A Highly Secure Video Steganography using Hamming Code (7, 4)

Ramadhan J. Mstafa and Khaled M. Elleithy, *Senior Member, IEEE*
Department of Computer Science and Engineering
University of Bridgeport
Bridgeport, CT 06604, USA
rmstafa@bridgeport.edu

Abstract— **Due to the high speed of internet and advances in technology, people are becoming more worried about information being hacked by attackers. Recently, many algorithms of steganography and data hiding have been proposed. Steganography is a process of embedding the secret information inside the host medium (text, audio, image and video). Concurrently, many of the powerful steganographic analysis software programs have been provided to unauthorized users to retrieve the valuable secret information that was embedded in the carrier files. Some steganography algorithms can be easily detected by steganalytical detectors because of the lack of security and embedding efficiency.**

In this paper, we propose a secure video steganography algorithm based on the principle of linear block code. Nine uncompressed video sequences are used as cover data and a binary image logo as a secret message. The pixels' positions of both cover videos and a secret message are randomly reordered by using a private key to improve the system's security. Then the secret message is encoded by applying Hamming code (7, 4) before the embedding process to make the message even more secure. The result of the encoded message will be added to random generated values by using XOR function. After these steps that make the message secure enough, it will be ready to be embedded into the cover video frames. In addition, the embedding area in each frame is randomly selected and it will be different from other frames to improve the steganography scheme's robustness. Furthermore, the algorithm has high embedding efficiency as demonstrated by the experimental results that we have obtained. Regarding the system's quality, the Pick Signal to Noise Ratio (PSNR) of stego videos are above 51 dB, which is close to the original video quality. The embedding payload is also acceptable, where in each video frame we can embed 16 Kbits and it can go up to 90 Kbits without noticeable degrading of the stego video's quality.

Keywords– **Video Steganography, Hamming Code, Linear Block Code, Security, Embedding Efficiency, Embedding Payload.**

I. Introduction

INTERNET makes people's lives much easier than before; they can use it to pay their bills, buy their goods, exchange important messages between parties at far distances, and many other things. Without protecting that valuable information, attackers can obtain them in different ways. Steganography is one of the methods that protects and hides valuable data from unauthorized people and even without them having any suspicion of the data's existence. Human Visual System (HVS) can't recognize a slight change that happens in the media cover such as audio, image and video [1].

There are two important factors that every successful steganography system should take into consideration, which are embedding efficiency and embedding payload. First, the steganography scheme that has a high embedding efficiency means good quality of stego data and less amount of host (carrier) data are going to be changed [2]. Any obvious distortion to the viewers will increase the probability of the attacker's suspicion and the secret information can be easily detected by some of the steganalysis tools [3]. These kinds of schemes are difficult to be detected by the steganalytical detectors. The security of the steganography scheme is depending directly on the embedding efficiency [4].Second, the high embedding payload means the capacity of secret information to be hidden inside host data is large. To be more specific, the two factors embedding efficiency and embedding payload have a type of contradiction. Increasing efficiency will cause the capacity of embedding to have a low payload. Changing the balance between these two factors mainly depends on the users and the type of steganography scheme [2].

The rest of the paper is organized as follows. Section 2 presents some of the previous work. Section 3 introduces an overview of the Linear Block Code and Hamming code, and then presents our Steganography scheme. Section 4 we discuss experimental results and analyze them. Section 5 provides the conclusions.

II. Related work

In 2009, Eltahir et al presented a video steganography based on the Least Significant Bit (LSB). Authors tried to increase the size of the secret message into the video frames. They analyzed video into frames then each frame was used as a still image. A 3-3-2 approach has been used which means taking the LSB of all RGB color components (3-bits of Red, 3-bits of Green, and 2-bits of Blue). The reason for taking 2-bits of blue color is because the HVS is more sensitive to the blue color than the other two colors. The results demonstrated that the hidden message can take one third of overall video size. This is considered an improvement of the LSB algorithm [5].

In 2010, Feng et al proposed a novel of video steganography scheme based on motion vectors as carriers to embed the secret message in H.264 video compression processing. The algorithm also uses the principle of linear block codes to reduce motion vectors' modification rate. The algorithm has a good quality of stego data, which is proved by the low modification rate of motion vectors. The PSNRs that were obtained in both flower and foreman videos are more than 37 dB [6].

In 2011, Hao et al proposed a novel video steganography method based on a motion vector by using matrix encoding. A motion vector component that has high amplitude among both horizontal and vertical components is chosen to embed the secret message. The Human Visual System can see the change that occurs when the object is moving slowly, while if the same object moves quickly the HVS won't be able to feel the change that happens. Motion vectors with large size are selected for embedding the secret message. The macro blocks that are moving quickly will generate motion vectors with large amplitude. The direction of macro blocks depends on the motion vectors' components. For example, if the vertical component is equal to zero that means the macro block direction is moving vertically. The quality of the tested videos that was obtained is more than 36 dB [7].

In 2012, Rongyue et al proposed an efficient BCH coding for steganography which is embedding the secret information inside a block of cover data by changing some coefficients. Authors have improved the computational time of the system and the complexity becomes low because of the system's linearity [8].

In 2013, Liu et al proposed a robust data hiding scheme in H.264 compressed video stream, where they have prevented a drift of intra-frame distortion. To give the system more robustness, authors have encoded the message using BCH code before making the embedding process. The host data is the DCT coefficients of the luminance I-frame component. The obtained results have a high quality and robustness [9].

III. The Proposed Steganography Scheme

Our algorithm uses an uncompressed video stream which is based on the frames as still images. First the video stream is separated into frames and each frame's color space is converted to YCbCr. The reason for using YCbCr color space is that it removes correlation between Red, Green, and Blue colors. A luminance (Y) part is brightness data, which the human eyes are more sensitive to than the color parts. As a result, the color parts (chrominance) can be subsampled in the video stream and some information will be discarded.

A. Linear block codes

A block code is a linear block code if a summation of two codewords is also a codeword, and the binary linear block code is applied to bits of blocks. An (n, k) binary linear block code has 2^k columns and 2^{n-k} rows in a linear code array. Where k is refers to k-dimensional subspace and n refers to n-dimensional vector space.

$V_n = \{(C_0, C_1, \ldots, C_{n-1})|C_j \in GF(2)\}$ where n is the length of the code and k is a number of symbols. In the standard array, there are no two equal vectors at the same row. Assume C is a (n, k) code on Galois Field GF (2), then:

- All X vectors of length n belong to a coset of C.
- Each coset has 2^k vectors.
- Two cosets either overlap or intersect completely or not at all.
- If C+Y is a coset of C and X as belong to (C+Y), then C+X=C+Y.

B. Hamming codes (7, 4)

The Hamming code is one of the most well-known block code methods that can do both error detection and correction on a block of data. In the Hamming code technique, the original information will be coded by adding some extra data with the minimum amount of redundancy, which is called the codeword, of length n bits [10]. The added part consists of parity information of length (n-k) bits where k is the length of message that is expected to be coded [2]. In this paper, the (7, 4) Hamming code is used that can detect and correct a single bit error of data or parity. First, the message (m_1, m_2, m_3, m_4) of length k bits (k=4) is encoded by adding three parity bits (p_1, p_2, p_3) to become the codeword of length n (n=7), which is ready for transmission. There are different ways to mix both types of data (message and parity) together and the general combination is to put the parity bits at position 2^i such as (p_1, p_2, m_1, p_3, m_2, m_3, m_4) where i=0, 1, ... ,(n-k-1).

The Hamming codes are linear codes so they have two matrices: parity-check matrix H and generator matrix G, which they need for both encoding and decoding. On the encoding side, each message M, which consists of 4-bits, will be multiplied by the generator matrix and then have modulo of 2 applied; the result is the codeword X of 7-bits ready to be sent through a noisy channel.

$$\mathrm{X = M \times G\ Where\ G =} \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

On the decoding side, for the purpose of checking the encoded message of 7-bits R (data + parity) will be received, then will be multiplied by the transpose of the parity-check matrix, and taking modulo of 2 again.

$$Z = R \times H^T \text{ , where } H^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

The result is a syndrome vector Z (z_1, z_2, z_3) of three bits, which has to be all zeroes (000) if it's an error-free message. Otherwise, any change in the message during transmission will lead to flipping one or more bits of the message; then it needs an error correction process.

Example: Assume we have a message M_1 of 4-bits (1, 1, 1, 1) and the Hamming code (7, 4) is done by the following steps:

1- Calculate $X = M_1 \times G$, the result is (3, 3, 3, 1, 1, 1, 1) then taking modulo of 2, the result is the codeword X=1111111, which is sent through the communication channel.

2- At the destination, to get the correct message the syndrome vector Z must be zero. The first assumption R=1111111 is received without any errors. Then Z will become (0, 0, 0), where $Z = R \times H^T$.

3- In the second assumption, suppose that during transmission due to the noisy channel one of the bits has changed. The received data will be R=1111011, then calculating the syndrome we will get Z=433, and taking modulo of 2 the syndrome will become Z=011.

4- Comparing Z value with the parity-check matrix H, it appears that the Z value (0, 1, 1) is equal to the 5th row (0, 1, 1) of the H matrix, which means that the 5th bit of R has changed.

5- Correcting the 5th bit of R by flipping it to 1, R then is corrected to become (1, 1, 1, 1, 1, 1, 1).

6- The four first bits are the original message M_1 (1, 1, 1, 1) and the last three other bits will be ignored.

C. Data embedding phase

Data embedding is a process of hiding a secret message inside host videos, and it can be done by the following steps:

1- Convert the video stream into frames.

2- Separate each frame into Y, U and V components.

3- Change the position of all pixels in three components Y, U and V by a special key.

4- Convert the message (which is a binary image) to a one dimension array, and then change the position of the whole message by a key.

5- Encode each 4 bits of the message using Hamming (7, 4) encoder.

6- The result of the encoded data, which consists of 7 bits (4 bits of message + 3 bits of parity) is XORed with the 7 bits of random value using a key.

7- Embed the result of those 7 bits in one pixel of YUV components (3-bits in Y, 2-bits in U and 2-bits in V).

8- Reposition all pixels of YUV components to the original frame pixel position.

9- Rebuild the video stream again from those embedded frames.

There are three keys that have been used in this work, which give to our steganography scheme an improvement in both security and robustness. Those keys are shared between sender and receiver in both data embedding and extracting processes. The first key is used to reposition pixels in Y, U, V, and the secret message into a random position, which makes the data chaotic. In order to select the locations for embedding the secret message into the host data, the second and third keys are used. They are used to pick the random rows and columns respectively in each chaotic Y, U and V component. The XOR function that has been used increases the quality of the system. The block diagrams of the data embedding phase and the data extracting phase are illustrated in Figure 1 and Figure 2 respectively.

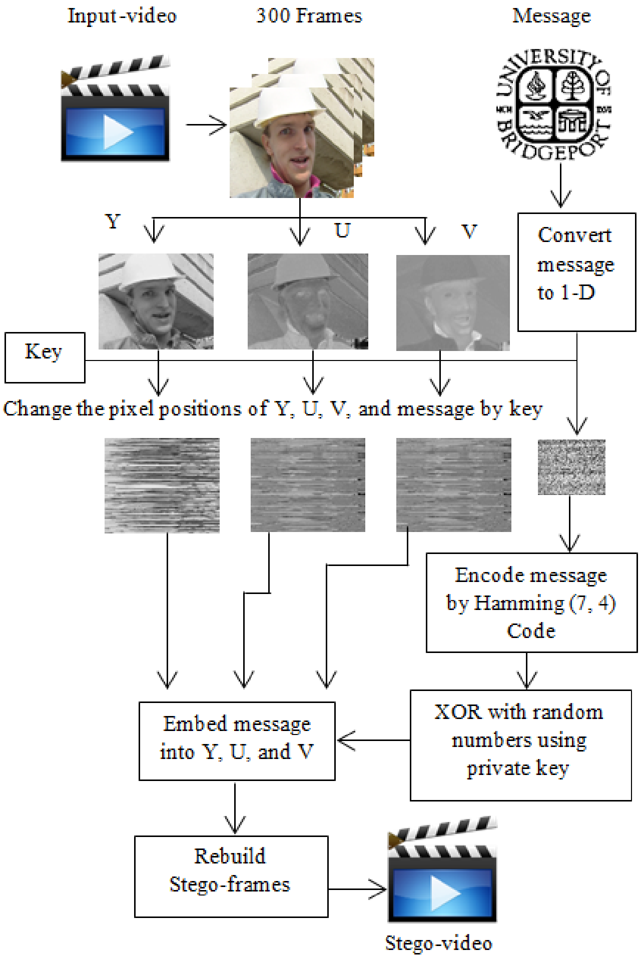
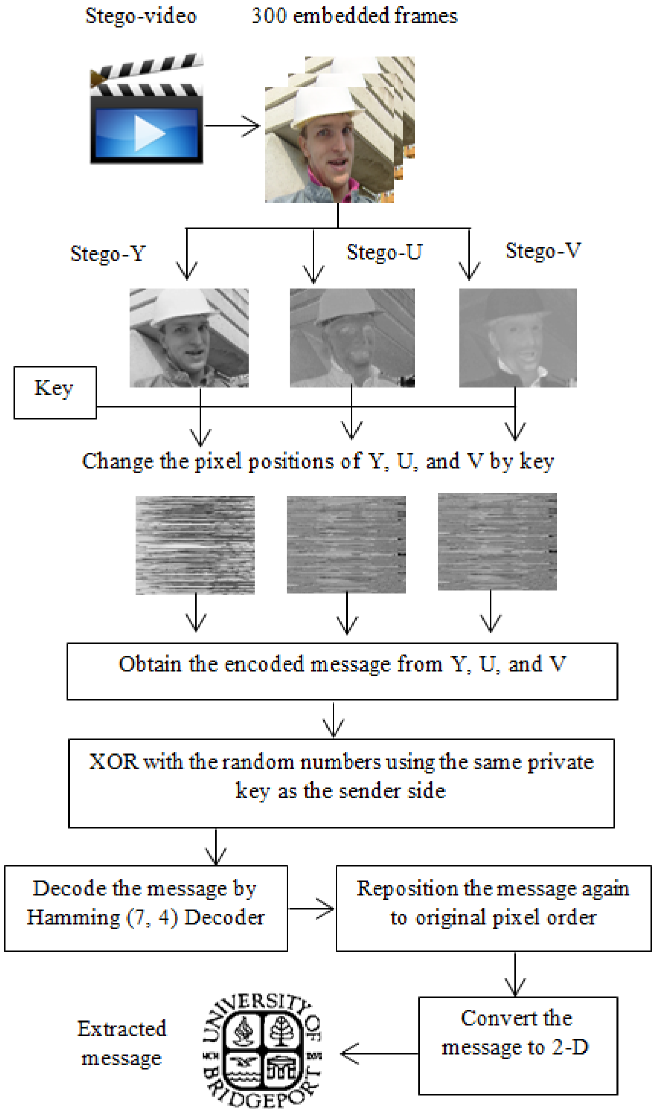

Input-video
300 Frames
Message
Y
U
V
Convert message to 1-D
Key
Change the pixel positions of Y, U, V, and message by key
Encode message by Hamming (7, 4) Code
Embed message into Y, U, and V
XOR with random numbers using private key
Rebuild Stego-frames
Stego-video

Figure 1: Block diagram for data embedding phase.

D. Data extracting phase

Data extracting is a process of retrieving the secret message from the stego videos which can be done by the following steps:

1- Convert the video stream into frames.
2- Separate each frame into Y, U and V components.
3- Change the position of all pixel values in the three Y, U, and V components by the special key that was used in the embedding process.
4- Obtain the encoded data from the YUV components and XOR with the random number using the same key that was used in the sender side.
5- Decode 4 bits of the message by the Hamming decoder.
6- Reposition the whole message again into the original order.
7- Convert the message array to 2-D.



Stego-video
300 embedded frames
Stego-Y
Stego-U
Stego-V
Key
Change the pixel positions of Y, U, and V by key
Obtain the encoded message from Y, U, and V
XOR with the random numbers using the same private key as the sender side
Decode the message by Hamming (7, 4) Decoder
Reposition the message again to original pixel order
Convert the message to 2-D
Extracted message

Figure 2: Block diagram for data extracting phase.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this paper, a database of nine standard Common Interchange Format (CIF) video sequences is used, with the size (288 X 352) and the format 4:2:0 YUV. All video sequences are equal in length with 300 frames in each one. The secret message is a binary image logo for the University of Bridgeport (UB) with a size of 128 X 128 pixels. The MATLAB software program is used to implement this work and test our experiment results.

In Figure 3, an example of one frame (frame no. 111) in the Foreman video is chosen. The first part of the figure shows that the three components of the 111th frame are separated. Then it shows some locations that have been chosen randomly for the secret message. The embedded locations are different in each component inside one frame and they differ from one frame to next, which mainly depends on the private key. The second part of the figure shows frame no. 111 before and after the embedding process. The last part of the figure shows the whole message that has been embedded and extracted 100% correctly.

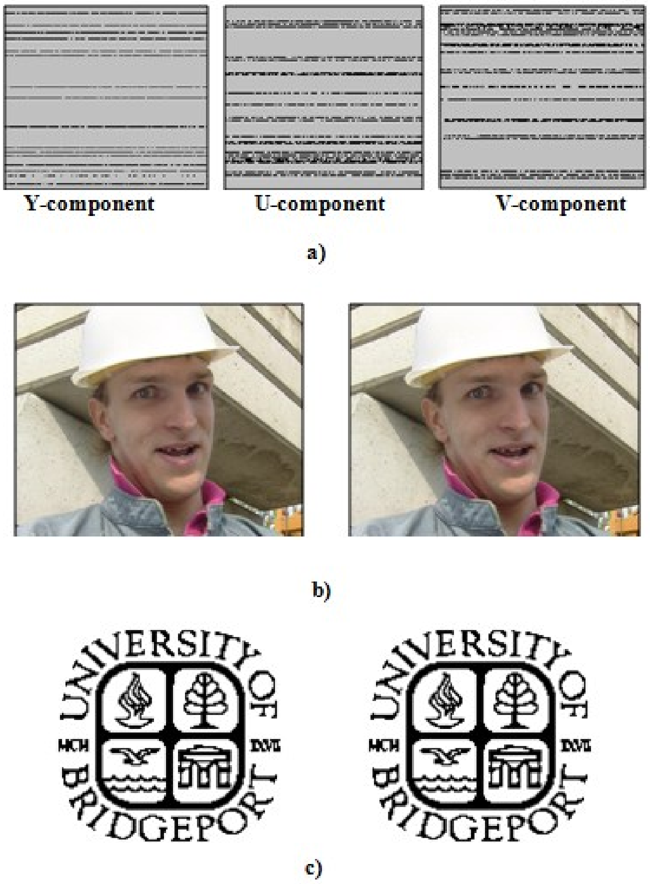


Figure 3: A sample result of frame number 111 for the Foreman video. a) Shows the selected areas for embedding in YUV components for frame number 111. b) Shows the 111th frame both the original and the stego frames. c) Shows the embedded and extracted message.

Figure 4 shows an example of frame no. 222 in the Akiyo video. The first part of the figure shows the separating of the YUV video components and also shows the areas that have been selected for embedding the secret message. The selected areas in this frame are different from the selected areas in other frames in the same video, which are chosen randomly by the private key. This gives the system more security and robustness against attackers. The second part of the figure shows the original and the stego frames. The third part of the figure shows the hidden message before and after embedding.

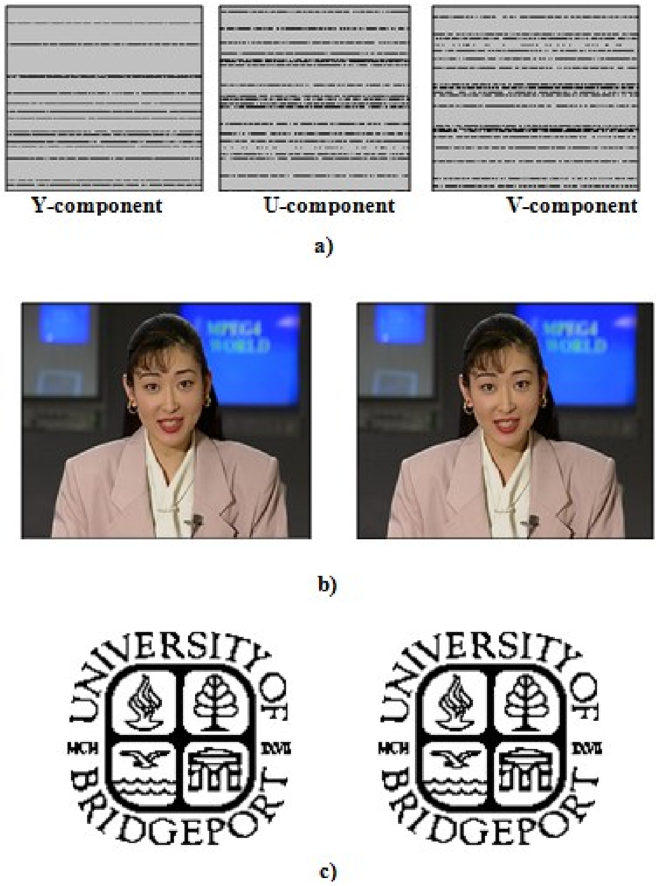


Figure 4: A sample result of frame number 222 for Akiyo video. a) Shows the selected areas for embedding in YUV components for frame number 222. b) Shows the 222nd frame both the original and the stego frames. c) Shows the embedded and extracted message.

In Table I, the average PSNR for all video sequences is shown for each Y, U, and V component and all are greater than 51dB. The quality of the stego videos are mostly the same as the original videos.

Figure 5 shows the PSNRs of 300 stego frames in the Mother-daughter video. The quality of the results that have been obtained from our proposal are very close to the quality of the original videos before embedding. In general, PSNRs are greater than 51 dBs, and the V component has a better quality among the three components.

Figure 6 shows the comparison of visual quality between nine stego videos. The PSNR of each component, Y, U, and V, is calculated, of which the average is 300 frames per video. All the results of PSNRs are between 51 and 52.5 dBs, which are considered very good results with regard to the purpose of quality.

TABLE I
THE AVERAGE PSNR OF Y, U, AND V FOR ALL VIDEO SEQUENCES

Sequences	Frame No.	PSNRY	PSNRU	PSNRV
Foreman	1-100	51.901	51.940	51.920
	101-200	51.857	51.924	52.049
	201-300	51.817	52.059	52.038
Akiyo	1-100	51.881	51.988	52.431
	101-200	51.859	51.978	52.458
	201-300	51.859	51.943	52.428
Coastguard	1-100	51.835	51.664	51.854
	101-200	51.824	51.682	51.806
	201-300	51.823	51.655	51.795
Container	1-100	51.821	52.146	52.067
	101-200	51.806	52.117	52.008
	201-300	51.785	52.056	51.970
Hall	1-100	51.787	52.084	52.021
	101-200	51.797	52.079	52.016
	201-300	51.785	52.063	52.005
Mobile	1-100	51.862	52.127	52.065
	101-200	51.829	52.076	52.074
	201-300	51.834	52.064	52.072
Mother-daughter	1-100	51.686	51.857	51.995
	101-200	51.702	51.868	51.992
	201-300	51.687	51.876	51.946
News	1-100	52.027	52.167	51.781
	101-200	52.012	52.139	51.769
	201-300	51.998	52.135	51.764
Stefan	1-100	51.885	52.082	51.961
	101-200	51.810	52.111	51.964
	201-300	51.848	52.081	51.904

V. CONCLUSIONS

In this paper, a secure video steganography has been proposed based on the Hamming code concepts. The steganography scheme used frames as still images. It divides the video stream into frames and then converts the frames to the YUV format. This algorithm is considered a high embedding efficiency algorithm due to the low modification on the host data that makes the stego videos have a very good quality. The visual quality is measured by the PSNR and all the obtained experimental results have a PSNR above 51 dBs. By having a good visual quality for stego videos, attackers are not likely to be suspicious. Regarding security purposes our algorithm is a secure enough to thwart any endeavor by unauthorized users to retrieve the secret message, even if they are suspicious of a message's existence. Security has been satisfied by having more than one key to embed and extract the secret message. In addition to the three keys that we have used, we also encode and decode the message before and after embedding, which improves the security of our scheme to be even better.

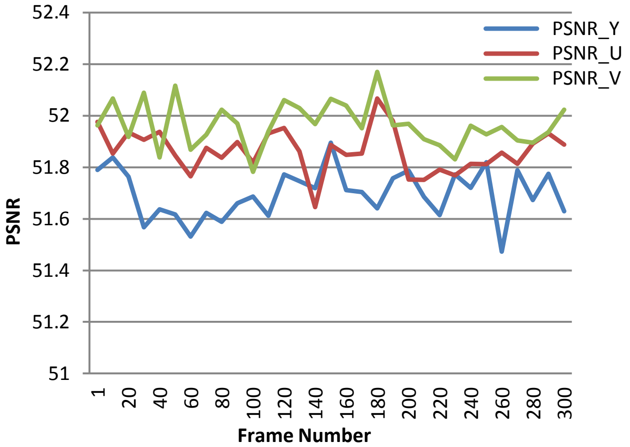


Figure 5: PSNR of 300 stego frames for the Mother-daughter video.

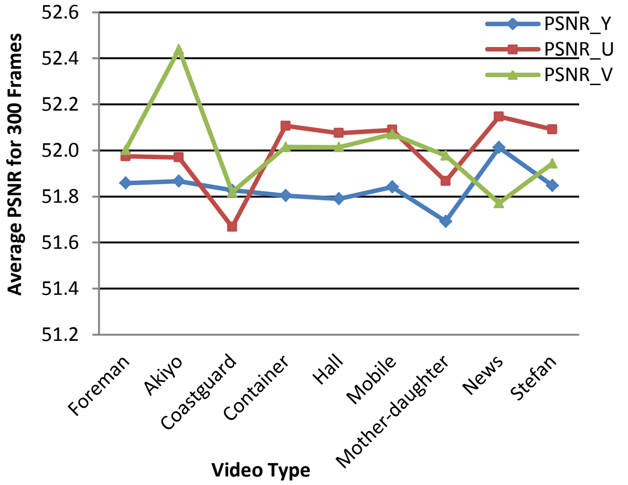


Figure 6: Comparison between the averages of the PSNR Y, U, and V components for nine video sequences

With regard to the embedding payload, in each frame the University of Bridgeport's logo, which has 16 Kbits, has been embedded. We have been used all 300 frames in each video the total number of secret information that we can hide in one video 300 times 16Kbits (300 X 16 Kbits). Our steganography scheme can increase the capacity up to 90 Kbits in each frame with the slight degradation of visual quality. Finally, our proposal have added new parameters in term of security purpose to the encoded message of Hamming encoder (7, 4) to make it more secure before and after embedding process.

References

[1] H. Yuh-Ming and J. Pei-Wun, "Two improved data hiding schemes," in *Image and Signal Processing (CISP), 2011 4th International Congress on*, 2011, pp. 1784-1787.

[2] C. Chin-Chen, T. D. Kieu, and C. Yung-Chen, "A High Payload Steganographic Scheme Based on (7, 4) Hamming Code for Digital Images," in *Electronic Commerce and Security, 2008 International Symposium on*, 2008, pp. 16-21.

[3] L. Guangjie, L. Weiwei, D. Yuewei, and L. Shiguo, "An Adaptive Matrix Embedding for Image Steganography," in *Multimedia Information Networking and Security (MINES), 2011 Third International Conference on*, 2011, pp. 642-646.

[4] W. Jyun-Jie, C. Houshou, L. Chi-Yuan, and Y. Ting-Ya, "An embedding strategy for large payload using convolutional embedding codes," in *ITS Telecommunications (ITST), 2012 12th International Conference on*, 2012, pp. 365-369.

[5] M. E. Eltahir, L. M. Kiah, and B. B. Zaidan, "High Rate Video Streaming Steganography," in *Information Management and Engineering, 2009. ICIME '09. International Conference on*, 2009, pp. 550-553.

[6] P. Feng, X. Li, Y. Xiao-Yuan, and G. Yao, "Video steganography using motion vector and linear block codes," in *Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on*, 2010, pp. 592-595.

[7] B. Hao, L.-Y. Zhao, and W.-D. Zhong, "A novel steganography algorithm based on motion vector and matrix encoding," in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, 2011, pp. 406-409.

[8] Z. Rongyue, V. Sachnev, M. B. Botnan, K. Hyoung Joong, and H. Jun, "An Efficient Embedder for BCH Coding for Steganography," *Information Theory, IEEE Transactions on,* vol. 58, pp. 7272-7279, 2012.

[9] Y. Liu, Z. Li, X. Ma, and J. Liu, "A Robust Data Hiding Algorithm for H. 264/AVC Video Streams," *Journal of Systems and Software,* 2013.

[10] A. Sarkar, U. Madhow, and B. S. Manjunath, "Matrix Embedding With Pseudorandom Coefficient Selection and Error Correction for Robust and Secure Steganography," *Information Forensics and Security, IEEE Transactions on,* vol. 5, pp. 225-239, 2010.

Ramadhan J. Mstafa

Ramadhan Mstafa is originally from Dohuk, Kurdistan Region, Iraq. He is pursuing his Doctorate in Computer Science and Engineering at the University of Bridgeport, Bridgeport, Connecticut, USA. He received his Bachelor's degree in Computer Science from University of Salahaddin, Erbil, Iraq. Mr. Mstafa received his Master's degree in Computer Science from University of Duhok, Duhok, Iraq. His research interests include image processing, mobile communication, security and steganography.

Prof. Khaled M. Elleithy

Dr. Elleithy is the Associate Dean for Graduate Studies in the School of Engineering at the University of Bridgeport. He is a professor of Computer Science and Engineering. He has research interests are in the areas of wireless sensor networks, mobile communications, network security, quantum computing, and formal approaches for design and verification. He has published more than two hundred fifty research papers in international journals and conferences in his areas of expertise.

Dr. Elleithy is the editor or co-editor for 12 books by Springer. He is a member of technical program committees of many international conferences as recognition of his research qualifications. He served as a guest editor for several International Journals. Also, he is the General Chair of the International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering virtual conferences.