IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 11, November 2015.
www.ijiset.com
ISSN 2348 – 7968

Secret Data Hiding In Encrypted Compressed Video Bitstreams For Privacy Info Protection

Haseeba T Badarudeen[1]

[1] Department of Electronics and Communication, MG University
Ernakulam, Kerala, India

***Abstract*—Digital depiction of media supports access and can potentially improve the portability, effectiveness, and accuracy of the information presented. Unwanted effects of facile data access include an increased opportunity for damage of patent and tampering with or without alteration of content. Thus for maintaining content notation and tampering exposure data hiding is performed in the encrypted version of the videos. Here, data hiding directly in the encrypted version of H.264/AVC video stream is approached, which includes the following three parts, i.e., encryption of H.264/AVC video, data embedding of data in the form of image, and extraction of data. The code words of intra prediction modes, the code words of motion vector differences, and the code words of residual coefficients are encrypted using encryption key and data hider may add additional data in the encrypted version by using bit replacement technique. The data to be hidden is in the form of image. Various images are taken and the parameters such as Mean Square Error (MSE), PSNR, and Correlation are evaluated to measure the efficiency of the method.**

Keywords—Data hiding, H.264/AVC video, Bit replacement

I. INTRODUCTION

Recently, internet and digital medium are getting more and more established. So requirement of safe broadcast of data has also enlarged. Various good methods are proposed and already taken into exercise. Data hiding is the procedure of secretly hiding information inside a data source without altering its perceptual value. Data hiding is the art and skill of writing secret messages in such a way that no one apart from the sender and intended recipient even understand there is a something hidden in it.

Thus data hiding represents a group of processes used to add data, such as patent information, into various forms of media such as image, audio, text, or video with a minimum quantity of perceivable degradation to the host signal; i.e., the added data should be unseen and inaudible to a human spectator. Note that data hiding, while similar to compression, is different from encryption. Its goal is not to limit or control access to the host signal, but rather to make sure that embedded data remain inviolate and recoverable [1]. The needs of any data hiding system can be classified into security, capacity and robustness. Data hiding in video stream by text substitution is concerned with video steganography. Data embedding in videos seems comparable to images. Since videos contain more number of pixels or increased number of coefficients in transform domain, a video has more capacity than a still image and more data can be embedded in the video.

II. PROPOSED SCHEME

The proposed scheme includes three parts, encryption of H.264/AVC video, embedding of data and extraction of data. The sender encrypts the original H.264/AVC video stream with encryption keys to produce an encrypted video stream. Then, the data hider can embed the secret data into the encrypted video stream by using codeword substitution method. This can be carried out even without the knowledge of the original video content. At the receiver end, the hidden data extraction can be accomplished either in encrypted or in decrypted version. The diagram of the proposed framework is shown below. Figure 1 depicts the encryption and data embedding, and the Figure 2 depicts data extraction and video decryption.

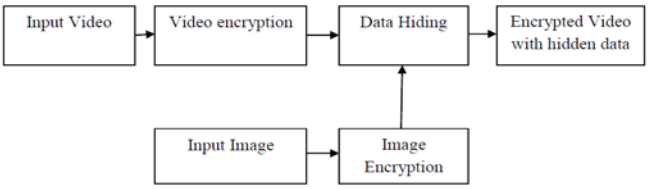

Figure 1: Video encryption and data embedding at the sender end

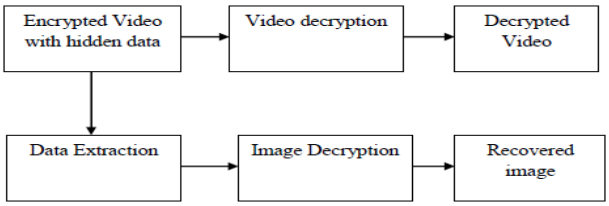

Figure 2: Data extraction and video display at the receiver end

A video file in .avi format is taken which is then H.264 encoded and encrypted, meanwhile a secret image is embedded

ISSN 2348 – 7968

into the encrypted file in encrypted format. The video file is being encrypted to preserve safety and privacy and with the intention of content notation and tampering exposure. In this way, one can maintain the confidentiality by hiding data in encrypted domain without decryption. The image is also encrypted that can also offer better security. At the receiver side the reverse operations are carried out to get back the original video as well as the image. A performance evaluation is made in terms of Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), and Correlation etc by comparing the original image with the retrieved image.

A. Sender Side Algorithm

Step 1: Input the video (.avi) and convert it into frames.
Step 2: Apply H.264 coding to the frames.
Step3: Enter the encryption password and encrypt the H.264 coded frames
Step 4: Select the secret image and encrypt it.
Step 5: Enter data hiding key and embed the data into the Frames.
Step 6: Convert frames to video to get encrypted video with hidden data.

1) Encryption Of H.264/Avc Video Stream: The property of H.264/AVC codec is analysed and three sensitive parts such as Intra Prediction Modes (IPM), Motion Vector Differences (MVD), and residual coefficients are encrypted using stream ciphers [2].

Intra-Prediction Mode (IPM) Encryption: According to H.264/AVC standard, the following four types of intra coding are supported, which are denoted as Intra_4 × 4, Intra_16×16, Intra_chroma, and I_PCM [3]. Here, IPMs in the Intra_16 × 16 blocks are chosen to encrypt. The IPM for Intra_16 × 16 block is specified in the mb_type (macroblock type) field which also specifies other parameters about this block such as coded block pattern (CBP). Thus for Intra_16 × 16 block, the IPM encryption is performed by applying a bitwise XOR operation between the last bit of the codewords and a bit of the pseudorandom sequence to keep the value of CBP and the length of codeword unchanged [4]. The pseudorandom sequence is produced via an encryption key.

Motion Vector Difference (MVD) Encryption: The motion vectors should be encrypted along with the IPM, in order to protect both texture as well as motion information. In H.264/AVC, motion vector prediction is further performed on the motion vectors, which yields MVD. In H.264/AVC baseline profile, Exp-Golomb entropy coding is used to encode MVD. Table 1 shows Exp-Golomb codewords. The last bit of the codeword is encrypted by applying the bitwise XOR operation with a standard stream cipher, which is determined by an encryption key.

Residual Data Encryption: In order to maintain high safety, the residual data in both I-frames and P-frames should be encrypted. A method for encrypting the residual data depending on the features of codeword is given in detail. In H.264/AVC baseline profile, CAVLC entropy coding is used to encode the quantized coefficients of a residual block [5]. Each CAVLC codeword can be expressed as the following format:

{Coeff_ token, Sign _of _Trailing Ones, Level, Total _zeros, Run _before} [2]

TABLE 1: EXP-GOLOMB CODEWORDS

code_num	Codeword
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
...	...

TABLE 2: LEVELS AND CORRESPONDING CODEWORDS [2]

suffixLength	*Level(>0)*	*Codeword*	*Level(<0)*	*Codeword*
0	1	1	-1	01
	2	001	-2	0001
	3	00001	-3	000001
	4	0000001	-4	00000001
1	1	10	-1	11
	2	010	-2	011
	3	0010	-3	0011
	4	00010	-4	00011
	5	000010	-5	000011
	6	0000010	-6	0000011
	7	00000010	-7	00000011
	8	000000010	-8	000000011
2	1	100	-1	101
	2	110	-2	111
	3	0100	-3	0101
	4	0110	-4	0111
	5	00100	-5	00101
	6	00110	-6	00111
	7	000100	-7	000101
	8	000110	-8	000111
	9	0000100	-9	0000101
	10	0000110	-10	0000111
	11	00000100	-11	00000101
	12	00000110	-12	00000111
	13	000000100	-13	000000101
	14	000000110	-14	000000111
3	1	1000	-1	1001
	2	1010	-2	1011
	3	1100	-3	1101
	4	1110	-4	1111
	5	01000	-5	01001
	6	01010	-6	01011
	7	01100	-7	01101
	8	01110	-8	01111
	9	001000	-9	001001
	10	001010	-10	001011
	11	001100	-11	001101
	12	001110	-12	001111
	13	0001000	-13	0001001
	14	0001010	-14	0001011

To keep the bitstream compliant, not all syntax elements can be modified during encryption process. For example, *Coeff_ token, Total _zeros, and Run _before* should remain unchanged

ISSN 2348 – 7968

[6]. Therefore, residual data encryption can be attained by modifying the codewords of *Sign_of_Trailing Ones* and *Level*. The *Sign_of_Trailing Ones* is encoded with a single bit. Bit "0" is assigned for +1 and bit "1"is assigned for −1. The codeword of *Sign_of_Trailing Ones* is encrypted by applying the bitwise XOR operation with an encryption key. The codeword for each Level is made up of a prefix (*level_prefix*) and a suffix (*level_suffix*) as Level

codeword = [*level pref i x*], [*level su f f i x*]

Table 2 shows Levels with different suffix Length and corresponding codewords. The last bit of the codeword is encrypted by applying the bitwise XOR operation with an encryption key.

2) Data Embedding: In the encrypted bitstream of H.264/AVC, the data embedding is carried out by substituting appropriate codewords. It can be seen that there are no corresponding substituted codewords when suffix Length is equal to 0 or 1. Then the codewords of Levels whose suffix Length is 2 or 3 would be divided into two opposite codespaces as shown in Figure 3[2].

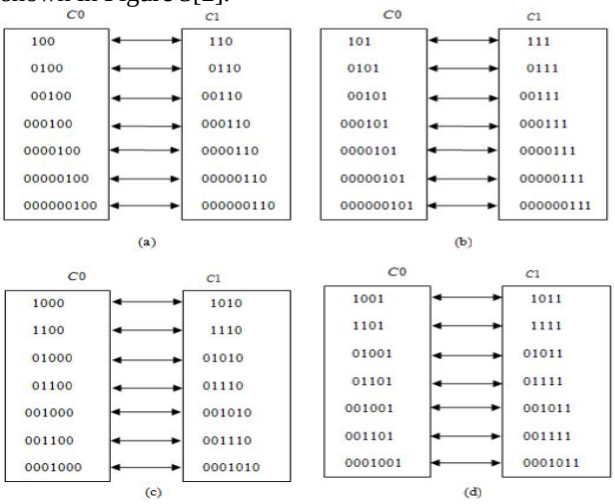

Figure 3: CAVLC codeword mapping [2]

a) Suffix Length = 2 & Level > 0 b) Suffix Length = 2 & Level < 0
b) Suffix Length = 3 & Level > 0 d) Suffix Length = 3 & Level < 0

```
Procedure
    if (data bit==0)
    {
        if (the codeword belongs to C0)
            The codeword is unmodified;
        else if (the codeword belongs to C1)
            The codeword is replaced with the corresponding codeword in C0
        }
    else if (data bit==1)
    {
        if (the codeword belongs to C1)
            The codeword is unmodified;
        else if (the codeword belongs to C0)
            The codeword is replaced with the corresponding codeword in C1
        }
```

Figure 4: The procedure of codeword mapping [2]

Data hiding is performed directly in encrypted bitstream using the following steps.

Step1. In order to enhance the safety, the additional data is encrypted with any pseudo-random sequence $P = \{p(i)|i = 1, 2, \cdots, L, p(i) \in \{0, 1\}\}$ to generate the to-be-embedded sequence $E = \{E(i)|i = 1, 2, \cdots, L, E(i) \in \{0, 1\}\}$. The sequence P is generated using the data hiding key. It is very difficult for anyone who does not possess the data hiding key to recover the additional data [2].

B. Receiver Side Algorithm

Step 1: Load encrypted video with hidden data and convert it into frames
Step 2: Apply H.264 decoding to the frames.Step3: Enter the decryption password and decrypt the H.264 coded frames
Step 4: Enter data hiding key and extract the data (secret image) from the frames.
Step 5: Display the secret image.
Step 6: Convert frames back to the original video

The hidden data can be extracted either in encrypted or decrypted domain. Data extraction process is quick and easy. The extraction in encrypted domain is only discussed in the work. To preserve confidentiality, receiver may only get the data hiding key and have to manipulate data in encrypted domain. In encrypted domain, encrypted video with hidden data is directly sent to the data extraction module, and the extraction process is given as follows.

Step1: The codewords of Levels are firstly identified from the encrypted bitstream.
Step2: If the codeword belongs to first codespace, the extracted data bit is "0". If the codeword belongs to second codespace, the extracted data bit is "1".
Step3: According to the data hiding key, the same pseudo-random sequence P that was used in the embedding process is utilized. Then the extracted bit sequence could be decrypted by using P to get the original data added. Since the whole process is entirely operated in encrypted field, it successfully avoids the outflow of original video content

III. EXPERIMENTAL RESULTS

The proposed scheme is implemented in three videos using four secret images.

ISSN 2348 – 7968

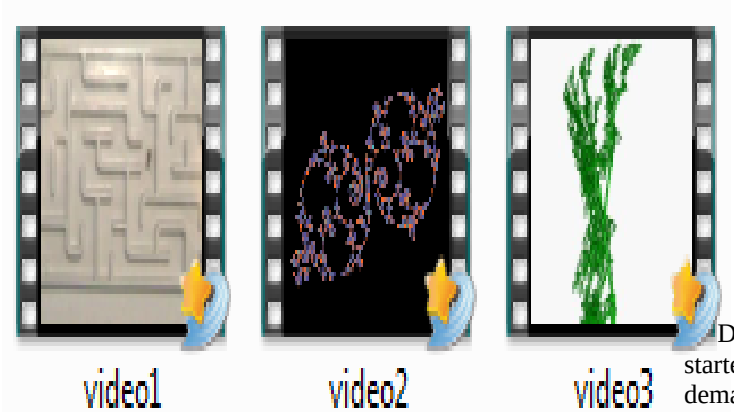

Figure 5: Test videos

Four secret images are hidden in each of these test videos and the performance is evaluated in terms of Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) and Correlation by comparing the original image with the retrieved image.

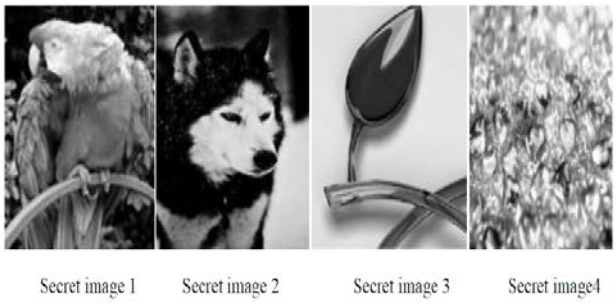

Figure 6: Secret images used

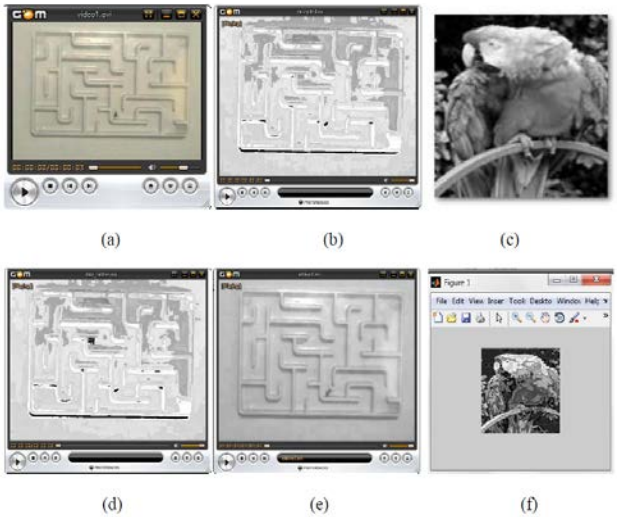

Figure 7: Simulation results (a)Video 1.avi (b) Encrypted video (c) Secret image (d) Encrypted video with secret image (e) Retrieved video (f) Retrieved secret image.

The results are tabulated as shown below in Table 3.

TABLE 3: PERFORMANCE EVALUATION IN VIDEO1.AVI

	MSE	PSNR	Correlation
Secret image1	1.0552e+03	17.8975	0.8732
Secret image2	1.0967e+03	17.7299	0.9467
Secret image3	0.995e+03	18.1511	0.9167
Secret image4	1.0386e+03	17.9663	0.8231

IV. CONCLUSION

Data hiding in encrypted media is a new topic that has started to draw consideration because of the increasing demand for preserving confidentiality. An algorithm to embed secret data in encrypted version of H.264/AVC bitstream is presented in the thesis, which comprises encryption of video, embedding of data and extraction of data. The data hider can add additional data into the encrypted bitstream using bit replacement technique. Since data hiding is performed entirely in the encrypted video, the method can maintain the privacy of the content completely.

Data is taken in the form of image and four secret images are used here. The parameters such as Mean Square Error (MSE), PSNR, and correlation of the original image are compared with the retrieved image. The algorithm provides various advantages such as it can avoid the leakage of video content in mass storage, provides reduced consumption of time, useful to observe video tampering and is a better suitable system for privacy protection. Main application of the scheme is in protection of copyright, medical and surveillance systems and multimedia safety.

References

[1] Arup Kumar Bhaumik,' Data Hiding in Video', International Journal of DatabaseTheory and ApplicationVol. 2, No. 2, June 2009.

[2] Dawen Xu, Data Hiding in Encrypted H.264/AVC Video Streams by Codeword Substitution', IEEE Transactions on Information Forensics and Security, Vol. 9, No. 4, April 2014.

[3] Thomas Wiegand, 'Overview of the H.264/AVC Video Coding Standard', IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, July 2003.

[4] J. G. Jiang, Y. Liu, Z. P. Su, G. Zhang, and S. Xing, "An improved selective encryption for H.264 video based on intra prediction mode scrambling," J. Multimedia, vol. 5, no. 5, pp. 464–472, 2010.

[5] I. E. G. Richardson, H.264 and MPEG-4 Video Compression: Video. Coding for Next Generation Multimedia. Hoboken, NJ, USA: Wiley,2003.

[6] Shahid, M. Chaumont, and W. Puech, "Fast protection of H.264/AVC by selective encryption of CAVLC and CABAC for I and P frames," IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 5, pp. 565–576, May2011.

Haseeba T Badarudeen is currently doing her Masters in Communication Engineering under MG University. She has completed her Batchelor degree in electronics and Communication during academic year of 2009-2013. Her area of interest is in communication field and in image processing.