Eng.& Tech. Journal ,Vol. 30, No.4 , 2012

Hiding Encrypted Color Image within MPEG-2 Video

Dr. Alaa A. Abdul Latef
Ibn Al_Haitham College, University of Baghdad/ Baghdad
Email: Alaa_3b@yahoo.com
Dr. Firas A. Abdul Latef
Ibn Al_Haitham College, University of Baghdad/ Baghdad
Email: firas.alobaedy@gmail.com

Received on: 25/5/2011 & Accepted on: 5/1/2012

ABSTRACT

In any communication, security is the most important issue in today's world. Lots of data security and data hiding algorithms have been developed in the last decade. Hiding data in digital videos makes possible to hide a great quantity of information when compared to techniques used in images. This work presents the encryption process of an image with combination of bit and pixel permutation technique depending on random key, and then steganography technique that hiding each bit of encrypted image in selected block 8*8 pixels(selected block process depending on high blue color ratio), of the frame and can extracted that bit after compressed with mpeg-2. Experimental results show the success of hidden extracted data from the sequence of frames, and also indicate the effectiveness of the implementation steganography compressed video with high security features.

Keywords: steganography; hiding data; encryption; video compression; mpeg2.

اخفاء صورة ملونة مشفرة في فيديو (MPEG-2)

الخلاصة

في اي مجال من مجالات الاتصال، الامنية هي اهم عنصر في العالم الان. العديد من طرق امنية البيانات و خوارزميات اخفاء البيانات قد تطورت في العقود الاخيرة. ان اخفاء البيانات في ملفات الفديو تسمح لاخفاء كمية كبيرة من البيانات مقارنة بالطرق التي تخفي البيانات في صورة مثلا. في هذا البحث هناك تقنية لتشفير الصورة باستخدام مزج بين طريقة تبديل مواقع النقاط الضوئية وطريقة ابدال مواقع البتات في نفس النقطة الضوئية وتتم عملية الابدال في الطريقتين حسب مفتاح عشوائي و يتم اخفاء كل بت من الصورة المشفرة في مجموعة 8*8 من النقاط الضوئية(و ان اختيارالمجاميع للاخفاء يكون بالاعتماد على نسبة لونية عالية من اللون الازرق). ويمكن استرجاع البيانات بعد الضغط بطريقة-MPEG(2).

اظهرت النتائج نجاح هذه الطريقة في مرحلة الاخفاء واسترجاع البيانات من مجموعة من الصور. وايضا هي طريقة ناجحة في الاخفاء باستخدام ضغط الفيديو حيث كانت الامنية عالية.

الكلمات المرشدة: اخفاء البيانات، التشفير، ضغط الفيديو، MPEG-2

https://doi.org/10.30684/etj.30.4.7
2412-0758/University of Technology-Iraq, Baghdad, Iraq
This is an open access article under the CC BY 4.0 license http://creativecommons.org/licenses/by/4.0

INTRODUCTION

The success of the Internet facilitates communications of people, but also enables illegal users to access data transmitted on the Internet. To protect the important data from being illegally accessed, various modern cryptosystems and steganography can be used to protect the content of these data prior to their transmissions [1][2].

In simple words, Steganography can be defined as the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information.

Though the concept of steganography and cryptography are the same, but still steganography differs from cryptography. Cryptography [3] focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret.

In recent years, most of the researchers in this filed are focusing their work on hiding data within Video files. The reason of that is the possibility of hiding a large amount of data within frames instead of using one still image. However, direct image techniques cannot be simply applied to compressed videos because temporal compression add noise to the hidden information potentially making it useless [4].

In this paper, steganography algorithm method based on the mpeg-2 compression is presented. The presented method used to embed each bit in encrypted image in chosen 8*8 block of blue color by changing its pixels value with small amount depending on threshold and the extracted process can achieve after the MPEG-2 compression was applied.

The rest of the paper is organized as follows. In section 2, the review of MPEG compression is introduced. It will also include some details of MPEG-2 compression. In section 3 the encryption system that used to encrypt the color image is explained. The proposed algorithm will be described in section 4. All the obtained results and discussions presented in section 5. Section 6 will present the conclusions referring to the results presented in this paper.

REVIEW OF MPEG COMPRESSION

The basic job of MPEG is to take analog or digital video signals and convert them into packets of digital information that are more efficiently transported on modern networks. In theory, a video stream is a sequence of discrete images. In practice, successive images are highly interrelated. Barring cut shots or scene changes, any given video frame is likely to bear a close resemblance to neighboring frames. MPEG exploits this strong correlation to achieve far better compression rates than would be possible with isolated images [5].

Each frame in an MPEG image stream is encoded using one of three schemes:

I-frame , or intra-frame, are coded as isolated images.

P-frame , or predictive coded frame, are based on the previous I- or P-frame.

PDF created with pdfFactory Pro trial version www.pdffactory.com

B-frame, or bidirectionally predictive coded frame, are based on either or both the previous and next I- or P-frame [5].

Figure (1) shows an MPEG stream containing all three types of frames. I-frames and P-frames appear in an MPEG stream in simple, chronological order. However, B-frames are moved so that they appear *after* their neighboring I- and P-frames. This guarantees that each frame appears after any frame upon which it may depend. An MPEG encoder can decode any frame by buffering the two most recent I- or P-frames encountered in the data stream. Also, Figure 1 shows how B-frames are postponed in the data stream so as to simplify decoder buffering. MPEG encoders are free to mix the frame types in any order. When the scene is relatively static, P- and B-frames could be used, while major scene changes could be encoded using I-frames. In practice, most encoders use some fixed pattern [5].

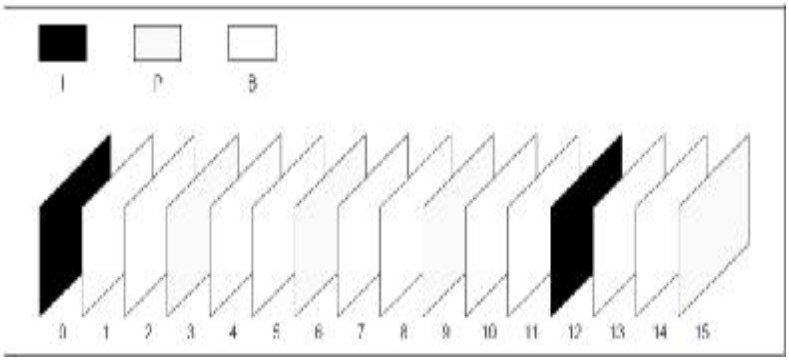


Figure (1) typical sequence of pictures in display order.

Since I-frames are independent images, they can be encoded as if they were still images. The particular technique used by MPEG is a variant of the JPEG technique (the color transformation and quantization steps are slightly different). I-frames are very important for use as anchor points so that the frames in the video can be accessed randomly without requiring one to decode all previous frames. To decode any frame needed only find its closest previous I-frame and go from there. This is important for allowing reverse playback, skip-ahead, or error-recovery [5].

The intuition behind encoding P-frames is to find matches, *i.e.*, groups of pixels with similar patterns, in the previous reference frame and then coding the difference between the P-frame and its match. To find these "matches" the MPEG algorithm partitions the P-frame into 16x16 blocks. The process by which each of these blocks is encoded is illustrated in Figure 2 [5][6].

The MPEG-2 standards are a strict superset of MPEG-1 (MPEG-2 decoders can decode MPEG-1) and the data format is upwardly compatible. MPEG-2 aims for a different set of optimal compression parameters than the earlier standard [6].

PDF created with pdfFactory Pro trial version www.pdffactory.com

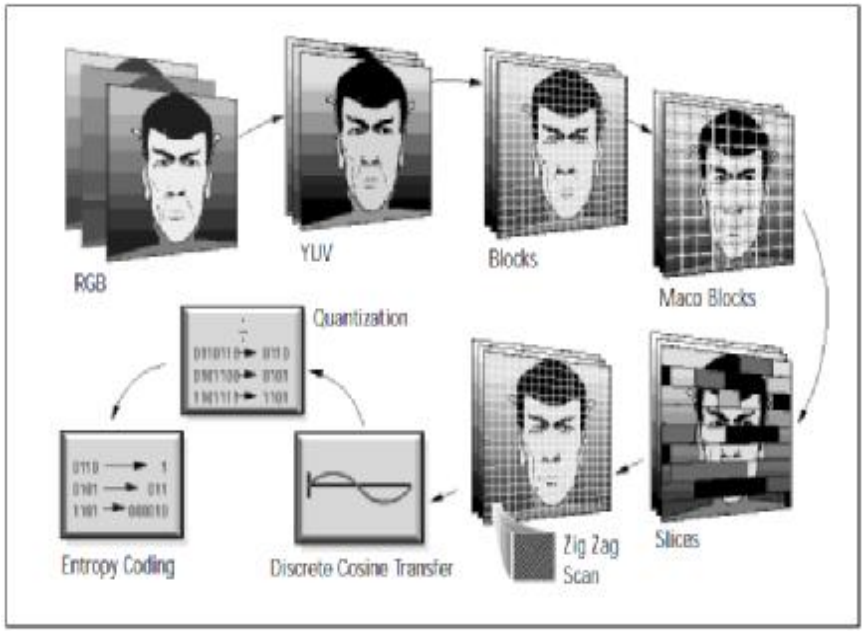


Figure (2) MPEG Compression Process.

ENCRYPTED IMAGE

When it is necessary to securely transmit data in open networks, the encryption must be performed. Most of the cryptographic algorithms were mainly developed for text data. Some of these algorithms are good for textual data and also suitable for image. One of these algorithms is permutation techniques [7].

Encryption Process

The main idea behind this proposed work is modified the permutation techniques to be suitable for color image using same random key for bit and pixel permutation for each color in the image as shown in Algorithm (1):

ALGORITHM (1): ENCRYPTION IMAGE

Input: Color Image.

Output: Encrypted color image.

1. Generate three different encryption keys for (red, green, and blue) colors randomly from (1-8) without redundancy to get red_key, green_key and blue_key.
2. Convert each pixel in red color of image to binary code then permute each bit in the pixel depending on the red_key, this process is repeated for green and blue color depending on its keys.

PDF created with pdfFactory Pro trial version www.pdffactory.com

3. Take each 8 pixels of red color and permute them depending on red_key, this process is repeated for green and blue color depending on its keys.

End

Decryption Process:

The decryption process is the reverse order process of encryption and it mentioned as shown in Algorithm (2):

ALGORITHM (2): DECRYPTION COLOR IMAGE

Input: Encrypted color image, three keys.

Output: Decrypted color image.

1. Take each 8 pixels of red color of the encrypted image and permute them depending on red_key, this process is repeated for green and blue color depending on its keys.
2. Convert each pixel in red color of image to binary code then permute each bit with the red_key, this process is also repeated for green and blue color depending on its keys.

End

PROPOSED ALGORITHM FOR HIDING PROCESS

The proposed steganography method comprises an embedding process and an extraction process. The details of the embedding and extraction process are described in the following subsections:

Embedding Process

The proposed embedding process is as shown in Algorithm (3):

Algorithm (3): Embedding image in video

Input: video file, image

Output: stego video

Step 1: encrypted the hiding image by using permutation techniques as mention in section 3.1 then convert it to binary numbers.

Step 2: selected video file.

Step 3: read the frames and selected I-frame.

Step 4: divide the blue color of the frame into 8*8 blocks (this division is suitable for compression algorithm), and then selected the blocks that have high ratio of blue color (the average value of the 64 pixels is greater than some high threshold like 200).

Step 5: each bit in encrypted image embedding by the following process:

a) If the embedded bit is 0:

I. Divide 64 pixels of the selected block by threshold like 21 and store the remainder in temporary array called remain_arr.

II. Change the 64 pixels value with different amount that make all the remain_ arr after changing equal to 5(the value between 0-10), with consideration that each pixel after changing not exceed 255 or less than 0.

PDF created with pdfFactory Pro trial version www.pdffactory.com

b) If the embedded bit is 1:

I. Divide 64 pixels of the selected block by threshold like 21 and store the remainder in temporary array called remain_arr.

III. Change the 64 pixels value with different amount that make the remain_ arr after changing equal to 15(the value between11-20), with consideration that each pixel after changing not exceed 255 or less than 0.

Repeat step 5 three times for three blocks (that mean store one bit in three blocks to achieve very high accuracy)

Step 6: if selected blocks is not equal to hiding bit, take the successor I-frame and go to step 4.

End.

Extracting Process:

The proposed extracting process is surviving after MPEG-2 compression. The details of it are as shown in Algorithm (4):

Algorithm (4): Extracted image from video

Input: Stego video

Output: Image

Step 1: read the frames from the stego video and selected the I-frame.

Step 2: divide the blue color of the frame into 8*8 blocks, then selected the blocks that have high ratio of blue color (for example the average value of the 64 pixels is greater than 200).

Step 3: for each of selected block extract the embedding bit as the following process:

I. Divide 64 pixels of the selected block by the same threshold of embedding like 21 and store the remainder in temporary array called remain_arr.

II. If the values of remain_arr are between (0 -10) then the extracted bit is 0.

III. If the values of remain_arr are between (11-20) then the extracted bit is 1.

Step 4: take each 3 extracted bits and check if two or three equal to 0 then the extracted bit is 0, if two or three bit is equal to 1 the extracted bit is 1.

Step 5: if the extracted bits not equal to hidden bit, then take the successor I-frame and go to step 2.

Step 6: after extracted all bits then reconstruct the encrypted image and then decrypted using algorithm in section 3.2.

End.

EXPERIMENTAL RESULTS

Several experiments have been done to examine the performance of the proposed embedding method. Many I-frame images with different textural properties were taken as the cover images; one of them is shown in Fig. 3 (a).

PDF created with pdfFactory Pro trial version www.pdffactory.com

The first step is encrypted the hiding image by using the permutation technique as described in section 3, Figure 4(a) shows the hiding image with size 128*128 before encryption while Figure 5(a)shows the hiding image after encryption. Then take the blue color of the I-frame and divided it into 8*8 blocks, select the suitable blocks (the blocks that have the average of all pixels value is equal or greater to 200) in Fig. 3 (a)there are 353 selected blocks with this condition. After that use the selected blocks to hiding the bits of encrypted image, choosing number 21 as threshold for hiding process, because the changing in pixel value depending to that threshold is not very big (not over as possible 4 least significant bits) and this threshold larger enough to cover the error of compression process.
Fig.6 (a) shows the stego frame.

The proposed algorithm applied to many other frames, like in Fig. 3(b) as a cover frame with Fig. 4 (b) as a hiding image, and Fig. 3(c) as a cover with Fig 4(c) as a hiding image.
In this proposed work, the amount of noise added into the color (cover) frame calculated using Peak Signal to Noise (*PSNR*) equation [8].

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \qquad \text{.....(1)}$$

Mean Square Error (MSE): It is the measure used to quantify the difference between the initial and the distorted or noisy image. Let N, M the size of the cover image(x), and stego image (x^) respectively:

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [x(m,n) - \hat{x}(m,n)]^2 \qquad \text{.....(2)}$$

Table (1) shows the PSNR and MSE between the cover frames and stego frames

(a)

(b)

PDF created with pdfFactory Pro trial version www.pdffactory.com

(C)

Figure (3) the cover frames (a) train frame. (b) car frame (c) persons frame

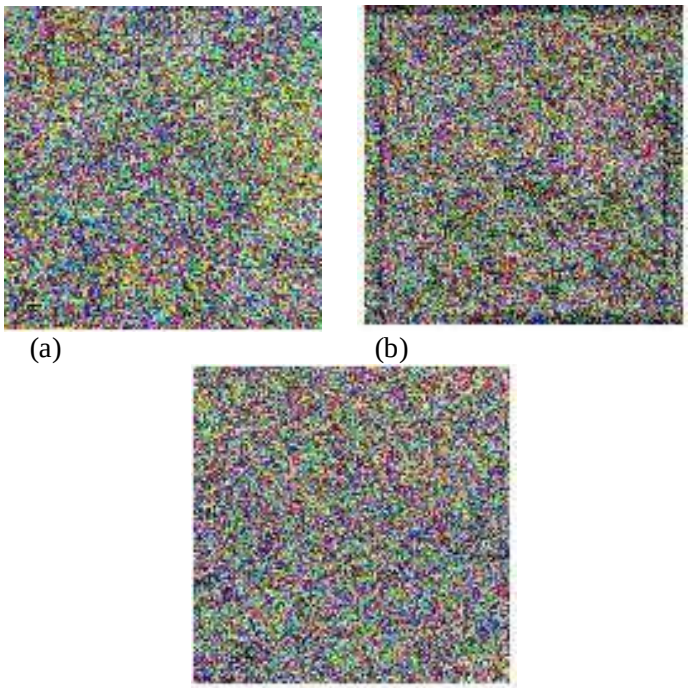
(a)

(b)

(c)

Figure (4) The hiding images,(a) bird (b) roses (c) trees

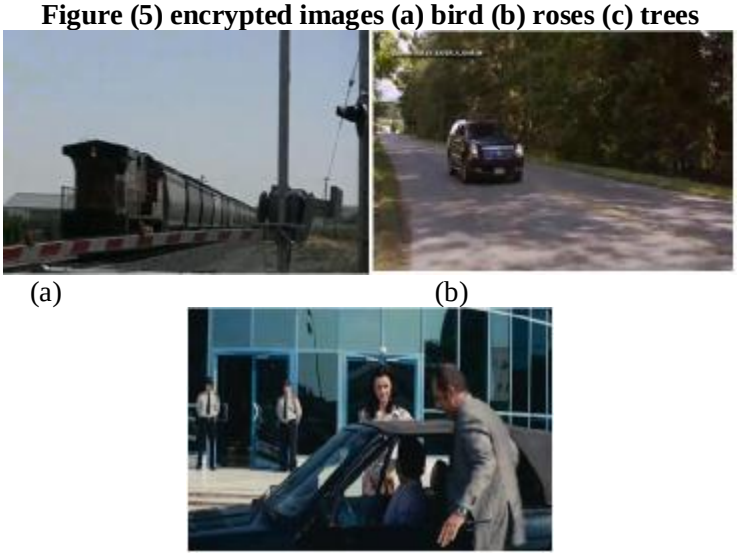
PDF created with pdfFactory Pro trial version www.pdffactory.com



(a) (b)

(c)

Figure (5) encrypted images (a) bird (b) roses (c) trees



(a) (b)

(c)

Figure (6) The stego_frame (a) train frame. (b) car frame (c) persons frame

PDF created with pdfFactory Pro trial version www.pdffactory.com

Table (1): comparison the PSNR and MSE between cover frames and stego frames

Name of frames	PSNR	MSE
Train frame	44.02	2.57
Car frame	50.87	0.53
Persons frame	43.75	2.73

CONCLUSIONS

In this paper, proposed approach of secure video steganography has been invented. The basis of this method is use the MPEG-2 video as separate frames and select I-frames to hides the information inside. As the experiment result shows the success (depending on PSNR values) of the hidden data within select frame, extract data from the frames sequence, these functions without affecting the quality of the video. To achieve more security the secret image is also encrypted using permutation techniques. In the video steganography there is a flexibility of make a selective frame and selected blocks in that frame to higher the security of the system or using the whole frame too high a huge amount of data hidden. Due the security and accuracy issues in the proposed method taking only the blue color and select 8*8 blocks of high ratio of blue which is in buffer, this idea make to guarantee the protection of data.

REFERENCES

[1]Stallings W., "Cryptography and Network Security: Principles and Practice", third ed., Pearson Education, 2003.

[2]Ahmed J., "An Overview of Image Steganography", 2005

[3]Kumari M., Khare A.,and Khare P. "JPEG Compression Steganography &Crypography Using Image-Adaptation Technique", journal of advances in information technology, vol. 1, no. 3, august 2010.

[4]Tarik F., Jumari K., Abd. Samad S., abdulgader A. "A Large Capacity Steganography Using Bit Plane Complexity Segmentation (BPCS) algorithm for MPEG-4 Video", International Journal of Computer and Network Security, 67 Vol. 2, No. 7, July 2010

[5]Blelloch G., "Introduction to Data Compression", September 25, 2010

[6]Ruiu D., Packard H., "An Overview of MPEG-2" Test and Measurement publications,1997.

[7]Mitra A., Subba Y. V., Prasanna S.,"A New Image Encryption Approach using combination Permutation Techniques", International Journal of Electrical and Computer Engineering,2006.

[8]Rabbani M. and jones P., "Digital image compression Techniques", VolTT7, SPIE Optical Engineering Press, Bellvue, Washington (1991).

PDF created with pdfFactory Pro trial version www.pdffactory.com