2017 1st International Conference on Informatics and Computational Sciences (ICICoS)

Random Pixel Embedding for Hiding Secret Text over Video File

Muhammad Khaerul Anam, Eko Adi Sarwoko, Edy Suharto,Kharis Khasburrahman
Department of Informatics
Diponegoro University
Semarang, Indonesia,
KhaerulAnam@gmail.com,eko.adi.sarwoko@gmail.com,edys@undip.ac.id,khasburrahman@yahoo.com

***Abstract*—Data hiding can be done using steganography technique. The secret data is embedded in a carrier file such as video in such an invisible manner. Video format is chosen due to its capacity to store large data within frames. Moreover, because these frames are displayed rapidly, any tiny modification as a result of data hiding process is hardly observed by human eye. This study employed a Random Number Generator function into a method called Random Pixel Embedding. This was implemented by developingan application to hide text data in a video file as container. Quality testingfor insertion phase result was done by measuring Signal to Noise Ratio which yielded a value of 99%. Meanwhile, testing for extraction phase result was done by measuring Character Error Rate which yielded value of 0.06. Those tests proved that this research produced a steganography application for hiding text message over video file which isready to use.**

Keywords—Steganography, Video, Random Pixel Embedding

I. INTRODUCTION

Data communication over the Internet requires high-level security. A number of techniques have been proposed in order to assure data security. One of them is steganography, where the secret data is hidden within a cover file, being unnoticeable as a carrier. The mechanism of data hiding in steganography can be used to deceive eves-droppers so that the secret data remains secure.

Nowadays, steganography is developed using image and video as cover file. Data hiding in an image file can apply various methods. Study in [1] uses grey-scale image steganography applying Pattern Pixel Difference. Steganography can be combined with some cryptography methods such as RC4 Stream Cipher as studied in [2]. Implementing steganography in video file can use methods applicable in image file as studied in [3], where encryption is based-on Least-Significant Bit (LSB), or in [4], where symmetric XOR encryption key with embedding RGBBGRRG is used. In order to leverage capacity, embedding data can use Histogram Constant Value method using 4 last bits as studied in [5].

LSB method is widely implemented in steganography due to its simplicity. This method takes advantage on human-perception limit, which is unable to detect tiny difference between two objects observed. In video file, its frames appear subsequently so fast that can be used to hide data without being recognized. LSB method works by substituting least-significant bits of bytes in frames with bits of secret data. The location of pixel is determined in sequence. Unfortunately, this substitution tends to be vulnerable since steganalyst can extract these least-significant bits. Besides, the extracted data becomes insecure.

In this study, randomize the locations of data hiding is applied to improve data security in steganography. The aims of this proposed method is to make steganalysis difficult to locate the true pixels carrying the bits of hidden data. The experiment works are tested objectively by measuring the Signal to Noise Ratio and testing using Character Error Rate.

II. LITERATURE REVIEW

A. Steganography

Steganography is a practice of hiding secret messages into certain media so that the secret message cannot be easily known to the parties other than the sender and receiver. Steganography requires a container media and the secret messages [6]. Steganography differs from cryptography, the difference being that it lies in the concealment of data and the end result of both techniques. Cryptography renders the process of randomness to the original data resulting in a new data that is different from the original data that potentially lead to suspicion of others, while steganography inserts data to the media container to produce a media containing original data and secret messages that potentially reduce suspicion to the other parties. The steganography mechanism can be seen in Fig. 1.

With
FE : Insertion emb to cover
FE – 1 : Extraction emb from cover
Cover : Container
Cover* : Container after extraction
Key : Key that is used
Emb : Secret messages
Emb* : Secret messages obtained
Stego : Stego object

Fig. 1. Steganography Mechanism

B. MP4 Video Format

Video is a composite of many digital images. Digital images are shown in sequence with a certain time period so that the image appears to move. To be able to process the video, color data on a number of frames that exist in the video is required. Because the frame is a digital image, video processing cannot be separated from digital image processing [7]. One of the popular video format today is MP4. In general, MP4 video format files are composed of blocks of blocks. In each block there are pieces of information, such as resolution, bit stream, and group of pictures. The group of pictures is composed of information about the image in which the image can be one or more. Those images can be used as a media container for the process of steganography.

C. Pseudo-Random Number Generator Function

Pseudo-Random Number Generator is using sequential mathematical functions, each new number is a deterministic function of the previous numbers (recurrent). The basis of the recursive function is called seed, which is an initial number. There is no mathematical function that can produce a perfect or natural sequence of random numbers. Therefore this generator is called Pseudo-Random Number Generator (PRNG).

PRNG is a generator of semi-random numbers, in which the generation of each element depends on the mathematical formulation and the seeds used. Its function is like in the following equation (1).

$$X_n = (aX_{n-1} + b)\ mod\ m \quad (1)$$

With
X_n : Random number-n in its sequence
a, b : Coefficient parameter
m : Modulus

The basis of this function is the X_0 called the seed. PNRG excellence lies in its speed and it requires only a few mathematical operations [8].

D. Random Pixel Embedding

The RPE method is an insertion method which is the development of the LSB method combined with the PRNG function as a determinant of the order of the pixel dots of a secret message insert [9]. LSB insertion is a common and simple approach to insert information on a file. In this method, the least significant bit (the far right) bit is converted to an inserted message stream bit. This technique works well on image steganography. In the human eye, the stego of the object will look identical to the container media that has not been inserted. The greater the quality of the image resolution the better the concealment of messages in it. This is due to the considerable space and the fewer bits that need to be changed due to the insertion process.

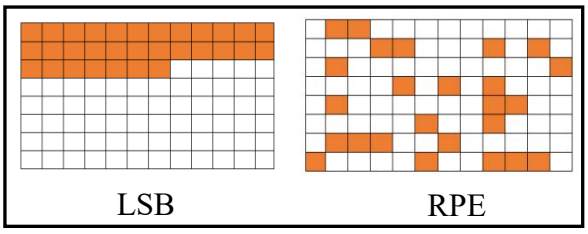

Fig. 2 Modified Bytes using LSB compared to RPE

When using 24-bit images, each image can store on different color components on the Red, Green, and Blue color channels. While the weakness of this method, the other party can easily get a secret message because the LSB method is to change the pixels at the end of the file. In order to strengthen the data hiding technique, a development method called RPE was introduced. This method makes the secret data bits not be used to replace the byte in sequence, but selected byte random arrangement. Random numbers are generated with PRNG functions. The purpose of the PRNG function is to generate the same set of random numbers for each of the same keys. Therefore, the sender and receiver must have the same key (symmetric). Then, the insertion step of the text to the image is performed on each frame in the video in the order of bytes corresponding to that generated by PNRG. Differences in LSB and RPE insertion results can be seen in the Fig. 2.

E. Message Insertion Procedure

The procedure used in message insertion using the RPE method is similar to the message insertion procedure in the LSB. The message insertion procedure can be explained in the following equation (2).

$$I_s(i,j) = \begin{cases} I(i,j) - 1\ , & LSB\big(I(i,j)\big) = 1\ and\ m = 0 \\ I(i,j)\ , & LSB\big(I(i,j)\big) = m \\ I(i,j) + 1\ , & LSB\big(I(i,j)\big) \neq 0\ and\ m = 1 \end{cases} \quad (2)$$

With
$I_s(i, j)$: Stego Object
I(i, j) : Container
LSB (I(i, j)) : Least Significant Bit
m : Secret Messages Bit

F. Message Extraction

The procedure used in message extraction using the RPE method is very simple. This procedure reads only the rightmost bit value of each object stego pixel (LSB (i, j)), according to the sequence generated by the PRNG function. The result of this process is a sequence of bits which is then converted into text message characters.

G. Signal to Noise Ratio

Signal to Noise Ratio (SNR) is used to measure the noise level between the container media files and the stego video files of the object, so that the error value is known. The higher the SNR percentage value, the quality of the video file is said to be good because the ratio of signal to noise is higher. By calculation, SNR can be formulated by the following equation (3).

$$SNR = \frac{FW - BT}{FW} * 100\% \quad (3)$$

With
FW : Sum of bit in video file
BT : Sum of changed bit in video

H. Character Error Rate

The Character Error Rate (CER) corresponds to the inserted data in the form of text, then a CER calculation is required which reflects how many characters are corrupted by application processing. The CER formula corresponds to the following equation (4).

$$CER = \frac{sum\ of character\ error}{sum\ of\ total\ character} \quad (4)$$

I. Message Extraction Procedure

The procedure used in message extraction using the RPE method is very simple. This procedure reads only the rightmost bit value of each object stego pixel (LSB (i, j)), according to the sequence generated by the PRNG function. The result of this process is a sequence of bits which is then converted into text message characters.

III. DISCUSSION

The process of inserting text messages on video media using the RPE method has the various stages described in the insertion process flow diagram. The process can be seen in the following pseudo code of Algorithm 1.

Algorithm 1: Message Insertion

input : *stego, messages, key*
output : *stego_video*

```
1  video ← read()
2  imframe ← video_to_imframe(stego)
3  messages ←read()
4  ascii ←text_to_ascii(messages)
5  binary ←ascii_to_binary(ascii)
6  key ← read()
7  random_number ← generate_random()
8  foreachimframe
9      frame ← insertion()
10 stego_video ← to_video(frame)
11 output(stego_video)
```

The explanationsare as follows: firstly, make sure the media file container contained with MP4 format, which will be inserted an array text message with a maximum length of 255 characters. The message becomes ASCII code.

The second stage of the sender enters the secret message key in the form of a number, which is used as a seed value to perform the PRNG function.

Furthermore, the application will generate random numbers based on the secret message key with PRNG function, will be done repeatedly until it produces the sequence of numbers as much as the length of the text message characters.

The fourth stage of the application inserts each character bit of the message into each video pixel bit by using the LSB method. The insertion is performed on all three RGB channels. Text messages are made on channel R on bits 1,2,3; on channel G at bit 4,5,6; and channel B on bits 7 and 8 so as to obtain its value into RGB.

The insertion process is repeated up to as many characters as the message and is repeated all the way to the entire frame of the container media video. Then the last phase of the app will rearrange it into a whole video called the stego object.

Pseudo code for extraction process is as following Algorithm 2.

Algorithm 2: Message Extraction

input : *stego*, *key*
output : *text*

```
1  stego← read()
2  imframe ← video_to_imframe(stego)
3  key←read()
4  random ← generate_random_number()
5  for eachimframe
6  binary ← extract()
7  output(binary)
8  ascii ← binary_to_ascii(binary)
9  text ← ascii_to_text(ascii)
10 output(text)
```

A. Process Experiment

The objective of the experiment is to measure the similarity of extracted message with its original text based on CER value. Various combination of text including character type and text length were used. The result of the experiment can be seen in TABLE I.

Based on Table I, it is known that the process of extracting secret messages in the form of text can be done from the stego of a video object. The extraction results in a secret text message that is not much different from the original secret text message. In the above experiments, the CER value = 0 in the second, third, seventh, eighth, tenth, eleventh, fourteenth, and fifteenth experiments. The value of CER = 0 means that the process of insertion and extraction is done perfectly without savoring the contents of the secret text message. In the first, fourth, fifth, sixth, ninth, twelfth and thirteenth experiments only one character was damaged.

B. Quality Experiment

The objective of this experiment is to measure the quality of video files after being inserted hidden text based on SNR value. The result of the experiment can be seen in TABLE II.

Based on table II, it is known that the insertion and extraction process does not change many bits of arrangement in the video media. Seen from the high SNR value, it can be said that the use of RPE method is very suitable for media container in the form of video.

TABLE I. PROCESS EXPERIMENT

No	Object	Key	Original Message	Extracted Message	CER
1	Stego Park.mp4	21	IniPesan RAHASIA	IniPesan AHASIA	0.0588
2	Stego Glass.mp4	100	InformatikaUndip 2012	InformatikaUndip 2012	0
3	Stego Flag.mp4	123	1234567890987654321	1234567890987654321	0
4	Stego Baby.mp4	777	ABC ~!@#$%^&*(xyz	A(C ~!@#$%^&*(xyz	0.0556
5	Stego Ski.mp4	2401031214 0028	AplikasiSteganografiPesanTeksp ada Media Video denganMenggunakanMetode Random Pixel Embedding	AelikasiSteganografiPesanTe kspada Media Video denganMenggunakanMetode Random Pixel Embedding	0.0102
6	Stego Paper.mp4	1024	Abcdefg_ijklmnopqrstuvwxyz	Abcdefghijklmopqrstuvwxyz	0.0417
7	Party.mp4	2020	PestaUlangTahun ke-20	PestaUlangTahun ke-20	0
8	Meeting.mp4	2013	HMIF Undip 2013	HMIF Undip 2013	0
9	Stego Band.mp4	9999	><?/:;”[]{}	><?/:;[]{}	0.0651
10	Stego Concert.mp4	777	Test123 Test123	Test123 Test123	0
11	Stego Jump.mp4	9092014	9 September 2014	9 September 2014	0
12	Stego Air Terjun.mp4	987	0987654321	987654321	0.0486
13	Stego Susnset.mp4	555	Sunset SUNSETsunsetsUnSeT	Sunset SNSET sunset sUnSeT	0.0509
14	Stego Restoran.mp4	1043	1043 KB	1043 KB	0
15	Stego Puncak.mp4	10	RPE & LSB	RPE & LSB	0

TABLE II.QUALITY TESTING

No	Object	Bit Per Frame		SNR
1	Stego Park.mp4	Total : 91.203 bit;	Difference : 45 bit	99.993 %
2	Stego Glass.mp4	Total : 121.615 bit;	Difference : 57 bit	99.993 %
3	Stego Flag.mp4	Total : 221.763 bit;	Difference : 53 bit	99.994 %
4	Stego Baby.mp4	Total : 101.643 bit;	Difference : 39 bit	99.995 %
5	Stego Ski.mp4	Total : 181.020 bit;	Difference : 238 bit	99.974 %
6	Stego Paper.mp4	Total : 160.034 bit;	Difference : 138 bit	99.953 %
7	Party.mp4	Total : 345.110 bit;	Difference : 41 bit	99.995 %
8	Meeting.mp4	Total : 1781.002 bit;	Difference : 98 bit	99.983 %
9	Stego Band.mp4	Total : 200.603 bit;	Difference : 33 bit	99.997 %
10	Stego Concert.mp4	Total : 218.400 bit;	Difference : 105 bit	99.989 %
11	Stego Jump.mp4	Total : 54.040 bit;	Difference : 25 bit	99.997 %
12	Stego Air Terjun.mp4	Total : 82.301 bit;	Difference : 87 bit	99.973 %
13	Stego Susnset.mp4	Total : 240.008 bit;	Difference : 145 bit	99.951 %
14	Stego Restoran.mp4	Total : 181.110 bit ;	Difference : 97 bit	99.985 %
15	Stego Puncak.mp4	Total : 38.900 bit;	Difference : 102 bit	99.981 %

IV. CONCLUSION

We proposed a steganography using Least Significant Bit modified by utilizing the Random Number Generator function, into a method called Random Pixel Embedding. Based on experimental results shows that the video quality of the stego object has no visible difference with high SNR value 99%due to insertion process in LSB. Moreover, the similarity of secret messages extracted back to this results in a CER value of 0.06 with an average message being corrupted for only 1 character at each insertion. In future works, various types of video formats will be applied with several secret messages formats such as pictures or sound.

REFERENCES

[1] D. Lerch-Hostalot and D. Megias, "LSB matching steganalysis based on patterns of pixel differences and random embedding," Computers andSecurity, vol. 32, pp. 192–206, 2013.

[2] O.Bardhan, A. Bhattacharya, B. P. Sinha, "A Steganographic Technique Based on VLSB Method using RC4 Stream Cipher", International Conference on Advances in Computing, Communications and informatics, 2014.

[3] P. Yadav, N. Mishra, and S. Sharma, "A secure video steganography with encryption based on LSB technique," IEEE International Conference on Computational Intelligence and Computing Research, 2013.

[4] M. Ramalingam and N. A. M. Isa, "A steganography approach for sequential data encoding and decoding in video images," International Conference on Computer, Control, Informatics and Its Applications, 2014.

[5] H. M. Kelash, O. F. Abdel Wahab, O. A. Elshakankiry, and H. S. El-Sayed, "Hiding data in video sequences using steganography algorithms," Int. Conf. ICT Converg., pp. 353–358, 2013.

[6] M. M. Emam, A. A. Aly, F. A. Omara, "An Improved Image Steganography Method Based on LSB Technique with Random Pixel Selection", International Journal of Advanced Computer Science and Applications, Vol. 7, No. 3, 2016.

[7] S.A. Abbas, T. I. B. El Arif, F. F. M. Ghaleb, S.M. Khamis, "Optimized Video Steganography Using Cuckoo Search Algorithm", IEEE Seventh International Conference on Intelligent Computing and Information Systems, pp.572 – 577. 2015.

[8] Swetha V, Prajith V, Kshema V, "Data Hiding Using Video Steganography - A Survey", IJCSET, Vol 5, Issue 6, pp. 206-213, June 2015.

[9] M.Ramalingam, N. A. M. Isa, "A Steganography Approach over Video Images to Improve Security", Indian Journal of Science and Technology, Vol 8 Issue 1, pp. 79 – 86, January 2015.