http://ejournal.seaninstitute.or.id/index.php/InfoSains
Jurnal Info Sains : Informatikan dan Sains, Volume 10, No 02 September 2020
E-ISSN.2797-7889, P-ISSN.2089-3329

Implementation Of Least Significant Bit (LSB) Algorithm For Data Security In Digital Imagery

Penda Sudarto Hasugian[1], Agustina Simangunsong[2]
[1,2]Teknik Informatika, STMIK Pelita Nusantara, Sumatera Utara
Email: pendasudarto01@gmail.com[1] , agustina45@gmail.com[2]

Keywords

LSB
Citra Digital
Security

Abstract. In this study, a software was built that can insert text messages into image files (steganography) and steganalisis or detect the presence or absence of secret messages in an image. Imagery is the most commonly used medium for inserting secret messages, as it can hide messages very well and is widely available. The built software implements steganalisis with the Least Significant Bit (LSB) algorithm which is the most widely used message insertion algorithm. Steganlysis software is built using LSB algorithms to perform insertion and extract insertion messages on image files. The LSB algorithm in this study used the microsoft Visual Basic 6.0 programming language tools. Based on the software tests conducted, it can be seen that the software can run well on insertion and detect the presence or absence of secret messages on image files.

1. INTRODUCTION

Steganography is the science and art of hiding messages in such a way that the existence of messages is not detected by the human senses. The word steganography comes from the Greek word for "hidden writing". Steganography requires twoproperties, namely:container and confidential data to be hidden. Digital steganography uses digital media as container, such as imagery, sound (audio), text, and video. Hidden confidential data can also be imagery, sound (audio), text, or video. If in cryptography, ciphertext remains available, ciphertext steganography can be hidden so that third parties do not know its existence [1].

An image is a two-dimensional image produced from a continuous two-dimensional analog image that has not yet been determined byitsmeasurements. Digital imagery is an image consisting of electromagnetic frequency signals that have been sampling so that it can be determined the size of the image point called pixels. To mathematically declare an image, a function of f(x,y) can be defined where x and y represent a position in two-dimensional coordinates and the f-price at the point (x,y) is the price that indicates the color of the image at that point. [2], [3] A digital image is an image expressed discretely (not continuously), both for its coordinate position and color. Thus, digital imagery can be described as a matrix, where the row index and column index of the matrix represent the position of a point in the image and the price of the matrix element represents the color of the image at that point. In digital imagery that is stated as a matrix arrangement like this, the matrix elements are also called pixels derived from the word picture element (pixel). [4], [5]

The insertion method used in designing this software is the Least Significant Bit (LSB) method. The Least Significant Bit bit is the last bit on a byte (8 bits). [6], [7] This bit is the least meaningful bit, because changes to the bits will not change the rerensation of images in the spatial domain. The basic principle of this method is to replace the last bit of the image pixel with the insertion bits. To insert messages into an image, the insertion image and message must be converted to binary. Image conversion to binary is intended to get one value on one bit of data, where one bit will be replaced with one bit of eight bits of one insertion message character. For example the letter A of the insertion message is changed to binary to "10000011" and one bit in a single byte of image data will be replaced. So if there are eight bits of the message to be hidden, it takes eight bytes from the container image (cover image)[8], [9].

2. METHODS

2.1 Steganography

Steganography is a technique of hiding or disguising the existence of secret messages in a container media so that others are not aware of the presence of messages in the media. The word steganography originally came from the word steganos, steganos itself is actually a Greek word. More details: steganos has the meaning of disguise or concealment and graphein or graptos has the meaning of writing. The definition of steganography that is quite often used in learning with historical methodologies is "writing hidden or veiled writing". [10]

Steganography has been used since time immemorial about 2500 years ago for political, military, diplomatic, and personal purposes as a tool. Conventional steganography techniques attempt to keep communication secret by hiding messages or camouflage messages. So the basic principle in steganography is more concentrated on the confidentiality of communication rather than on the data. [11]

2.2 Media Cover

Cover media is used in steganography as a medium to hold hidden messages. The hidden message may or may not even have any connection at all with the media in which the message is inserted (in the case of confidential communication) or the message may provide important information about the media, such as information otentification, title, date and time of creation, copyright, serial number of the digital camera used to take pictures, information about the content and access to imagery and so on. In theory, all files on a computer can be used as media in steganography, provided that they have bits of redundant data that can be modified. Some examples of cover media used in steganography techniques include:

1. Will
   In steganography algorithms that use text as its insertion medium, text that has been pasted with secret messages should not be suspicious to the person who sees it.
2. Voice
   This format is often chosen because usually files with this format are relatively large. So it can hold a large number of secret messages as well.
3. Image
   This format is most commonly used, because it is one of the most interchangeable file formats in the internet world. Another reason is the large number of steganography algorithms available for container media in the form of images.
4. Video
   This format is a format with a relatively very large file size but rarely used because of its too large size that reduces its practicality and also the lack of algorithms that support this format.

3. RESULTS AND DISCUSSION

In this application, the insertion of text messages into the cover image with the Least Significant Bit (LSB) method that generates the stego image as well as detecting the insertion or extracting to get the insertion message from the stego image.

3.1 Preprocessing Cover Image

Before insertion,the pixel value reading and calculation on the cover image is first performed. For example, a color image with a 200 x 300 pixel color image as shown in Figure 4

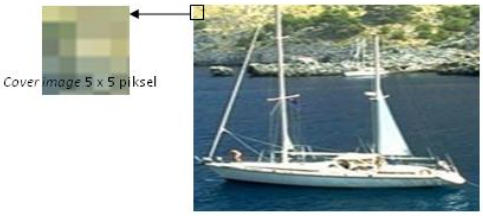


Figure 1. Cover Image (200 x 300 pixels)

The image in Amber1 above is calculated by dividing the image inpixels. For example, a 5 x 5 pixel snapshot of the image comes from the visible cover image as shown in Figure 2.

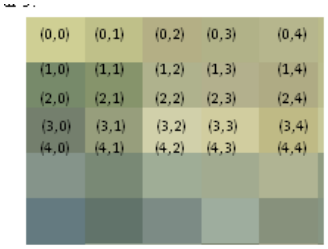


Figure 2 Cover Image (5 x 5 pixels)

citra in Figure 2 above is done reading the pixel value on the Cover Image bitmap data (5 x 5 pixels) as in Figure 3.

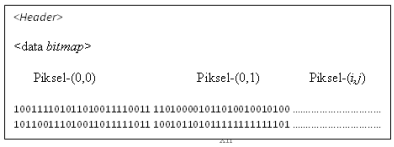


Figure 3 : Cover Image Pixel Value (5 x 5 pixels)

To get each value R, G and B performed modulo operations (rest of the divide) with the following formula:

Value R = C(i,j) and 255…… ..... …….. (3.1)
Value G = (C(i,j) and 65280) / 256…… ..... ……. (3.2)
Value B = (C(i,j) and 16711680) / 256/256 …… ..... … ..... ……... .. (3.3)
Where C(i,j) is the pixel value of the image in the coordinates (i,j) inbinary.
In Figure 3. The above 2 have pixel values including:

1. Value pixels (0,0)= 100111101011010011110011.
   Value R = 100111101011010011110011 and 11111111 = 11110011 = 243 (dec)
   Value G = (100111101011010011110011 and 1111111100000000)/ 100000000
   10110100 - 180 (fromc).
   Value B = (100111101011010011110011 and 111111110000000000000000)/
   100000000/10000000000 - 10011110 - 158 (fromc).
2. Value pixels (0,1) = 110100001011010010010100.
   Value R = 110100001011010010010100 and 11111111 = 10010100 = 148 (dec)
   Value G = (110100001011010010010100 and 1111111100000000)/ 100000000
   10110100 - 180 (fromc).
   Value B = (110100001011010010010100 and 111111110000000000000000)/
   100000000/10000000000 - 110100000 - 208 (fromc).

3.2 Preprocessing Embed Image

Before the text message is inserted into the image file, the data processing of the insertory message into binary is carried out. For example the insertion message is a RIKA string in binary is as follows:

R= 01010010 I= 01001001 K= 01001011 A= 01000001

3.3 Initial Password

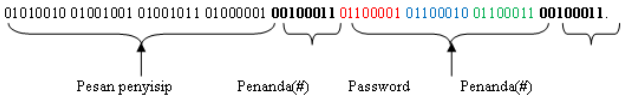
Initial password is the process of converting a password into Binary Value as the key to performing the insertion extraction. Initial passwords are done by adding bit markers at the beginning and end of the password bits that generate the password vector. For example the password is the string "abc" in binary is as follows:

a = 01100001
b = 01100010
c = 01100011
# as bit marker = 00100011
The password representation in the vector is:

3.4 Insertion Least Significant Bit

Insertion Least Significant Bit (LSB) is an algorithm that pastes information inside a cover image that generates a stego image file. In each byte of a file there are the least meaningful bits (LSB). For example at 0001100 bytes**1**, then theLSB bit is **1.** To insert a message, the most suitable bit to replace with the message bit is the LSB bit, because changing the bit will only change the Byte Value to one higher or one lower.

Rika binary messages in binary are represented in the "RIKA" message vector as follows:

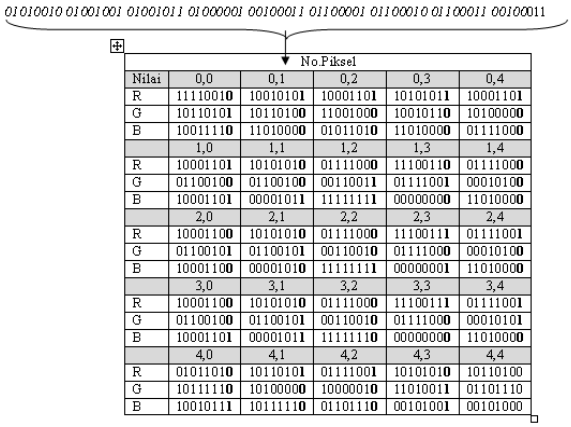


information:

1. The 4 bytes in front of the message vector is the RIKA message binary value.
2. The fifth byte is 1 byte with bold text being the marker bits (#).
3. The sixth to eighth bytes of 3 bytes in red, blue and green are bits of the password.
4. The ninth byte of 1 byte is the marker bits (#).

Next the message vector is merged with the password vector into the embed vector and then the embed vector is inserted by replacing the LSB bit of the binary matrix cover image as in Figure 4.

Vector embed:



Nilai	0,0	0,1	0,2	0,3	0,4
R	11110010	10010101	10001101	10101011	10001101
G	10110101	10110100	11001000	10010110	10100000
B	10011110	11010000	01011010	11010000	01111000
	1,0	1,1	1,2	1,3	1,4
R	10001101	10101010	01111000	11100110	01111000
G	01100100	01100100	00110011	01111001	00010100
B	10001101	00001011	11111111	00000000	11010000
	2,0	2,1	2,2	2,3	2,4
R	10001100	10101010	01111000	11100111	01111001
G	01100101	01100101	00110010	01111000	00010100
B	10001100	00001010	11111111	00000001	11010000
	3,0	3,1	3,2	3,3	3,4
R	10001100	10101010	01111000	11100111	01111001
G	01100100	01100101	00110010	01111000	00010101
B	10001101	00001011	11111110	00000000	11010000
	4,0	4,1	4,2	4,3	4,4
R	01011010	10110101	01111001	10101010	10110100
G	10111110	10100000	10000010	11010011	01101110
B	10010111	10111110	01101110	00101001	00101000

Figure 4. Stego Image Binary Matrix

From the insertion of the embed vector above obtained 9 bytes or 9 x 8 bit = 72 bit insertion, then the cover image needed as much as 72 bytes. Rgb image insertion result or stego image obtained is as in Figure 5

Nilai pixel	Nomor Piksel				
	(0,0) R=242 G=181 B=158	**(0,1)** *R=149 G=180 B=208*	**(0,2)** *R=141 G=200 B=90*	**(0,3)** *R=171 G=150 B=208*	**(0,4)** R=141 G=160 *B=120*
	(1,0) R=140 G=100 B=140	**(1,1)** R=170 G=100 B=10	**(1,2)** R=120 G=50 *B=255*	**(1,3)** R=230 G=120 B=0	**(1,4)** R=250 G=30 B=150
	(2,0) R=30 G=110 B=0	**(2,1)** R=60 G=130 B=100	**(2,2)** R=190 G=120 B=70	**(2,3)** R=100 G=40 B=200	**(2,4)** R=150 G=20 B=50
	(3,0) R=50 G=20 B=240	**(3,1)** R=100 G=110 B=0	**(3,2)** R=70 G=160 B=0	**(3,3)** R=130 G=110 B=130	**(3,4)** R=100 G=30 B=10
	(4,0) R=90 G=190 B=151	**(4,1)** R=181 G=160 B=190	**(4,2)** R=121 G=130 B=110	**(4,3)** R=170 G=210 B=40	**(4,4)** R=180 G=110 B=40

Figure 5. Stego Image RGB Matrix

The result of insertion in the form of stego image with RGB Value is almost the same as the Value RGB cover image which means there is no color difference when viewed with the human sense of vision because it only adds or decreases the Value of 1 bit of its LSB.

3.5 Extraction Least Significant Bit

To perform the extraction of the insertion with the LSB algorithm, the LSB bit reading of each stego image pixel is performed and represented every 8 bits of conversion to grayscale Value

Pixel R (0.0) = 11110010 LSB = 0
Pixel G (0.0) = 10110101 LSB = 1
Pixel B (0.0) = 10011110 LSB = 0
Pixel R (0.1) = 10010101 LSB = 1
Pixel G (0.1) = 10110100 LSB = 0
Pixel B (0.1) = 11010000 LSB = 0
Pixel R (0.2) = 10001101 LSB = 1
Pixel G (0.2) = 11001000 LSB = 0
Pixel B (0.2) = 01011010 LSB = 0
Pixel R (0.3) = 10101011 LSB = 1
Pixel G (0.3) = 10010110 LSB = 0
Pixel B (0.3) = 11010000 LSB = 0
Pixel R (0.4) = 10001101 LSB = 1
Pixel G (0.4) = 10100000 LSB = 0
Pixel B (0.4) = 01111000 LSB = 0
Pixel R (1.0) = 10001101 LSB = 1
Pixel G (1.0) = 01100100 LSB = 0
Pixel B (1.0) = 10001101 LSB = 1
Pixel R (1.1) = 10101010 LSB = 0
Pixel G (1.1) = 01100100 LSB = 0
Pixel B (1.1) = 00001011 LSB = 1
Pixel R (1.2) = 01111000 LSB = 0
Pixel G (1.2) = 00110011 LSB = 1
Pixel B (1.2) = 11111111 LSB = 1
Pixel R (1.3) = 11100110 LSB = 0
Pixel G (1.3) = 01111001 LSB = 1
Pixel B (1.3) = 00000000 LSB = 0
Pixel R (1.4) = 01111000 LSB = 0
Pixel G (1.4) = 00010100 LSB = 0
Pixel B (1.4) = 11010000 LSB = 0
Pixel R (2.0) = 10001100 LSB = 0
Pixel G (2.0) = 01100101 LSB = 1
Pixel B (2.0) = 10001100 LSB = 0
Pixel R (2.1) = 01100101 LSB = 1
Pixel G (2.1) = 10101010 LSB = 0
Pixel B (2.1) = 00001010 LSB = 0
Pixel R (2.2) = 01111000 LSB = 0
Pixel G (2.2) = 00110010 LSB = 0
Pixel B (2.2) = 11111111 LSB = 1
Pixel R (2.3) = 11100111 LSB = 1
Pixel G (2.3) = 01111000 LSB = 1
Pixel B (2.3) = 00000001 LSB = 1
Pixel R (2.4) = 01111001 LSB = 1
Pixel G (2.4) = 00010100 LSB = 0
Pixel B (2.4) = 11010000 LSB = 0
Pixel R (3.0) = 10001100 LSB = 0
Pixel G (3.0) = 01100100 LSB = 0
Pixel B (3.0) = 10001101 LSB = 1
Pixel R (3.1) = 10101010 LSB = 0
Pixel G (3.1) = 01100101 LSB = 1
Pixel B (3.1) = 00001011 LSB = 1
Pixel R (3.2) = 01111000 LSB = 0
Pixel G (3.2) = 00110010 LSB = 0
Pixel B (3.2) = 11111110 LSB = 0
Pixel R (3.3) = 11100111 LSB = 1
Pixel G (3.3) = 01111000 LSB = 0
Pixel B (3.3) = 00000000 LSB = 0
Pixel R (3.4) = 01111001 LSB = 1
Pixel G (3.4) = 00010101 LSB = 1
Pixel B (3.4) = 11010000 LSB = 0
Pixel R (4.0) = 01011010 LSB = 0
Pixel G (4.0) = 10111110 LSB = 0

Pixel B (4.0) = 10010111 LSB = 1
Pixel R (4.1) = 10110101 LSB = 1
Pixel G (4.1) = 10100000 LSB = 0
Pixel B (4.1) = 10111110 LSB = 0
Pixel R (4.2) = 01111001 LSB = 1
Pixel G (4.2) = 10000010 LSB = 0
Pixel B (4.2) = 01101110 LSB = 0
Pixel R (4.3) = 10101010 LSB = 0
Pixel G (4.3) = 11010011 LSB = 1
Pixel B (4.3) = 00101001 LSB = 1
Pixel R (4.4) = 10110100LSB = 0
Pixel G (4.4) = 01101110LSB = 0
Pixel B (4.4) = 00101000LSB = 0

From the LSB bit above then in the insertion binary is reconstructed into 8 bits as follows:

01010010 = 082 = R
01001001 = 073 = Message (embed)
01001011 = 075 = K
01000001 = 065 = A
00100011 = 35 = #Penanda
01100001 = 97 = a
01100010 = 98 = bpassword
01100011 = 99 = c
00100011 = 35 = #Penanda

From the reconstruction of LSB bits obtained are:
Message = RIKA
Marker bit = #
Password = abc

4. CONCLUSION

After designing and applying image steganography software in Message Security with The Least Significant Bit Algorithm (LSB),the system test results are as follows Can insert messages into digital image files, Can perform message extraction from digital imagery

REFERENCE

[1] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, "A novel image steganography method via deep convolutional generative adversarial networks," *IEEE Access*, vol. 6, 2018, doi: 10.1109/ACCESS.2018.2852771.

[2] D. Novianto, "IMPLEMENTASI KEAMANAN BERKAS MENGGUNAKAN TEKNIK STEGANOGRAFI DAN ALGORITMA KRIPTOGRAFI MENGGUNAKAN METODE LEAST SIGNIFICANT BIT (LSB) DAN ALGORITMA RIVEST CODE 4 (RC4)," *JUTIM (Jurnal Tek. Inform. Musirawas)*, vol. 3, no. 2, 2018, doi: 10.32767/jutim.v3i2.340.

[3] O. Soleh, F. Alfiah, and B. Yusuf, "Perancangan Aplikasi Steganografi Dengan Teknik LSB dan AlgoritmaRC4 & Base64 Encoding," *Technomedia J.*, vol. 3, no. 1, 2018, doi: 10.33050/tmj.v3i1.493.

[4] M. Hussain, A. W. A. Wahab, Y. I. Bin Idris, A. T. S. Ho, and K. H. Jung, "Image steganography in spatial domain: A survey," *Signal Process. Image Commun.*, vol. 65, 2018, doi: 10.1016/j.image.2018.03.012.

[5] S. S. Baawi, M. R. Mokhtar, and R. Sulaiman, "A comparative study on the advancement of text steganography techniques in digital media," *ARPN J. Eng. Appl. Sci.*, vol. 13, no. 5, 2018.

[6] E. A. Abbood, R. M. Neamah, and S. Abdulkadhm, "Text in image hiding using developed LSB and random method," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 4, 2018, doi: 10.11591/ijece.v8i4.pp2091-2097.

[7] I. Gunawan, Sumarno, E. Irawan, and H. S. Tambunan, "Pengamanan Berkas Dokumen Menggunakan Fungsi Algoritma Steganografi LSB," *Algoritm. J. Ilmu Komput. dan Inform.*, vol. 02, no. 01, 2018.

[8] M. Douglas, K. Bailey, M. Leeney, and K. Curran, "An overview of steganography techniques applied to the protection of biometric data," *Multimed. Tools Appl.*, vol. 77, no. 13, 2018, doi: 10.1007/s11042-017-5308-3.

[9] P. Wu, Y. Yang, and X. Li, "StegNet: Mega Image steganography capacity with deep convolutional network," *Futur. Internet*, vol. 10, no. 6, 2018, doi: 10.3390/FI10060054.

[10] M. Cem kasapbaşi and W. Elmasry, "New LSB-based colour image steganography method to enhance the efficiency in payload capacity, security and integrity check," *Sadhana - Acad. Proc. Eng. Sci.*, vol. 43, no. 5, 2018, doi: 10.1007/s12046-018-0848-4.
[11] Z. Fachrina, H. A. Humansyah, I. Fitri, and A. Rubhasy, "No Title," vol. 5, no. 2, doi: 10.35870/jtik.v5i2.203.