Use of One Time Pad Algorithm for Bit Plane Security Improvement

Suhardi¹, Saib Suwilo², Erna Budhiarti Nababan³

¹²³Department of Computer Science and Information Technology University of North Sumatera Medan, Indonesia

¹suhardi147038064@students.usu.ac.id

Abstract. BPCS (Bit-Plane Complexity Segmentation) which is one of the steganography techniques that utilizes the human vision characteristics that cannot see the change in binary patterns that occur in the image. This technique performs message insertion by making a switch to a high-complexity bit-plane or noise-like regions with bits of secret messages. Bit messages that were previously stored precisely result the message extraction process to be done easily by rearranging a set of previously stored characters in noise-like region in the image. Therefore the secret message becomes easily known by others. In this research, the process of replacing bit plane with message bits is modified by utilizing One Time Pad cryptography technique which aims to increase security in bit plane. In the tests performed, the combination of One Time Pad cryptographic algorithm to the steganography technique of BPCS works well in the insertion of messages into the vessel image, although in insertion into low-dimensional images is poor. The comparison of the original image with the stegoimage looks identical and produces a good quality image with a mean value of PSNR above 30db when using a large-dimensional image as the cover messages.

1. Introduction
Steganography is a technique that the secret data is hidden into vessel image without any suspicion[4]. Steganography is the art of hiding information that cover the messages in a vessel image [1][8]. Information security is very important in steganography because the information can only be known by the recipient. Bit-Plane Complexity Segmentation (BPCS) is one of the steganography techniques that replace all of the noise-like regions in the bit-planes of the vessel image with secret information[6]. Storing secret information into bit planes directly results in the process of extracting information becomes easier so that confidential information can be known by others. Therefore bits plane need to be modified. Security techniques to improve the security system can be done by modifying structure noise like regions[5]. One way to modify the bit plane is by combining it with encryption. It can generate a new bit order or cipher block that make extracting information more difficult.

2. Theories
2.1. Bit Plane Complexity Segmentation (BPCS)
Bit-plane complexity segmentation (BPCS) is a steganography technique introduced by Eiji Kawaguchi and Richard O. Eason in 1998. This technique is a steganography technique that has a large capacity, because it can accommodate confidential data with a relatively large capacity when compared with the steganography method Such as LSB (Least Significant Bit). This BPCS technique is a steganography technique that is not based on programming techniques, but a technique that uses the nature of human vision. The nature of human vision that is utilized is the inability of humans to interpret a very complicated binary pattern. The inserted data capacity can reach 50% from the size of the cover- image [3]. Eiji Kawaguchi and R. O. Eason introduced this BPCS technique for use on uncompressed color image documents with BMP format. The image document is divided into segments of 8x8 pixels per segment [3]. In an 8-bit image document, each segment will have 8 plane
bits representing the pixels of each bit. The process of dividing the 8x8 pixel segment into 8 pieces of bit plane is called bit slicing process. Representation of the eight bit plane is a PBC system (Pure Binary Code). In BPCS, the insertion process is performed on a bit plane with a CGC (Canonical Gray Code) system because bit slicing process on CGC tends to be better than PBC [3]. So in the insertion process, the bit plane with the PBC representation is converted into a plane bit with a CGC representation. The main principle of BPCS technique is the binary image is divided into informative region and noise-like region. The secret information is hidden in noise-like region of the vessel image[4].

2.2. **One Time Pad (OTP)**

One Time Pad algorithm (OTP) is a symmetric key type algorithm which means that the key used to perform the encryption and decryption is the same key. In order for OTP not to be solved, the private key must be as long as the message, very random and can only be used once therefore the name "One-Time" Pad[7]. In the encryption process, this algorithm uses a stream cipher derived from the XOR result between the plaintext bits and the key bits. In this method plain text is converted into ASCII code and then subjected to XOR operation against the changed key into ASCII code. Data encryption is the first part of the data security process. In this encryption process the original data will be process randomization with a predetermined algorithm, while the encryption process can be done by using the formula below:

\[ \text{Ciphertext}(i) = \text{Plaintext}(i) \oplus \text{Key}(i) \]  

Decryption process is a process done to restore files from the form of symbols back to their original form. Decryption can be done using the formula below:

\[ \text{Plaintext}(i) = \text{Ciphertext}(i) \oplus \text{Key}(i) \]

3. **Bit Plane Security Improvement**

For security enhancements to BPCS we utilize the One Time Pad cryptography algorithm to modify the original message bits before the message is inserted. Combination of Algorithm can be seen in fig.1
4. Result and Discussion
The image pixels are converted to numbers 0 through 255 that represent the intensity of the color. They are converted into pure Binary Code (PBC) systems. Each row and column represents the pixel position in the image. Image segmentation can be seen in Table 1. Each segment consists of 64 pixels.

Table 1. 8x8 pixel of Image Light Intensity.

| Pixel Value | 143 | 108 | 150 | 160 | 114 | 145 | 143 | 142 |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|
| 142         | 107 | 143 | 134 | 51  | 60  | 54  | 51  |     |
| 145         | 111 | 156 | 144 | 40  | 42  | 45  | 48  |     |
| 135         | 101 | 154 | 146 | 37  | 40  | 53  | 58  |     |
| 144         | 103 | 155 | 150 | 39  | 43  | 54  | 55  |     |
| 144         | 97  | 148 | 148 | 45  | 51  | 59  | 58  |     |
| 147         | 101 | 151 | 152 | 54  | 59  | 68  | 70  |     |
| 148         | 104 | 153 | 155 | 56  | 60  | 67  | 72  |     |

All pixels must be converted into binary numbers to be a Pure Binary Code (PBC). Once converted into a PBC form then converted again into Canonical Gray Code (CGC). Changes from PBC to CGC because in BPCS Steganography, image representation with CGC is better than using PBC system. Each pixel in the CGC segment will be split into bits to form several bit-planes. These changes will be shown in Table 2 and Table 3 and PBC Change Process to CGC example from row 0 and column 0 can be seen in fig 2

Table 2. Pure Binary Code (PBC).

| PBC          | 10001111 | 01101100 | 10010110 | 10100000 | 01110010 | 10010001 | 10001111 | 10001110 |
|--------------|----------|----------|----------|----------|----------|----------|----------|----------|
| 10001110     | 01101011 | 10001111 | 10001110 | 00110011 | 00111100 | 00110110 | 00110011 |
| 10010001     | 01111111 | 10011000 | 00101000 | 00101010 | 00101101 | 00101100 | 00101110 |
| 10000111     | 01100101 | 10011010 | 10010100 | 00100101 | 00101000 | 00101010 | 00101101 |
| 10010000     | 01100011 | 10011100 | 00110011 | 00111011 | 00111101 | 00111010 | 00111011 |
| 10010111     | 01101000 | 10011000 | 10011100 | 00110101 | 00110011 | 00110111 | 01000010 |
| 10010010     | 01101000 | 10011000 | 10011100 | 00111000 | 00111100 | 00111110 | 01001000 |

Table 3. Canonical Gray Code (CGC).

| CGC          | 11001000 | 01011010 | 11011101 | 11110000 | 01001011 | 11011001 | 11010001 | 11001001 |
|--------------|----------|----------|----------|----------|----------|----------|----------|----------|
| 11001001     | 01011110 | 11001000 | 11001101 | 00110100 | 00110010 | 00110100 | 00110101 | 00110100 |
| 11011001     | 01110000 | 11010010 | 11011000 | 00111100 | 00111110 | 00111010 | 00111010 | 00110100 |
| 11000000     | 01010011 | 11010111 | 11011101 | 00110110 | 00111100 | 00101110 | 00101100 | 00101100 |
| 11010000     | 01010001 | 11011110 | 11011110 | 00111010 | 00101010 | 00100100 | 00100100 | 00100100 |
| 11010100     | 01010111 | 11011100 | 11010100 | 00101101 | 00100100 | 01000100 | 01000100 | 01000100 |
| 11011100     | 01011100 | 11010101 | 11010110 | 00100100 | 00100010 | 01000010 | 01000010 | 01001100 |
The next stage is the Bit Plane formation stage generated by bit slicing process from CGC value in one segment. For one segment the value of CGC will produce 8 Bit Plane. One Bit Plane will be generated from bits with the same index on each pixel. Each of bit field has a complexity value. Threshold is the limit to determine whether bit fields can be inserted by message. Complexity is measured by the number of times the bit changes from 0 to 1 and from 1 to 0. The value of the maximum bit change of the 8x8 block is 112. The formula for calculating the complexity is shown in equation 3 and The division of the Bit Plane is shown in fig3.

\[ \alpha = \frac{k}{n} \]  

Where :
\( \alpha \) = Complexity Value  
\( k \) = Total of Bit Change  
\( n \) = Maximum Change Value of 8 x 8 Bit-Plane

![Figure 2. PBC to CGC process.](image)

![Figure 3. Bit Planes.](image)
The value of complexity for all Bit Plane has a boundary value between the Informative Region and the Noise-Like Region. The value of this boundary is called Threshold. Basically, the standard threshold value used is 0.3. If the Bit Plane’s complexity value is greater than the threshold value, the Bit Plane belongs to the Noise-Like Region and if its complexity value is less than the threshold value, then the Bit Plane belongs to the Informative Region. The Complexity Value for all Bit Plane can be seen in Table 3.

**Table 4. Complexity Value of All Bit Planes.**

| Bit Plane | Total of Bit Change | Complexity Value |
|-----------|---------------------|------------------|
| 1         | 28                  | 0.250000000      |
| 2         | 15                  | 0.133928571      |
| 3         | 14                  | 0.125000000      |
| 4         | 27                  | 0.241071429      |
| 5         | 46                  | 0.410714286      |
| 6         | 44                  | 0.392857143      |
| 7         | 59                  | 0.526785714      |
| 8         | 61                  | 0.544642857      |

In Table 4 can be seen that Bit Plane 1, 2, 3 and 4 are Informative Region so that secret message cannot be inserted into this Bit Plane. The secret message will be inserted into the next Bit Plane which is a Noise-Like Region.

At the next stage the secret message to be inserted will be represented first in the binary value so it will be a Bit Plane. In this research we will utilize the One Time Pad algorithm to encrypt the Bit Plane first before it is inserted. In this case, the word to insert is "KOMC2015". The word is eight characters. This will be one message block. The plain message and the bit-plane can be seen in Table 4 danTable.

**Table 5. Plain Message**

| Char | ASCII | Biner  |
|------|-------|--------|
| K    | 75    | 01001011 |
| O    | 79    | 01001111 |
| M    | 77    | 01001101 |
| C    | 67    | 01000011 |
| 2    | 50    | 00110010 |
| 0    | 48    | 00110000 |
| 1    | 49    | 00110001 |
| 5    | 53    | 00110101 |

**Table 6. Bit Plane of Message**

| Bit Plane Message |
|-------------------|
| 0 1 0 0 1 0 1 1   |
| 0 1 0 0 1 1 1 1   |
| 0 1 0 0 1 1 0 1   |
| 0 1 0 0 0 0 1 1   |
| 0 0 1 1 0 0 1 0   |
| 0 0 1 1 0 0 0 0   |
| 0 0 1 1 0 0 0 1   |
| 0 0 1 1 0 1 0 1   |
The next step is to set the key for the message to be encrypted. The key used in this algorithm is symmetric key. The keys used for encryption and decryption are the same. The key must also be converted into ASCII then converted to binary value. The key example used is "FASILKOM".

| Char | ASCII | Biner   |
|------|-------|---------|
| F    | 70    | 01000110|
| A    | 65    | 01000001|
| S    | 83    | 01010011|
| I    | 73    | 01001001|
| L    | 76    | 01001100|
| K    | 75    | 01001011|
| O    | 79    | 01001111|
| M    | 77    | 01001101|

Table 8. Bit Plane of Key

| Bit Plane of Key |
|------------------|
| 0 1 0 0 0 1 1 0  |
| 0 1 0 0 0 0 0 1   |
| 0 1 0 1 0 0 1 1   |
| 0 1 0 0 0 1 1 0   |
| 0 1 0 0 0 1 1 1   |
| 0 1 0 0 0 1 0 1   |
| 0 1 0 0 0 1 0 1   |
| 0 1 0 0 0 1 0 0   |

Each character of the secret message will be XOR with the key. This will produce a bit-plane cipher

Bit-plane ciphers will be inserted in the Noise-Like Region. After that, the bit-plane is restored to its original position and converted back into pure binary code before it is finally rewritten to the new stego image.

5. Experiment
In this paper, we experiment on 4 different resolution images as their vessel image and with 4 text files of different file sizes as secret messages. Each of secret message will be inserted into each of image file. The result of stego image and vessel image will be compared by calculating its PSNR value.
Table 10. Value of PSNR Results Comparison of Vessel Image and Stego Image.

| Vessel Image       | Message File                  | Stego Image       | PSNR Value |
|--------------------|-------------------------------|-------------------|------------|
| Hello Kitty.jpg(48x48) | Pesan1.txt(26 bytes)           | Hello Kitty+Pesan1.jpg | 39,717     |
| Hello Kitty.jpg(48x48) | Pesan2.txt(433 bytes)          | Hello Kitty+Pesan2.jpg | 19,126     |
| Hello Kitty.jpg(48x48) | Pesan3.txt(1.7 kb)             | Hello Kitty+Pesan3.jpg | 12,356     |
| Hello Kitty.jpg(48x48) | Pesan4.txt(3.55 kb)            | Hello Kitty+Pesan4.jpg | 12,421     |
| KFC.jpg(96x96)      | Pesan1.txt(26 bytes)           | KFC+Pesan1.jpg    | 46,475     |
| KFC.jpg(96x96)      | Pesan2.txt(433 bytes)          | KFC+Pesan2.jpg    | 29,204     |
| KFC.jpg(96x96)      | Pesan3.txt(1.7 kb)             | KFC+Pesan3.jpg    | 22,406     |
| KFC.jpg(96x96)      | Pesan4.txt(3.55 kb)            | KFC+Pesan4.jpg    | 19,584     |
| My Son.jpg(128x128) | Pesan1.txt(26 bytes)           | My Son+Pesan1.jpg | 53,089     |
| My Son.jpg(128x128) | Pesan2.txt(433 bytes)          | My Son+Pesan2.jpg | 43,141     |
| My Son.jpg(128x128) | Pesan3.txt(1.7 kb)             | My Son+Pesan3.jpg | 32,146     |
| My Son.jpg(128x128) | Pesan4.txt(3.55 kb)            | My Son+Pesan4.jpg | 29,246     |
| Singapore.jpg(256x256) | Pesan1.txt(26 bytes)         | Singapore+Pesan1.jpg | 59,926     |
| Singapore.jpg(256x256) | Pesan2.txt(433 bytes)          | Singapore+Pesan2.jpg | 41,029     |
| Singapore.jpg(256x256) | Pesan3.txt(1.7 kb)             | Singapore+Pesan3.jpg | 34,622     |
| Singapore.jpg(256x256) | Pesan4.txt(3.55 kb)            | Singapore+Pesan4.jpg | 27,515     |

Average of PSNR value 32,625

From table 9 has been proved that the utilization of one time pad to bit-plane in BPCS can improve the security of the message because the average value of PSNR resulting from testing of the stego image shows the value of 32.6256, in other words the resulting image has a good quality.

6. Conclusion
The One Time Pad cryptography algorithm can be combined well in steganography techniques of BPCS in terms of modifying message bits. Utilization of One Time Pad cryptography algorithm on steganography technique of BPCS is proven to increase security to message bits. This can be seen from the test results showing that the original message bits cannot be extracted directly but must go through the decryption stage first and the stego image looks identical to the original image and the value of PSNR resulting from the stego image shows the value above 30db or with Other words produce a good quality image.
References

[1] Johson N, Jajodia S 1998 Exploring Steganography: Seeing The Unseen IEEE Computer 26

[2] Katti J, Pote S, L B. K 2015 Two Level Encryption based on One Time Pad and Koblitz Method of Encoding International Journal of Computer Applications. 122 0975

[3] Kawaguchi E, Eason R O 1999 Principle and Applications of BPCS-Steganography SPIE

[4] Khaire S S, Nalbalwar S L 2010 Review: Steganography-Bit Plane Complexity Segmentation (BPCS) Technique International Journal of Engineering Science and Technology 2 4860

[5] Lahane P, Kumbhar Y, Patil S, More S, Barse M 2015 Visual Cryptography and BPCS Steganography for Data Shielding International Journal Of Engineering And Computer Science. 4 11997

[6] Mehta S, Dighe K, Jagtap M, Ekre A Web Based BPCS Steganography International Journal of Computer Technology and Electronic Engineering 2

[7] Nagaraj N 2012 One-Time Pad as a nonlinear dynamical system Commun Nonlinear Sci Numer Simul 17 4029

[8] Rudramath P R, Madki M R 2012 High Capacity Data Embedding Technique Using Improved BPCS Steganography International Journal of Scientific and Research Publications. 2