^2}{MSE}\right) \tag{10}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

Figure 5 presents these images used in experimental studies. In Figure 6, a sample image of the GUI Steganography is presented.

(a) (b)

Figure 5 Cover Image (a) Cameraman (b) Lena



(a) (b)

Figure 6 GUI for Steganography (a) Encode (b) Decode

Encryption and Decryption

The researchers' first trial used input data as the cover image cameraman.png, and the secret message "Army Polytechnic" totaling 16 bytes. The encoding and decoding process looks like Figure 7 below.

```
===== INPUT ======
Image Cover      : cameraman.png
Secret Message   : Army Polytechnic

===== ENCODE =====
OTP              : 5611
LCM(4, 5611)     : 22444
Private Key      : 4e26477ebb88ce2f8bbeb214d509d2db5f987042
Chipertext       : u×¯T¦ÑÛÖËÓÉ
Sengkalan        : Lek Maha Karengya Bajra
Public key       : MBAlBArBAgAANBAhBAoBAhBAgAALBAhBAyBAlBAuBAnBA5BAHEACBAhBAqBAyBAhBA
Hidden Message   : 000053MBAlBArBAgAANBAhBAoBAhBAgAALBAhBAyBAlBAuBAnBA5BAHEACBAhBAqBAyBAhBA» u×¯T¦ÑÛÖËÓÉ

 ===== DECODE =====
Public Key       : MBAlBArBAgAANBAhBAoBAhBAgAALBAhBAyBAlBAuBAnBA5BAHEACBAhBAqBAyBAhBA
Chipertext       : u×¯T¦ÑÛÖËÓÉ
Sengkalan        : Lek Maha Karengya Bajra
Origin Key (OTP) : 5611
LCM(4, 5611)     : 22444
Private Key      : 4e26477ebb88ce2f8bbeb214d509d2db5f987042
Plaintext        : Army Polytechnic
```

Figure 7 First Trial Results

Next, the researcher repeated the testing process again with the same data input as the first test. The test results look like Figure 8 below.

```
===== INPUT ======
Image Cover      : cameraman.png
Secret Message   : Army Polytechnic

 ===== ENCODE =====
OTP              : 6479
LCM(4, 6479)     : 25916
Private Key      : f8993b5615024f5875e30816bc38910b49e5a2d5
Chipertext       : §ª¦²S²¤¢ª©Ô
Sengkalan        : Guwa Kaswareng Marna Nem
Public key       : HBA1BA3BAhBAgAALBAhBAzBACEAyBAlBAuBAnBAgAANBAhBAyBAuBADEAOBAlBAtBA
Hidden Message   : 000053HBA1BA3BAhBAgAALBAhBAzBACEAyBAlBAuBAnBAgAANBAhBAyBAuBADEAOBAlBAtBA» §ª¦²S²¤¢ª©Ô

 ===== DECODE =====
Public Key       : HBA1BA3BAhBAgAALBAhBAzBACEAyBAlBAuBAnBAgAANBAhBAyBAuBADEAOBAlBAtBA
Chipertext       : §ª¦²S²¤¢ª©Ô
Sengkalan        : Guwa Kaswareng Marna Nem
Origin Key (OTP) : 6479
LCM(4, 6479)     : 25916
Private Key      : f8993b5615024f5875e30816bc38910b49e5a2d5
Plaintext        : Army Polytechnic
```

Figure 8 Second Trial Results

From two trials as in Figures 7 and 8, different hidden message data (public key and ciphertext) were obtained, even though the input data was the same. The different public key and ciphertext are caused by randomization of numbers generated by the On Time Password (OTP) function. For example, in the first trial, the OTP obtained was '5611'. The number '5611' was changed to the phrase "sengkalan", which reads "Lek Maha Karengya Bajra". Meanwhile, in the second test, the OTP obtained was '6479'. From the OTP, the "sengkalan" sentence

obtained is "Guwa Kaswareng Marna Nem". Next, the sentence "sengkalan" was encoded using the LZW method. Where the results of the LZW encoding are used as the public key.

For encryption, it uses a private key obtained from calculating the Least Common Multiple (LCM) of 2 numbers, namely the length of the OTP and the OTP number itself. For the first trial, the OTP number obtained was '5611'. If the OTP number is calculated by LCM with the number 4, a new number is obtained, namely '22444'. Next, the new number '22444' is randomized using the SHA-1 method. Meanwhile, in the second test, the OTP obtained was '6479'. So the LCM of the numbers 4 and 6479 is 25916. Next, the character '25916' is scrambled using the SHA-1 method. In the end, the results of the SHA-1 randomization are finally used as private keys.

The contribution to this research succeeded in answering the research question (RQ1), namely how to create asymmetric keys (private key and public key) to be applied to Vigenere encryption (text substitution)? From the test results, evidence was obtained that the text substitution-based encryption method could be equipped with asymmetric key security. Where the encryption process uses a private key, while the key distributed is a public key. However, the test results also show evidence that users do not need to share the public key. Because the public key is in the message hidden in the image. This is the second advantage of this research. If in previous research, one of the weaknesses of encryption with symmetric keys was the distribution of the encryption key (Subandi, 2017), then in this research, the weakness in the distribution of the encryption key was proven to be correctable. Where the method is improved is to provide an asymmetric encryption key. Figure 8 below shows an example of a message hidden in an image. Figure 9 below shows an example of a message hidden in an image.

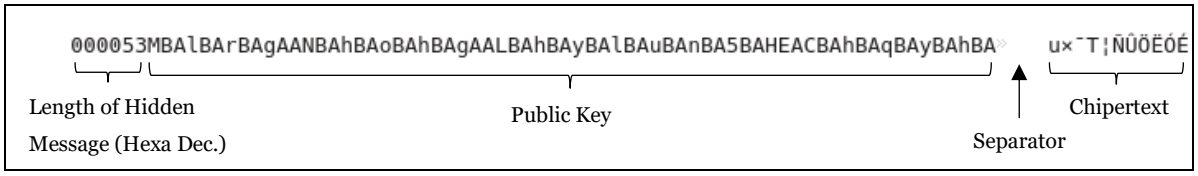


Figure 9 Example of a Hidden Message

The next test was carried out with the same secret message input, namely "Army Polytechnic" and the cover image 'lena.png'. The same as the two previous tests, the 3rd and 4th trials were also carried out twice. Figures 10 and 11 below show the test results.

```
 ===== INPUT ======
Image Cover     : lena.png
Secret Message  : Army Polytechnic

 ===== ENCODE =====
OTP             : 1562
LCM(4, 1562)    : 3124
Private Key     : ddca0f7ce8e8438c7b9d01aa7ac8309704b6b871
Chipertext      : ¥ÖÐÚP¶¦ÏÞ¬Ê¡¡Æ
Sengkalan       : kembar Lidhah Indriya Lek
Public key      : rBAlBAtBAiBAhBAyBAgAAMBApBAkBAoBAhBAoBAgAAJBAuBAkBAyBApBA5BAhBAGEAlBArBA
Hidden Message  : 000059rBAlBAtBAiBAhBAyBAgAAMBApBAkBAoBAhBAoBAgAAJBAuBAkBAyBApBA5BAhBAGEAlBArBA
¥ÖÐÚP¶¦ÏÞ¬Ê¡¡Æ

 ===== DECODE =====
Public Key      : rBAlBAtBAiBAhBAyBAgAAMBApBAkBAoBAhBAoBAgAAJBAuBAkBAyBApBA5BAhBAGEAlBArBA
Chipertext      : ¥ÖÐÚP¶¦ÏÞ¬Ê¡¡Æ
Sengkalan       : kembar Lidhah Indriya Lek
Origin Key (OTP) : 1562
KPK(4, 1562)    : 3124
Private Key     : ddca0f7ce8e8438c7b9d01aa7ac8309704b6b871
Plaintext       : Army Polytechnic
```

Figure 10 Third Trial Results

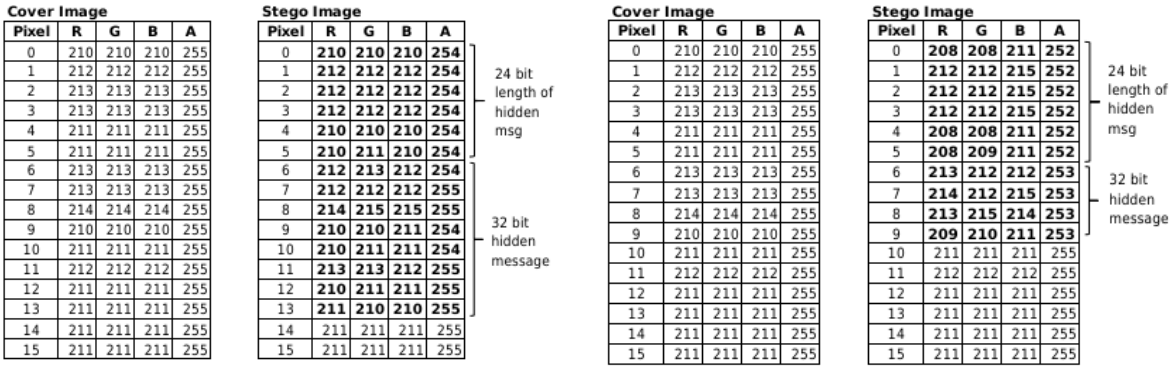
```
 ===== INPUT ======
Image Cover     : lena.png
Secret Message  : Army Polytechnic

 ===== ENCODE =====
OTP             : 2084
LCM(4, 2084)    : 2084
Private Key     : 4a67984f8d0fca27af38e2f50cc8bc252e072f49
Chipertext      : uÓ£°Y£Ò±ØÉËÏ
Sengkalan       : Karya Astha Muluk Nembeh
Public key      : LBAhBAyBA5BAhBAgAABBAzBA0BAoBAEEANBA1BAsBA1BArBAgAAOBAlBAtBAiBAlBAoBA
Hidden Message  : 000056LBAhBAyBA5BAhBAgAABBAzBA0BAoBAEEANBA1BAsBA1BArBAgAAOBAlBAtBAiBAlBAoBA
uÓ£°Y£Ò±ØÉËÏ
 ===== DECODE =====
Public Key      : LBAhBAyBA5BAhBAgAABBAzBA0BAoBAEEANBA1BAsBA1BArBAgAAOBAlBAtBAiBAlBAoBA
Chipertext      : uÓ£°Y£Ò±ØÉËÏ
Sengkalan       : Karya Astha Muluk Nembeh
Origin Key (OTP) : 2084
KPK(4, 2084)    : 2084
Private Key     : 4a67984f8d0fca27af38e2f50cc8bc252e072f49
Plaintext       : Army Polytechnic
```

Figure 11 Fourth Trial Results

Hide Data by Shifting the LSB bit

The contribution to this research also succeeded in answering research question 2 (RQ2), namely how to develop a bit modification algorithm that can maximize the capacity of characters to be inserted into the cover image? If in previous research the capacity was able to increase by 3,1 bits per pixels (Sahu, 2019), and 4 bits per pixels (Darwis, 2020). Then in this research the capacity can be increased to 8 bits per pixel (1 byte per pixel). So if in previous research, 2 pixels were needed to insert 1 character (8 bits), then in this research 1 character (8 bits) can be inserted into 1 pixel (8 bits per pixel). A comparison of the number of pixels that change when using the PVDMF (Darwis, 2020) methods with the proposed method is shown in Figure 12 below.



Pixel	R	G	B	A
0	210	210	210	255
1	212	212	212	255
2	213	213	213	255
3	213	213	213	255
4	211	211	211	255
5	211	211	211	255
6	213	213	213	255
7	213	213	213	255
8	214	214	214	255
9	210	210	210	255
10	211	211	211	255
11	212	212	212	255
12	211	211	211	255
13	211	211	211	255
14	211	211	211	255
15	211	211	211	255

Pixel	R	G	B	A
0	**210**	**210**	**210**	**254**
1	**212**	**212**	**212**	**254**
2	**212**	**212**	**212**	**254**
3	**212**	**212**	**212**	**254**
4	**210**	**210**	**210**	**254**
5	**210**	**211**	**210**	**254**
6	**212**	**213**	**212**	**254**
7	**212**	**212**	**212**	**255**
8	**214**	**215**	**215**	**255**
9	**210**	**210**	**211**	**254**
10	**210**	**211**	**211**	**254**
11	**213**	**213**	**212**	**255**
12	**210**	**211**	**211**	**255**
13	**211**	**210**	**210**	**255**
14	211	211	211	255
15	211	211	211	255

Pixel	R	G	B	A
0	210	210	210	255
1	212	212	212	255
2	213	213	213	255
3	213	213	213	255
4	211	211	211	255
5	211	211	211	255
6	213	213	213	255
7	213	213	213	255
8	214	214	214	255
9	210	210	210	255
10	211	211	211	255
11	212	212	212	255
12	211	211	211	255
13	211	211	211	255
14	211	211	211	255
15	211	211	211	255

Pixel	R	G	B	A
0	**208**	**208**	**211**	**252**
1	**212**	**212**	**215**	**252**
2	**212**	**212**	**215**	**252**
3	**212**	**212**	**215**	**252**
4	**208**	**208**	**211**	**252**
5	**208**	**209**	**211**	**252**
6	**213**	**212**	**212**	**253**
7	**214**	**212**	**215**	**253**
8	**213**	**215**	**214**	**253**
9	**209**	**210**	**211**	**253**
10	211	211	211	255
11	212	212	212	255
12	211	211	211	255
13	211	211	211	255
14	211	211	211	255
15	211	211	211	255

Figure 12 Comparison of the number of pixels (a) PVDMF (b) proposed method

Based on the description above, the contribution of this research has succeeded in correcting the weaknesses of previous studies. The first improvement is to successfully apply a layered security method to the secret message inserted into the cover image. In previous research (Zakaria, 2018), this layered security method had not been implemented. The second improvement is to successfully apply asymmetric keys to the text substitution encryption method. Where in previous research (Wibowo, 2022), a symmetric encryption key security system was still implemented. Meanwhile, the third improvement was successful in increasing the capacity of bits per pixel. If in previous research it was only possible to insert 4 bits per pixel (Darwis, 2020), then in this research it can be increased to 8 bits per pixel (1 byte per pixel).

However, this research also still has weaknesses. The first weakness is that the number of characters inserted into the image increases, because the characters inserted are the public key encoding, plus the encryption (ciphertext) of the secret message. If the initial number of secret message letters is only 16 bytes, then when you add the public key it will increase to 89 bytes.

Image Quality Analysis

From the results of four trials as in Figures 7, 8, 10 and 11, different hidden message data (public key and ciphertext) were obtained, even though the input data was the same. When the message is inserted into the cover image, the quality of the steganography image is not the same. This is proven by the PNSR test on the two steganography images, as shown in table 2 below.

Table 2 PSNR of Steganographic Images

Cover Image	Secret Message	Test No.	Hidden Message	PNSR
Cameraman.png (256x256 px)	Army Polytechnic	1	000053MBAlBArBAgAANBAhBAoBAhBAgAALBAhBAyBAlBAuBAnBA5BAHEACBAhBAqBAyBAhBA u×□¯T□¦ÑÛÖ□□ËÓ□É	35.9564 dB
		2	000053HBA1BA3BAhBAgAALBAhBAzBACEAyBAlBAuBAnBAgAANBAhBAyBAuBADEAOBAlBAtBA §ª¦²S²¤¢ª©□□□Ô□□	36.0673 dB
Lena.png (512x512 px)	Army Polytechnic	3	000059rBAlBAtBAiBAhBAyBAgAAMBApBAkBAoBAhBAoBAgAAJBAuBAkBAyBApBA5BAhBAGEAlBArBA ¥ÖÐÚP¶¦ÏÞ¬Ê□□¡¡Æ	39.1482 dB
		4	000056LBAhBAyBA5BAhBAgAABBAzBAoBAoBAEEANBA1BAsBA1BArBAgAAOBAlBAtBAiBAlBAoBA uÓ£°Y□£Ò±Ø□ÉËÏ□□	39.1684 dB

The next analysis is to compare the PNSR and MSE values of steganography images using the PVDMF methods with the proposed method. To avoid differences in PNSR and MSE values, hidden messages are not encrypted. As for how to analyze the image quality, it can be seen from these two values. The higher the PNSR value, the better the image quality. On the other hand, the lower the MSE value, the better the image quality. The results of the trials measuring the MSE and PNSR values can be seen in table 3 below.

Table 2 Comparison of PNSR and MSE of PVDMF Methods with the Proposed Method

Cover Image	Secret Message	Method	PNSR	MSE
Cameraman.png (256x256 px)	Army Polytechnic	PVDMF	41.4030 dB	0.00034
		Proposed Method	39.7116 dB	0.00074
Lena.png (512x512 px)	Army Polytechnic	PVDMF	44.5128 dB	0.00013
		Proposed Method	42.0935 dB	0.00024

From the test results, evidence was found that the steganography image quality of the proposed method was no better than previous methods (PVDMF). It can be seen from the PNSR value of the proposed method, it is lower than the previous method. Meanwhile, the MSE value of the proposed method is higher than the previous method.

Security Analysis

In this research, layered data security has been successfully implemented. It was proven from four trials that different encryption data (ciphertext) was obtained, even though the plaintext data was the same. However, the encrypted data in the image can still be cracked. In this

research, the distribution of public keys is included in the picture. By cracking the public key, the encrypted data can be cracked.

In this research, the public key that is distributed is only encoded using the Lempel-Ziv-Welch (LZW) method. Where the LZW algorithm itself is a common algorithm, and many people definitely know about it. If the public key is successfully decoded by an unauthorized person, then that person can get a "sengkalan" sentence. For people who understand Javanese literature, it is not difficult to translate the "sengkalan" sentence into a series of numbers. Where this series of numbers will form the private key. To fix this security gap, the public key should be encrypted first before being inserted into the image. For example, it is encrypted using the Caesar Chiper method, Reverse Chiper, etc.

Conclusions

In this research, a layered security method was successfully implemented, where the secret message was encrypted first before being inserted into the cover image. In general, text substitution encryption methods only provide symmetric keys. However, in this research it was successfully improved by providing asymmetric keys (public key and private key) for text substitution encryption. From the test results it can be proven that the message data hidden in the image, namely the public key and ciphertext, can be different, even though the input data is the same. The difference between public keys and ciphertext is caused by the randomization of numbers generated by the On Time Password (OTP) function. Due to the differences in message data, the resulting PNSR (Peak Signal-to-Noise Ratio) values are also different, even though the input data is the same.

Apart from that, this research also succeeded in increasing the insertion capacity (HC) of secret message characters, by 8 bits per pixel (1 byte per pixel), by utilizing the RGBA channel. In previous research using the PVDMF method, the insertion capacity (HC) was only 3.1 bits per pixel (BPP). However, if analyzed in terms of image quality, the proposed method is no better than previous research, namely Pixel Value Differentiating and Modulus Function (PVDMF). It is proven from the test results that the PNSR and MSE values are worse.

Acknowledgements

We would like to thank the Army Polytechnic, Batu-Malang, Indonesia for supporting and facilitating this research.

References

R.C. Jain, "A Survey of Image Steganography Techniques", International Journal of Advanced Science and Technology, 2020

P.C. Mandal at al, (2022. "Digital image steganography: A literature survey" , Elsevier, Information Sciences Volume 609, Pages 1451-1488, https://doi.org/10.1016/j.ins.2022.07.120, https://www.sciencedirect.com/science/article/abs/pii/S002002552200809X

ALBaaj, Ghassan Faisal Falih, "A Model to Share Hidden Data in Image for Military Access", Texas Journal of Engineering and Technology 2022, Accessed on : https://zienjournals.com/index.php/tjet/article/download/701/569/735

Putra, Randi Rian, at al, (2018), "Implementation of LSB Steganography on Embedding Messages in Digital Image", Article in International Journal of Scientific Research in Science and Technology, Volume 4, Issue 11, Print ISSN: 2395-6011, Online ISSN: 2395-602X, DOI : https://doi.org/10.32628/IJSRST18401112

Zakaria, Abdul Alif, at al, (2018). High-Capacity Image Steganography with Minimum Modified Bits Based on Data Mapping and LSB Substitution, MDPI, Appl. Sci. 2018, 8, 2199; doi:10.3390/app8112199

D. Darwis, N. B. Pamungkas, Wamiliana, (2020). Comparison of Least Significant Bit Pixel Value Differencing and Modulus Function on Steganography to Measure Image Quality Storage Capacity and Robustness, Journal of Physics: Conference Series, Volume 1751, The 3rd International Conference on Applied Sciences Mathematics and Informatics (ICASMI)

Sahu, A.K., Swain, G. (2019)An Optimal Information Hiding Approach Based on Pixel Value Differencing and Modulus Function. Wireless Pers Commun 108, 159–174. https://doi.org/10.1007/s11277-019-06393-z

Purba, Lia Cintia at al, (2021).Penggunaan Algoritma LSB dan Vigenere untuk Pengamanan Data melalui Pola Citra Digital, TECHSI: Vol. 13, No. 2, DOI: https://doi.org/10.29103/techsi.v13i2.5162

Subandi, Amin, at al, (2017). Three-Pass Protocol Implementation in Vigenere Cipher Classic Cryptography Algorithm with Keystream Generator Modification, Advances in Science, Technology and Engineering Systems Journal Vol. 2, No. 5, 1-5, DOI: 10.25046/aj020501

Wibowo, Sutrasno Andre, at al, (2022). Teknik Informatika, FakultaGenerator Kunci Tiga Lapis Pada Algoritma Vigenere Menggunakan Fungsi Random, Bilangan Euler dan Metode Blum Blum Shub, Jurnal Ilmiah Informatika Komputer Volume 27 No. 1, https://doi.org/10.35760/ik.2022.v27i1.6145

Wijaya, Kevin, at al, (2023). Time-Based Steganography Image with Dynamic Encryption Key Generation, Procedia Computer Science 227 (2023) 233–242, Elsevier B.V., DOI : https://doi.org/10.1016/j.procs.2023.10.521

Putra, Muhammad Andika, at al, (2023). Securing Text File Using Combination of Vigenere and One–Time Cipher Algorithm Pad Cipher Algorithm, Procedia Computer Science 227 (2023) 1030–1038, Elsevier B.V., DOI : https://doi.org/10.1016/j.procs.2023.10.612

Eastlake, D., at al, (2001). US Secure Hash Algorithm 1 (SHA1), Network Working Group, Request for Comments : 3174

Krokosz, Tomasz, Jarogniew Rykowski, Małgorzata Zajęcka, Robert Brzoza-Woch, and Leszek Rutkowski. 2023. "Cryptographic Algorithms with Data Shorter than the Encryption Key, Based on LZW and Huffman Coding" Sensors 23, no. 17: 7408. https://doi.org/10.3390/s23177408

Adi, Febrian Wisnu, (2014).Sengkalan, Makna Penanda Dalam Bentuk Kalimat atau Gambar Indah Sebagai Bahasa Komunikasi Seni, CORAK Jurnal Seni Kriya Vol. 2 No.2, Nopember-April 2014

Hore, A. , Ziou, D., “Image quality metrics: PSNR vs. SSIM”, In Pattern recognition (icpr) 20th international conference on IEEE, 2366-2369, 2010.