2013 International Conference on Advances in ICT for Emerging Regions (ICTer): 056 – 059

Video Steganography

A. Munasinghe[#1], Anuja Dharmaratne[#2], Kasun De Zoysa[#3]

[#]*University of Colombo School of Computing*
Colombo, Sri Lanka

[1]avinesh.ucsc@gmail.com

[2]atd@ucsc.lk

[3]kasun@ucsc.lk

Abstract— **With the development of the technology, people have tend to figure out methods which are not only capable in hiding a message, but also capable of hiding the existence of a message. Steganography was introduced as a result of such research work. The current study is conducted in order to hide a video in a visual file. We suggest changing the LSB (Least significant Bit) of each byte of the carrier file. As this method does not add any new data but only change the LSB, this method does not increase the size of the carrier file unusually. Thus, the existence of the message cannot be detected. To improve the better performance, cryptographic techniques are also used and implemented in this research. The system was evaluated by checking the ability of hiding the existence of a message and the ability of retrieving the message correctly. Results show that the system has addressed its research objectives . The limitation for this research is that the library avifill32.dll can only be used for uncompressed AVI files.**

Keywords— **Steganography, LSB, Cryptography, Encryption**

I. Introduction

Steganography is the art of sending hidden messages in a particular way that no one can apart from the sender and the receiver suspects the existence of a message. The word steganography literally means covered writing as derived from Greek [6]. The goal of cryptography is to make data unreadable by a third party whereas, the goal of steganography is to hide the existence of the data from a third party. Steganography works by replacing bits of useless or unused data in regular computer files (such as graphics, sound, text, html, or even floppy disks) with bits of different or invisible information.

According to Bhaumik et. al [2], a proper data hiding method should contain several requirements such as Imperceptibility, Robustness, Capacity and Security. An application known as watermarking refers to embedding a mark into an object which can be used to identify the object [7]. The requirements for data hiding differ from those of watermarking [4]. For example, while transparent or visible watermarks are acceptable in many cases, hidden data for control or secure communication need to be perceptually invisible.

The general process of Steganography is that a data message is hidden (embedded) within a cover signal. The output of the embedder is called a stego signal. After transmission, recording and other signal processing which may contaminate and distort the stego signal, the embedded message is retrieved using the appropriate stego key in the block called extractor [7]. The carrier of steganography can be an image, text, audio or a video file [8]. Most of the steganography systems are developed in order to embed a text file, image or an audio file in a carrier file. Only a few algorithms are developed to embed a video file in a video file. This research is mainly carried out in order to embed a video in a video.

978-1-4799-1276-6/13/$31.00 ©2013 IEEE

The existing methods have several issues. The GOP method (group of picture), increases the size of the embedded video unusually. Thus, it is easy to detect the existence of a hidden message. The constraints of embedding in DCT domain are that many of the 64 coefficients are equal to zero and changing too many zeros to non-zero values will have an effect on the compression rate [1]. The aim of this research is to embed a visual message file to a visual carrier file where the difference of sizes of carrier file and the message file is not much substantial.

II. Methodology and Design

Fig. 1 shows the high level design diagram of the system. The avifil32.dll functions are used in order to process the video file. In this approach, C# wrappers are used for access functions available in avifil32.dll. In order to create the system, 4 modules are used:

- Video Frame splitting & Generating Factory
- Byte code Factory
- Cryptographic Factory
- Steganographic Factory

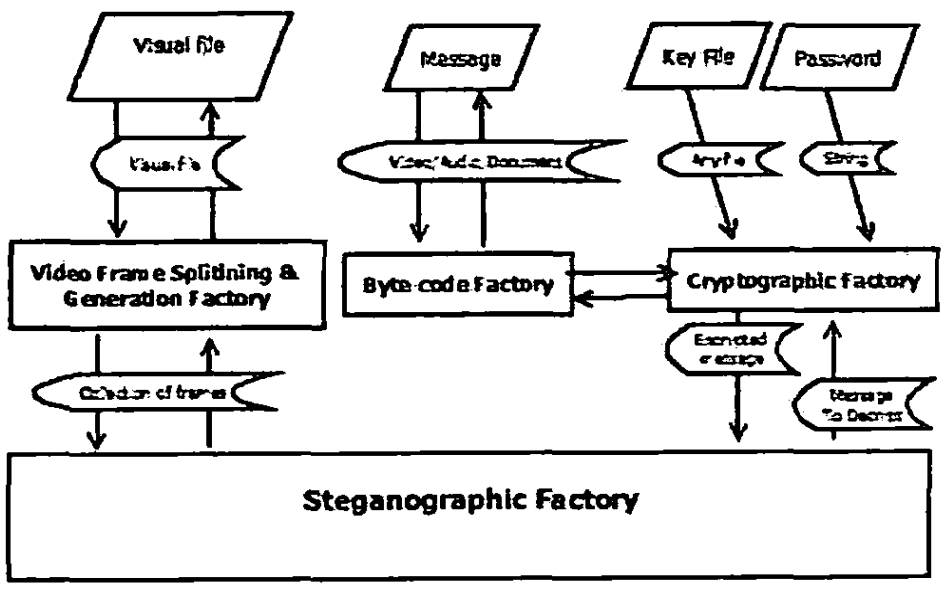

Fig. 1: High Level Design diagram

There are 4 types of data required in order to carry out the research: the key file, password, carrier and the message. The carrier file should be larger than the message with the proportion depending on the method used.

- Password: A password was used in order to encrypt the message before embedding it into the carrier file.
- Key file: A key file is used to increase the security of the hidden message. Individual passwords can be guessed in various ways. The key file prevents guessing the password and therefore, it increases the security of the method.
- Carrier file: A video file is used as the carrier file in this research. The format of the video file is AVI.

- The message: The message will be a video file which is smaller than the carrier file.

In data hiding process, the visual file is the carrier file. Using video frame splitting & generation factory, the frames are generated for the carrier file. The message is translated into a byte code by the byte code factory. Then the byte code of the message encrypts using cryptography factory. There are 2 inputs required for cryptography factory: The key file and the password. This causes to increase the security as the password alone can be generated by guessing. This encrypted message hides in frames of the carrier file in the module, steganographic factory. Finally, the visual recreates using these frames.

Frame Splitting & Generation Factory

In this module, the visual is divided into frames and sent to steganographic factory. Once the message is embedded into the frame, this factory will create the visual using message embedded frames. Using C# wrapper for functions available in avifil32.dll (system32 dll) it can access relevant functions available in that dll to split frames and generate videos.

Byte code Factory

In this module, the message is converted into its byte code and sent to the cryptographic factory. Once you need to extract the message from the video, the byte code of the message, which is provided by the cryptographic factory, will be converted to the message.

Cryptographic Factory

Use of steganography will help to hide the existence of the hidden message. However, by embedding an encrypted message, it causes to produce more secure systems. Thus, encryption is done as a part of this steganography process.

To encrypt a message, many researchers had found that it is more secure to use a key file with a password than using password alone. This will omit the issue of guessing the password. Thus, for the current system, both key file and password are use.

Two XOR gates are used in this encryption method. Fig. 2 shows how the logic gate operates for this encryption. A denotes the key file and B denotes the password. As the password is smaller than the key file, the password will be repeated. C denotes the message.

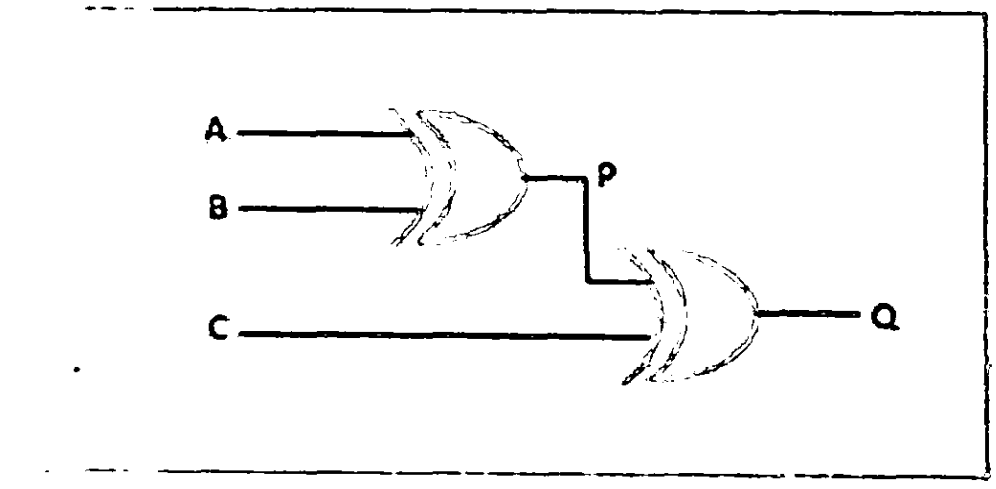

Fig. 2: Logic gate for encryption

Fig. 3 shows how the logic gate operates for decryption. Q denotes the encrypted message. R denotes the decrypted message. From Equation (1) to (7), it shows that the hidden message C is equal to the decrypted message R.

$$P.\bar{C} + C.\bar{P} = \quad (1)$$

$$P.C + \bar{C}.\bar{P} = \quad (2)$$

$$P.\bar{Q} + Q.\bar{P} = \quad (3)$$

By applying equation (1) and equation (2) to equation (3),

$$P.(P.C + \bar{C}.\bar{P}) + (P.\bar{C} + C.\bar{P}).\bar{P} = \quad (4)$$

$$P.C + 0 + 0 + C.\bar{P} = \quad (5)$$

$$C.(P + \bar{P}) = \quad (6)$$

$$C = \quad (7)$$

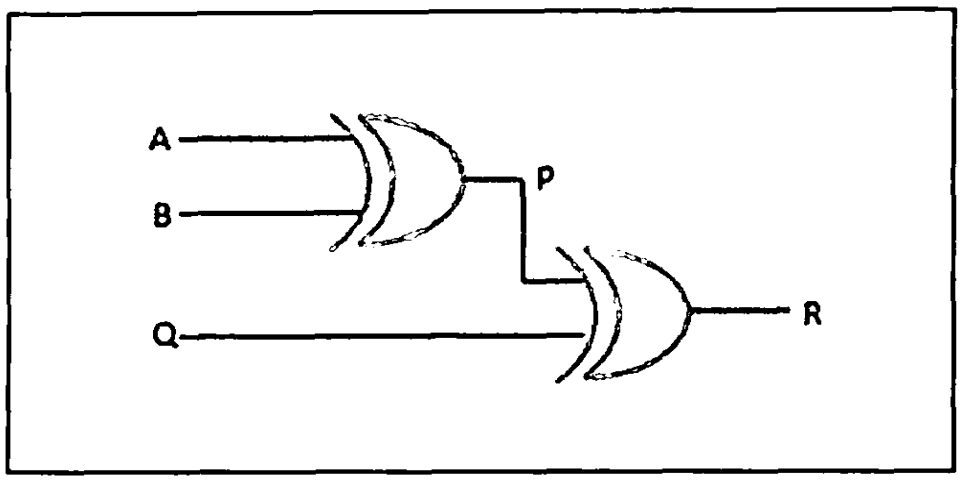

Fig. 3: Logic gate for decryption

D. Steganography Factory

1) Encoder

In this module, the encrypted message will be embedded to the carrier file. The size of the message will depend on the following factors.

- Frame width of the carrier file (P)
- Frame height of the carrier file (Q)
- No of bytes in pixel (R)
- Frame count (S)

Thus, the size of the hidden file can be calculated using equation (8).

$$Max\ Store\ Bytes = \frac{P \times Q \times R}{S} \quad (8)$$

A frame of visual carrier file divides into pixels and a pixel divides into three bytes which contain red, green and blue values. The byte code message embeds into these pixels of the carrier file by changing each least significant bit value. For example assume that the first byte of the message is [10010110]. Assume that the 3 pixels of the frame contains following RGB values as [10011001, 01101011, 11001011] for 1st pixel, [10010010, 11101011, 10001010] for 2nd pixel and [10110110, 01101110, 10010111] for 3rd pixel. The message was embedded by changing the LSB value as [10011001, 01101010, 11001010] in first pixel [10010011, 11101010, 10001011] in second pixel and [10110111, 01101110, 10010111] in third pixel. Thus two pixels and two bites of the carrier file needs to represent one byte of the message.

III. EVALUATION

The performance of the steganography analysis can be measured in two areas. In a proper steganographic analysis, the main feature is that the system should be able to hide the existence of the hidden message from the man-in-the-middle. The second quality is that the receiver should be able to extract the message properly from the embedded video. This section describes the evaluation of the above mentioned performance.

A. Performance of hiding the existence

This is the main benefit of using steganography instead of cryptography. Thus, it is important to evaluate the performance of hiding the message from the man-in-the-middle. In order to measure the performance of hiding the message, the embedded videos are sent to an investigator to identify the existence of a message. For that, 17 embedded and 17 non embedded video files are used. The investigator is requested to test these videos and classify them as message embedded videos and normal videos. The results can be categorized as follows.

Investigator classifies the video as

- an embedded video and the video was truly embedded (TP)
- an embedded video and the video was not embedded (FP)
- a non embedded video and the video was embedded (FN)
- a non embedded video and the video was truly not embedded (TN)

These categories can be used to check the performance of the suggested method. For a proper steganographic method, the ratio between correct classification and the wrong classification should not be significant. Therefore, Odds ratio [3] can be used to measure the performance.

The odds are a way of representing probability, especially familiar for betting. The odds are the ratio of the probability that the event of interest occurs to the probability that it does not. This is often estimated by the ratio of the number of times that the event of interest occurs to the number of times that it does not [3]. Bland and Altman [3] suggested 3 reasons to use odds ratio when comparing performance.

- Firstly, they provide an estimate (with confidence interval) for the relationship between two binary (yes or no) variables.
- Secondly, they enable us to examine the effects of other variables on that relationship, using logistic regression.
- Thirdly, they have a special and very convenient interpretation in case control studies.

$$odds\ ratio = \frac{TP \times}{FN \cdot} \tag{9}$$

Glas [5] shows that the odds ratio can be calculated using equation (9). The range of the odds ratio is from 0 to infinity. The value 1 means there is no significant discrimination between embedded video and the normal video. Therefore, the 95% confidence interval for the log value of the odds ratio can be calculated using equation (10). The Standard error can be calculated using equation (11). Calculating antilog provides the confidence interval for these values. If the value "1" belongs to the confidence interval, it proves that there is no significant discrimination between the embedded video and the normal video.

$$C.I = \log(OR) \pm 1.96 \times SE(\log O \tag{10}$$

$$SE(\log OR) = \sqrt{\frac{1}{TP} + \frac{1}{TN} + \frac{1}{FP} +} \tag{11}$$

The results of the test are shown in Table I.

TABLE I

CATEGORIZED RESULTS OF THE TEST

		True state of video file	
		Embedded	Not Embedded
Predicted state of the video file	Embedded	8	9
	Not Embedded	9	8

According to the table, the calculated odds ratio (OR) was 0.7901 and the confidence interval for the odds ratio was {0.205, 3.037}. The calculation is given in equations from (12) to (19).

$$odds\ ratio = \frac{\xi}{\varsigma} \tag{12}$$

$$odds\ ratio = 0.79 \tag{13}$$

$$\log Odds\ ratio = -0.2 \tag{14}$$

$$SE(\log OR) = \sqrt{\frac{1}{\varsigma} + \frac{1}{\vartheta} + \frac{1}{\varsigma} \cdot} \tag{15}$$

$$SE(\log OR) = 0.6 \tag{16}$$

$$C.I = -0{:}236 \pm 1.96 \times 0.6 \tag{17}$$

$$C.I = -1.583, 1.1 \tag{18}$$

This is the confidence interval for log(OR). In order to calculate the confidence interval for odds ratio, the intervals need to convert into its antilog. Equation (19) gives the antilog of the limit.

$$C.I = -0.205, 3.0 \tag{19}$$

This confidence interval includes 1. Thus, it can be concluded that the odds ratio is approximately equal to 1. This shows that there is no significant difference between correct identification and wrong identification. Thus, it can be assumed that the selection was done randomly. Therefore, hiding the existence of the message of steganography is successful.

B. Performance of extracting the message

In steganography evaluation, it is important to check the ability of extracting the correct message from the carrier file. The message can be a video file, an audio file, an image or a text document. In order to check the performance, 5 video files, 4 audio files, 3 images and 3 text files were chosen. 10 students reading for a Computer Science degree are chosen from an academic institution to evaluate the original message with the extracted message. They were asked to rank the messages from 0 to 4 where 0 indicates that there is no difference between the original message and the extracted message and 4 indicates that there is a 100% difference between the two messages. Table II displays the ranks assigned for each situation. Table III shows the results obtained from one of the participants. The average rank for video, audio, image and text are calculated for all 10 participants. Table IV shows the average rank for all the participants. Fig 4 displays the graph of the average rank.

TABLE II

RANKS ASSIGNED FOR EVALUATION

Rank	Meaning
0	No difference
1	Very small difference
2	Average difference
3	Large difference
4	Total difference

TABLE III

RESULTS OF A SINGLE PARTICIPANT

	File 1	File 2	File 3	File 4	File 5	Average rank
Video	0	1	0	0	0	0.2
Audio	1	1	0	0		0.25
Image	0	0	1			0.33
Text	0	0	0			0

TABLE IV

RESULTS (AVERAGE RANK) FOR ALL PARTICIPANTS

	Video	Audio	Images	Text
Participant 1	0.2	0.25	0.33	0
Participant 2	0	0	0	0
Participant 3	0.25	0.25	0	0
Participant 4	0.5	0	0	0
Participant 5	0	0.25	0	0
Participant 6	0	0.25	0	0
Participant 7	0.25	0	0	0
Participant 8	0	0.25	0	0
Participant 9	0	0	0.33	0
Participant 10	0.25	0	0	0
Average	0.145	0.125	0.663	0

From these results, it is clear that within the rank range 0-4, the average rank for video is 0.145, average rank for audio is 0.125 which is less than average rank for video, average rank for image is 0.0663 which is less than both average rank for video and audio, and average rank for text is exactly 0. Thus, it is clear that the system has a high performance of extracting the data correctly from the carrier file.

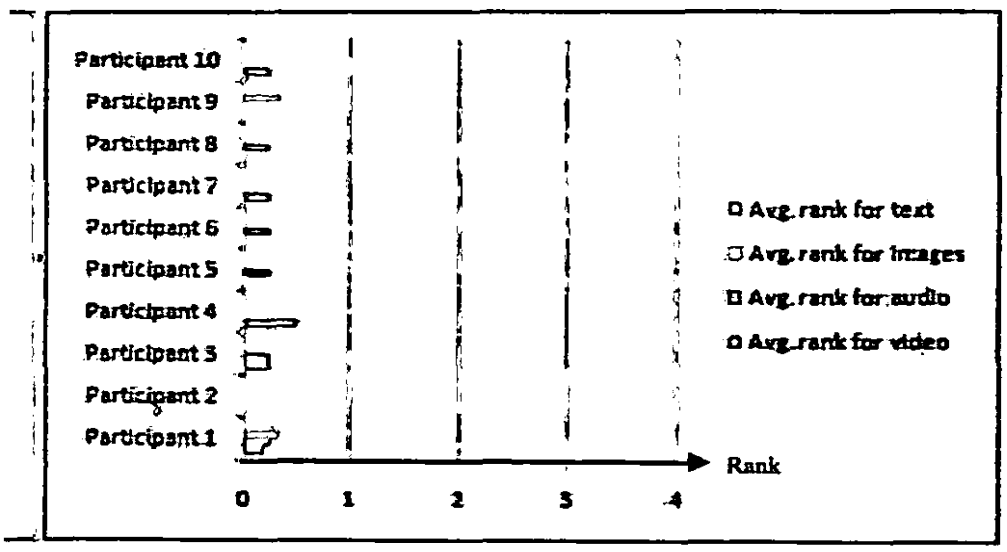

Fig. 4 Graph for results of participants

IV. DISCUSSION

For various security issues, people tend to have an interest beyond cryptography techniques as cryptography cannot hide the existence of the message. Steganography was introduced in order to address this matter. The advantage of steganography over cryptography is that only the message will be hidden in cryptography where as in steganography, the existence of the message can be hidden.

The current study was done in order to hide a video in a visual content. The importance of hiding a video is that a video usually contains a large file because it consists of a lot of data. The main technique used in this study is changing the LSB value of the carrier file. In order to do that, the visual file is separated into frames. Each frame consists of a set of byte values. The LSB value is changed in each byte so that the LSB carries the hidden message.

The limitation of the suggested approach is that it only supports an Uncompressed MS AVI file to use as a carrier file. However, for the specified file type (uncompressed AVI), it is well tested and it works well. In order to increase the security of data, the message was encrypted before embedding to the carrier file. This encryption method uses both a key file and a password. This will omit the password guessing and 2 XOR logic gates are used to encrypt the message.

Even though the main objective of this project is to hide a video in a visual, we have tested hiding an image, an audio file and a text file as well in a visual. All these types provide perfect performance. However, when hiding a text message, if at least 1 byte is dropped from the original message accidently, then the file will be totally corrupted and the message will not be retrieved back successfully.

V. CONCLUSION

The main objective of this research is to hide a video in a visual. It is implemented by changing the least significant bit of the visual file bite stream into a message file.

The message was then converted into byte code and encrypted before embedding to a carrier file. The functions of avifill32.dll were used with C# wrapper files.

Even though this approach is successful, the issue faced in this method is that the carrier file should be an uncompressed AVI file.

The system was evaluated for its main functionalities, hiding the existence of the message and extracting the message correctly. The results show that the system performs well in hiding the message file and in extracting the message from the carrier file.

Further suggestions for the system are to use more advanced encryption methods to encrypt the message, and to develop the system using video with sounds as container.

REFERENCES

[1] A. Al-Frajat, H. Jalab, Z. Kasirun, A. Zaidan, and B. Zaidan. Hiding data in video file: An overview. J. Appl. Sci, 10(15):1644–1649, 2010.

[2] A. Bhaumik, M. Choi, R. Robles, and M. Balitanas. Data hiding in video. International Journal of Database Theory and Application, 2(2), 2009.

[3] J. Bland and D. Altman. Statistics notes: the odds ratio. BMJ: British Medical Journal, 320(7247):1468, 2000.

[4] J. Chae and B. Manjunath. Data hiding in video. In Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on, volume 1, pages 311–315. IEEE, 1999.

[5] A. Glas, J. Lijmer, M. Prins, G. Bonsel, and P. Bossuyt. The diagnostic odds ratio: a single indicator of test performance. Journal of clinical epidemiology, 56(11):1129–1135, 2003.

[6] N. Johnson. Steganography, George Mason University. Information System and Software Engineering, www. jjtc. com/stegdoc/steg1995. html, 1995.

[7] R. Petrovic, J. Winograd, K. Jemili, and E. Metois. Data hiding within audio signals. In Telecommunications in Modern Satellite, Cable and Broadcasting Services. 1999. 4th International Conference on, volume 1, pages 88–95. IEEE, 1999.

[8] C. Xu, X. Ping, and T. Zhang. Steganography in compressed video stream. In Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on, volume 1, pages 269–272. IEEE, 2006.