IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

Data Hiding in Encrypted HEVC/AVC Video Streams

Saltanat Shaikh[1], Prof. Shahzia Sayyad[2]

Student, Computers, SAKEC, Mumbai, India [1]

Professor, Computers, SAKEC, Mumbai, India [2]

Abstract: Transmission of video over internet is prone to attacks from untrustworthy system and administrators. Hence it is necessary to perform encryption of video content for maintaining security of those contents. Data hiding in encrypted domain without decryption preserves the confidentiality of the content. For authentication and damage detection and covert communication, it is useful to embed secret information in these encrypted videos. A method is proposed where the secret information is embedded directly into encrypted video stream, thereby maintaining confidentiality of video content. The input video is compressed using H.264/AVC (Advance Video Compression) or HEVC (High Efficiency Video Coding encoder). This compressed video is converted into frames for the purpose of data hiding using codeword substitution and those encoded data hidden video is encrypted using DES (Data encryption standard) and random generated password. The codewords of residual coefficients, motion vector differences and intra-prediction modes are encrypted using a stream cipher. Encrypted video along with encrypted file which contain password and frame number is send for storage or to receiver for decryption.

Keywords: HEVC/AVC Video, Data hiding, H.264/AVC (Advance Video Compression), random generated password

I. INTRODUCTION

With the ever growing popular usage of internet, multimedia data that is transmitted over the internet is prone to attacks from intruders and untrustworthy system administrators. Thus it is necessary to safeguard the data by performing encryption over the multimedia data. Video finds its applications in many fields like medical surveillance, military etc. For protecting the ownership rights, covert communication and content notation, it is necessary to embed secret information into this video. In medical videos or surveillance videos, in order to protect the privacy of the people, the videos are encrypted. It is then required that a database manager embed personal information like name, age, account number, password of the person directly into the encrypted video. This act of performing data hiding directly into encrypted video streams avoids the leakage of the video content, since the data hider can embed the additional information into video without knowing the content of the video.

Cloud computing has become an important technology trend, which can provide highly efficient computation and large-scale storage solution for video data. Given that cloud services may attract more attacks and are vulnerable to untrustworthy system administrators, it is desired that the video content is accessible in encrypted form. For example, a cloud server can embed the additional information (e.g., video notation, or authentication data) into an encrypted version of a video by using data hiding technique. With the hidden information, the server can manage the video or verify its integrity without knowing the original content, and thus the security and privacy can be protected. In addition to cloud computing, this technology can also be applied to other prominent application scenarios. Till now, few successful data hiding schemes in the encrypted domain have been reported in the open literature. With the increasing demands of providing video data security and privacy protection, data hiding in encrypted videos will undoubtedly become popular in the near future. Obviously, due to the constraint of the underlying encryption, it is very difficult and sometimes impossible to transplant the existing data hiding algorithms to the encrypted domain. The proposed scheme can achieve excellent performance in the following three different prospects.

- The data hiding is performed directly in compressed video and encrypts those video bit stream using password. Transmit encrypted password and encrypted encoded video stream for decryption purpose.
- This scheme can ensure both the format compliance and the strict file size preservation.
- This scheme can be applied to two different application scenarios by extracting the hidden data either from the encrypted video stream or from the decrypted video stream.

In this work, the video is compressed using, H.264/AVC, HEVC compression format, which is the most widely used video compression format in Blu-ray, DVDs and for transmission of video streams over internet. It is found from the Video baseline profile that the sensitive information of video is contained in 3 parts- Intra-prediction mode (IPMs), motion vector differences (MVDs) and residual coefficients. This poses few challenges as we should find out where and how to modify the bit stream So that the encrypted video with the hidden

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

data is still a compliant compressed bit stream. Secondly, it must ensure that the decrypted videos containing hidden data will still appear to be of high visual fidelity. Thirdly the file size after encryption and information embedding must be maintained [1]. Encryption of video is necessary to protect video from hacking, editing, stealing, cropping, prone to noise etc. Video is rich in contents used for uploading and downloading in new era. Data like summarization, date, time, serial number, categories, personal information which is used by server to manage, analyze, compute, clustering, assure integrity etc.

II. LITERATURE SURVEY

As video file consist of several image sequence, so considering the data hiding technique of image will also apply for video data hiding. The most widely used technique to hide data is the usage of the LSB (Least-significant bit modifications) [12]. Although there are several disadvantages to this approach, the relative easiness to implement it, makes it a popular method. To hide a secret message inside an image, a proper cover image is needed. Because this method uses bits of each pixel in the image, it is necessary to use a lossless compression format, otherwise the hidden information will get lost in the transformations of a lossy compression algorithm. When using a 24 bit color image, a bit of each of the red, green and blue color components can be used, so a total of 3 bits can be stored in each pixel. Thus, a 800 × 600 pixel image can contain a total amount of 1.440.000 bits (180.000 bytes) of secret data.
For example, the following grid can be considered as 3 pixels of a 24 bit color image, using 9 bytes of memory:

(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

When the character A, which binary value equals 10000001, is inserted, the following grid results:

(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)

In this case, only three bits needed to be changed to insert the character successfully. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximal cover size. The resulting changes that are made to the least significant bits are too small to be recognized by the human eye, so the message is effectively hidden. While using a 24 bit image gives a relatively large amount of space to hide messages, it is also possible to use an 8 bit image as a cover source. Because of the smaller space and different properties, 8 bit images require a more careful approach. Where 24 bit images use three bytes to represent a pixel, an 8 bit image uses only one. Changing the LSB of that byte will result in a visible change of color, as another color in the available palette will be displayed. Therefore, the cover image needs to be selected more carefully and preferably be in grayscale, as the human eye will not detect the difference between different gray values as easy as with different colors.
Masking and filtering Masking and filtering techniques [11] usually restricted to 24 bits or grayscale images, take a different approach to hiding a message. These methods are effectively similar to paper watermarks, creating markings in an image. This can be achieved for example by modifying the luminance of parts of the image. While masking does change the visible Properties of an image, it can be done in such a way that the human eye will not notice the anomalies. Since masking uses visible aspects of the image, it is more robust than LSB modification with respect to compression, cropping and different kinds of image processing. The information is not hidden at the ”noise” level but is inside the visible part of the image, which makes it more suitable than LSB modifications in case a lossy compression algorithm like JPEG is being used.
A more complex way of hiding a secret inside an image comes with the use and modifications of discrete cosine transformations [10]. Discrete cosine transformations (DST)), are used by the JPEG compression algorithm to transform successive 8 x 8 pixel blocks of the image, into 64 DCT coefficients each. Each DCT coefficient F(u, v) of an 8 x 8 block of image pixels f(x, y) is given by:

$$F(u,v) = \frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y) * cos\frac{(2x+1)u\pi}{16} cos\frac{(2y+1)v\pi}{16}\right]$$

where $C(x) = 1/\sqrt{2}$ when x equals 0 and $C(x) = 1$ otherwise. After calculating the coefficients, the following quantizing operation is performed:

$$F^{Q}(u,v) = \left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor$$

Where Q(u, v) is a 64-element quantization table. A simple pseudo-code algorithm to hide a message inside a JPEG image could look like this:
Input: Message, cover image
Output: Steganographic image containing message
While data left to embed do
Get next DCT coefficient from cover image
If DCT not equal to 0 and DCT not equal to 1 then
Get next LSB from message
Replace DCT LSB with message bit
End if
Insert DCT into Steganographic image
End while
Although a modification of a single DCT will affect all 64 image pixels, the LSB of the quantized DCT coefficient can be used to hide information. Lossless compressed images will be suspect able to visual alterations when the LSB are modified. This is not the case with the above described method, as it takes place in the frequency domain inside the image, instead of the spatial domain and therefore there will be no visible changes to the cover image [3]. When information is hidden inside video the

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

program or person hiding the information will usually use the DCT (Discrete Cosine Transform) method. DCT works by slightly changing the each of the images in the video, only so much though so it's isn't noticeable by the human eye. To be more precise about how DCT works, DCT alters values of certain parts of the images, it usually rounds them up. For example if part of an image has a value of 6.667 it will round it up to 7. Data Hiding in Videos is similar to that of Data Hiding in Images, apart from information is hidden in each frame of the video. When only a small amount of information is hidden inside of video it generally isn't noticeable at all, however the more information that is hidden the more noticeable it will become.

In [3,6], Watermarking scheme using Paillier cryptosystems is carried in encrypted image domain. A new method of secure digital watermarking detection protocol is proposed in this paper. Methodology applies to the protection of the digital contents against illegal use. Based upon the principle of methodological induction, an improvement of protecting copyright contents has been achieved by means of allowing watermark verifier to detect the embedded information with no secret information exposed in extraction process. Where demerit they faced is Encryption of image results in high overhead in storage and computation.

A robust watermarking algorithm is proposed to insert watermark directly into compressed and encrypted JPEG2000 images [7]. Digital watermarking, which has been proven effective for protecting digital data, has recently gained considerable research interest. This study aims to develop an enhanced technique for producing watermarked images with high invisibility. During extraction, watermarks can be successfully extracted without the need for the original image. We have developed discrete wavelet transform with a Haar filter to embed a binary watermark image in selected coefficient blocks. A probabilistic neural network is used to extract the watermark image but the disadvantage is Very less or no reports on data embedding in compressed video streams are present.

Presented Encryption and watermarking. Commutative Encryption and Watermarking (CEW) [8] combines encryption and watermarking to provide a comprehensive security protection for multimedia data. On the other hand, the storage and transmission is an important dilemma due to enormous size of multimedia data, therefore an effective solution to both bandwidth and storage problem is the use of data compression. In this paper, we propose a commutative watermarking and encryption (CEW) scheme for jpeg2000 compression standard. The commutative property of the proposed scheme allow to encrypt a watermarked image without interfering with the embedded watermark, or to watermark encrypted image and still allowing a perfect decrypting. The encryption and watermarking are carried out after the quantization step in jpeg2000 compression standard. It is based on decomposed data after quantization into two parts. One part is encrypted and the other is watermarked. The security of the scheme is assured since the encryption and watermarking is implemented in compressed domain. Experimental Results show the effectiveness of the proposed scheme. Luminance block, sign bits of texture and sign bits of MVDs are encrypted, IPM is used for watermarking. Watermarked bit stream is not fully complaint, so the standard decoder may crash as it cannot parse watermarked bit streams.

In [1], Data hiding is one of the important techniques in the emerging world for reducing the increased attacks. It is also known as data encapsulation or information hiding and is mainly used for hiding internal object details. In order to maintain security and privacy, digital video sometimes needs to be stored and processed in an encrypted format. The performance of data hiding in these encrypted videos is very necessary for the purpose of content notation or tampering detection. Without decryption, data hiding in these encrypted videos will protect the confidentiality of the content. The capacity for performing the data hiding directly in these encrypted H.264/AVC video stream can eliminate the leakage of video content and can help the privacy concerns with cloud computing. In this paper, it proposes a new method to embed secret data directly in the encrypted H.264/AVC bit stream.

III.PROPOSED SYSTEM

Compression of video can be done by any compression standards such as H.264, HEVC etc using Brorsoft Video Converter, WinX HD Video Converter Deluxe, any video convertor etc. DivX player is used to play HEVC, H.264 video contents. Existing project on Data hiding in H.264 are already available. Since HEVC are latest compression video standard came into picture. HEVC is better in performance and give same quality as H.264 with smaller size than H.264.

Compressed video is converted into frames and Data like text, doc file containing images etc are encoded into frames. Data is hiding into contents using codeword substitution. Those encoded video is encrypted using DES and symmetric key/ randomly generated password. This password and frame number is transmitted to receiver by encrypting using Act on File software.

High Efficiency Video Coding:

High Efficiency Video Coding (HEVC) is currently being prepared as the newest video coding standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group [14]. The main goal of the HEVC standardization effort is to enable significantly improved compression performance relative to existing standards in the range of 50% bit-rate reduction for equal perceptual video quality. This paper provides an overview of the technical features and characteristics of the HEVC standard. High Efficiency Video Coding (HEVC), also known as H.265, is a video compression standard, one of several potential successors to the widely used AVC (H.264). In comparison to AVC, HEVC offers about double the data compression ratio at the same level

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

of video quality, or substantially improved video quality at the same bit rate. It supports resolutions up to 8192×4320, including 8K UHD.

In most ways, HEVC is an extension of the concepts in H.264/MPEG-4 AVC. These redundant areas are then replaced with a short description instead of the original pixels. The primary changes for HEVC include the expansion of the pattern comparison and difference-coding areas from 16×16 pixel to sizes up to 64×64, improved variable-block-size segmentation, improved "intra" prediction within the same picture, improved motion vector prediction and motion region merging, improved motion compensation filtering, and an additional filtering step called sample-adaptive offset filtering. Effective use of these improvements requires much more signal processing capability for compressing the video, but has less impact on the amount of computation needed for decompression.

HEVC has new feature which is QuadTree Structure. HEVC is not using fixed macroblock size as earlier standards anymore. Instead of macroblock, the term Coding Tree Unit (CTU or just CU) is used. CTU consists of luma Coding Tree Block (CTB), Chroma CTBs and syntax. For luma component, the CTB can be of size 16x16, 32x32 or even 64x64. These CTBs can then be further divided into smaller blocks called Coding Blocks (CB).HEVC uses both intra-prediction and inter-prediction modes. It introduce deblocking filter and Sample Adaptive Offset (SAO) filter. HEVC is designed to use Context Adaptive Binary Arithmetic Coding for entropy coding (CABAC). Higher efficiency usually comes with a cost complexity H.265 is far more difficult to encode as a result of its complexity. The freeware Video LAN player is currently your best bet, but support will be native to PCs with the release of Windows 10.

Working of HEVC:

Like H.264 and MPEG-2, HEVC uses three frame types, I-, B- and P-frames within a group of pictures, incorporating elements of both inter-frame and intraframe compression. HEVC incorporates numerous advances, including: Coding Tree Blocks: Where H.264 used macroblocks with a maximum size of 16x16, HEVC uses coding tree blocks, or CTBs, with a maximum size of 64x64 pixels. Larger block sizes are more efficient when encoding larger frame sizes, like 4K resolution.

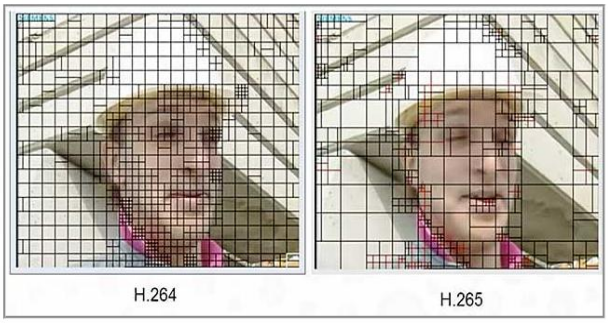

Figure 1: Larger blocks sizes enhance encoding efficiency [14].

More intra-prediction directions: Where H.264 used 9 intra prediction directions; HEVC can use over 35, adding more potential reference pixel blocks that fuel more efficient intra-frame compression. The obvious cost is the additional encoding time required to search in the additional directions.

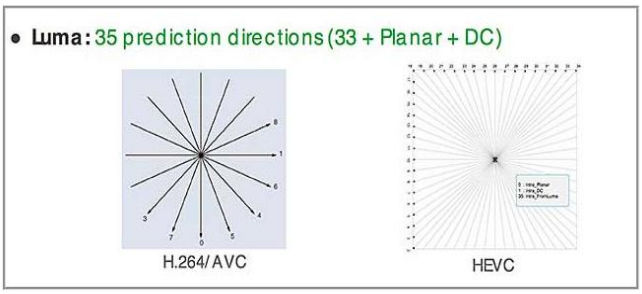

Figure 2: Searching is expanded to find more reference pixel blocks [14].

Other advances include.

- Adaptive Motion Vector Prediction, which allows the codec to find more inter-frame redundancies
- Superior parallelization tools, including Wave front parallel processing, for more efficient encoding in a multi-core environment
- Entropy coding is CABAC only, no more CAVLC
- Improvements to the deblocking filter and the creation of a second filter called Sample Adaptive Offset that further limits artifacts along block edges

IV. PROJECT DESIGN

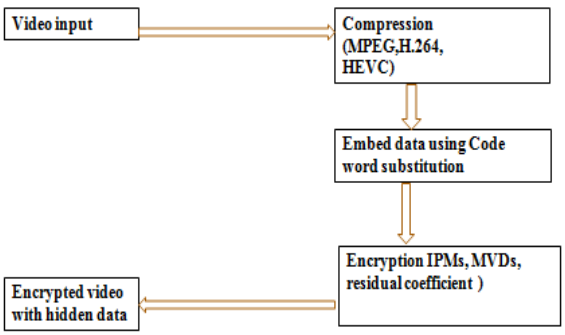

Figure 3: Encryption of video from sender end.

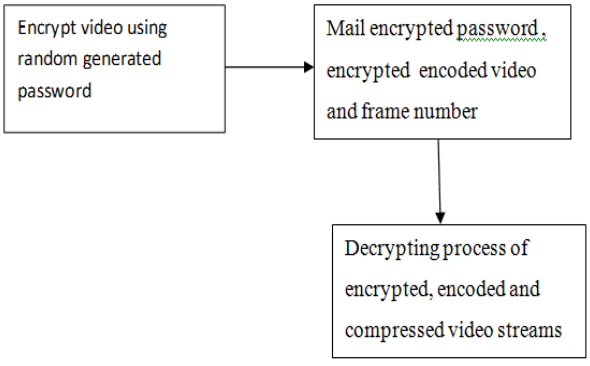

Figure 4: Transmission Process.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

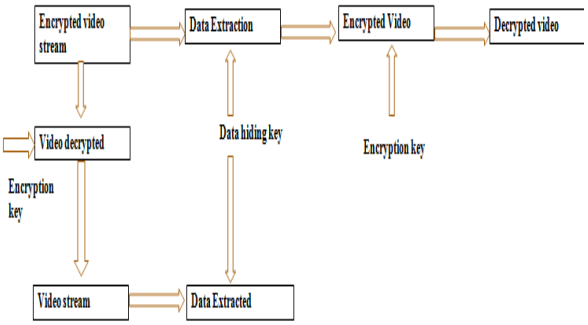

Figure 5: Decryption of video in receiver end.

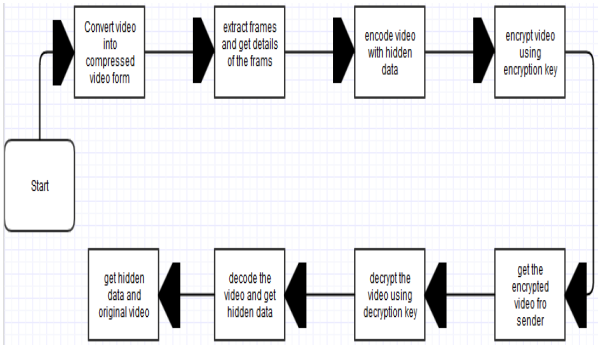

Figure 6: Flow diagram

V. RESULTS AND ANALYSIS

Analysis of different size and different space of video are shown below. Videos are compressed using software like brorsoft, any video converter, WinX HD video converter deluxe etc. Video player DivX Player is used to play this compressed video.

TABLE I
ANALYSIS OF DIFFERENT VIDEO BEFORE AND AFTER COMPRESSION

	Analysis of Videos		
	Video I	Video II	Video III
Original video size, Type of files, Length, Frame height, Data rates, Total bitrates, Frame rates	1.93 MB (2,027,543 bytes), MP4, 00.00.01, 1280, 720, 8028kbps, 8101kbps, 29 frames/sec	1MB, MP4, 00.00.06 640, 368, 994kbps, 1403kbps, 25 frames/sec.	3.83MB, MP4, 00.00.01, 2048, 858, 31818kbps, 32149kbps, 23frames/sec.
H.264/AVC video size, Type of files, Length, Frame height, Data rates, Total bitrates, Frame rates	3.27 MB (3,436,469 bytes), MP4, 00.00.01, 1920, 1080, 13458kbps 13734kbps 30 frames/sec	10.7 MB (11,287,189 bytes),MP4 00.00.06, 1920, 1080, 14696kbps, 15040kbps 30 frames/sec	2.25 MB (2,362,030 bytes), 00.00.01, 2048, 858, 18774kbps, 18878kbps, 29 frames/sec
Encrypted and decrypted of AVC video size, Type of files, Length, Frame height, Data rates, Total bitrates, Frame rates	58MB (258,554,216 bytes) MOV files, 00.00.02, 1920, 1080, 13458kbps 13734kbps 30 frames/sec	92MB, MOV files 00.00.08, 1920, 1080, 18696kbps, 18040kbps 20 frames/sec	56.4MB MOV file, 00.00.02 2048, 864, 157901kbps, 157901kbps, 10frames/sec Hidden data 104KB
HEVC?H.265 video size, Type of files, Length, Frame height, Data rates, Total bitrates, Frame rates	1.32MB, MP4, 00.00.01, 1920, 1080, 5267kbps, 5544kbps, 30 frames/sec	660KB, MP4, 00.00.06, 640, 360, 786kbps, 889kbps 29 frames/sec	593KB, MP4, 00.00.01, 2016, 858, 4727kbps, 4831kbps, 29 frames/sec
Encrypted and decrypted of HEVC video size, Type of files, Length, Frame height, Data rates, Total bitrates, Frame rates	20.32MB, MOV, 00.00.01, 1920, 1080, 5267kbps, 5544kbps, 30 frames/sec	800KB, MOV, 00.00.06, 640, 360, 786kbps, 889kbps 29 frames/sec	900KB, MOV, 00.00.01, 2016, 858, 4727kbps, 4831kbps, 29 frames/sec

File size and other features are maintained before and after encryption of video stream in both the case of HEVC and AVC video quality of HEVC is same as AVC with comparatively less size.

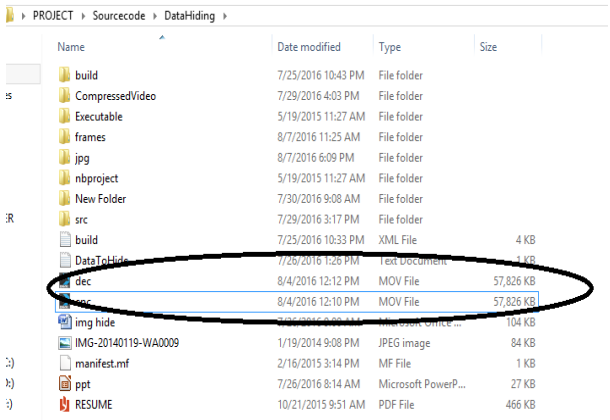

Figure 7: Size of video is maintained after encryption and decryption

DOI 10.17148/IJARCCE.2016.5811

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

VI. CONCLUSION

The existing system just addresses the calculation of size and lacks in addressing the data encryption & data embedding process. Time consumption rate is also high when compared to the recent methods developed. The block encryption methods are not robust to noise. Data hiding in encrypted media is a new topic that has started to draw attention because of the privacy-preserving requirements from cloud data management. An algorithm is used to embed additional data in encrypted video bit stream, which consists of video encryption, data embedding and data extraction phases. The data-hider can embed additional data into the encrypted bit stream using codeword substitution, even though he does not know the original video content. Since data hiding is completed entirely in the encrypted domain, here we can preserve the confidentiality of the content completely. Privacy preserving for encrypted media is new topic for growing research field. This technique facilities better way for data hiding directly in the encrypted domain without decryption of the content thus preserves confidentiality of the content. HEVC can efficiently improve video quality and limit the space.

ACKNOWLEDGEMENT

This project would not have possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I would like to express my special thanks to the Management of Shah and Anchor Kutchhi Engineering College. I would also like to thank our Principal **Dr. Vinit Kotak**, Head of the Department **Prof. Uday Bhave**, who guided us in proper direction. I sincerely thanks my guide **Prof. Vidyullata Devmane** and my co-guide **Prof. Shahzia Sayyad,** for giving me the opportunity to perform this project .I am highly indebted to their guidance, constant supervision and the necessary information at all time. Their willingness to motivate me contributed tremendously to my dissertation work.

Without helps of the particular that mentioned above, I would face many difficulties while doing this. The guidance and support received from all the members who contributed and who are contributing to this seminar, was vital for the success of the dissertation report.

Last but not the least I am greatly indebted to my devoted family members for their support.

REFERENCES

[1] Dawen Xu, Rangding Wang, and Yun Q. Shi, Fellow, “Data Hiding in Encrypted H.264/AVC Video Streams by Codeword Substitution” IEEE 2015.B.

[2] Zhao, W. D. Kou, and H. Li, "Effective watermarking scheme in the encrypted domain for buyer-seller watermarking protocol," Inf. Sci., vol. 180, no. 23, pp. 4672-4684, 2010.

[3] W. Puech, M. Chaumont, and O. Strauss, "A reversible data hiding method for encrypted images,"Proc. SPIE, vol. 6819, pp. 68191E-1-68191E-9, Jan. 2008.

[4] X. P. Zhang, "Reversible data hiding in encrypted image," IEEE Signal Process. Lett. vol. 18, no. 4, pp. 255-258, Apr. 2011.

[5] W. Hong, T. S. Chen, and H. Y. Wu, "An improved reversible data hiding in encrypted images using side match," IEEE Signal Process.

[6] Lett., vol. 19, no. 4, pp. 199-202, Apr. 2012.K. D. Ma, W. M. Zhang, X. F. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," IEEE Trans. Inf. Forensics Security, vol. 8, no. 3, pp. 553-562, Mar.2013.

[7] A. V. Subramanyam, S. Emmanuel, and M. S. Kankanhalli, "Robust watermarking of compressed and encrypted JPEG2000 images," IEEE Trans. Multimedia, vol. 14, no. 3, pp. 703-716, Jun. 2012.S.

[8] G. Lian, Z. X. Liu, and Z. Ren, "Commutative encryption and watermarking in video compression," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 6, pp. 774-778, Jun. 2007.

[9] S. W. Park and S. U. Shin, "Combined scheme of encryption and watermarking in H.264/scalable video coding (SVC)," New Directions Intell. Interact. Multimedia, vol. 142, no. 1, pp. 351-361, 2008.

[10] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560-576, Jul. 2003.

[11] T. Stutz and A. Uhl, "A survey of H.264 AVC/SVC encryp-tion," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 3, pp. 325-339, Mar. 2012.

[12] Arup Kumar Bhaumik1, Minkyu Choi2, Rosslin J.Robles3, and Maricel O.Balitanas4 International Journal of Database Theory and Application Vol. 2, No. 2, June 2009 9 Data Hiding in Video.

[13] A.S. Korde, 2A.P.Hatkar secured data hiding in encrypted video stream (2015).

[14] Gary J. Sullivan, Fellow, IEEE, Jens-Rainer Ohm, Member, IEEE, Woo-Jin Han, Member, IEEE, and Thomas Wiegand, Fellow, IEEE Overview of the High Efficiency Video Coding (HEVC) Standard