Secured Data Transmission in Compressed Encrypted Video Stream

Sadaf .A. Mulani
1Electronics, Shah and Anchor Kuttchi Engineering College, Mumbai, 400088, India

Abstract - **In today's life Cloud computing becomes an important technology trend. Data hiding techniques can be used to embed a secret message and secret image into a video bit stream for copyright protection, access control and transaction tracking. They are some data hiding techniques to assess the quality of video in the absence of the original reference. 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. It can have the following three parts: H.264/AVC video encryption, data embedding, and data extraction. The, data hider may embed additional data in the encrypted domain by using codeword substitution technique, without knowing the original video content. In order to adapt to different application scenarios, data extraction can be done either in the encrypted domain or in the decrypted domain. Here, the data extraction can be done either in the encrypted or in the decrypted domain in order to adapt to different application scenarios. Also, video file size is strictly preserved even after encryption and data embedding. Furthermore, to avoid the occupancy of more memory, enhance capacity of storage devices, to reduce the time for the transmission of videos and to reduce computation arithmetic compression technique is done to achieve high compression ratio. Thus, utilizing minimum bandwidth during transmission. Experimental results have demonstrated the feasibility and efficiency of the proposed scheme.**

Index Terms - Data hiding, encrypted domain, H.264/AVC ,codeword substituting.

1. INTRODUCTION

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. The capability of performing data hiding directly in encrypted H.264/AVC video streams would avoid the leakage of video content, which can help address the security and privacy concerns with cloud computing [1]. For example, a cloud server can embed the additional information (e.g., video notation, or authentication data) into an encrypted version of an H.264/AVC 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. For example, when medical videos or surveillance videos have been encrypted for protecting the privacy of the people, a database manager may embed the personal information into the corresponding encrypted videos to provide the data management capabilities in the encrypted domain.

2. LITERATURE SURVEY

Till now, few successful data hiding schemes in the encrypted domain have been reported in the open literature.

Data Hiding Technique the capacity is not high enough to embed the large video. In previous proposed approach computation and large storage for video data is attracted by more untrustworthy administrators. The direct performing of data hiding in H.264/AVC will avoid leakage of video content which will provide security for technique [1]. Data hiding can be performed by using FFMPEG, Steganography, Visual Cryptographic Scheme, Invisible Watermarking and Base 64 Encoder/Decoder technique the security and video payload can be increased.

In RDH histogram shift, in which space is saved for data embedding and shifting the bins of histogram to the gray valves which help us to convert the image into the binary image [2]. Binary image will help to hide the data into the image easily. The recursive binary code is constructed to achieve rate distortion between the data compression and binary covers.

To avoid the loss and errors while encryption and decryption separate memory is allocated to recover the original image[3]. At the receiver side the data extraction and image recovery is restored with the original key content to restore the image. If the original key content is received, the image will be restored but the extract the exact data. So the image is converting to gray scale and then binary image to get the exact data in decryption.

For example [4] fingerprints and faces are obtained by outsiders, the biometric templates misuses them for its own purposes. This type of biometric threats exists which had become difficult to prevent unauthorized parties to encrypt the video. To prevent the unauthorized parties to encrypt the video the base 64 encoder and decoder algorithm for codeword substitution is used to prevent the security and transfer through secured system.

Development of computer technology, Internet technology and multimedia data uses images; videos and audios are encrypted using various algorithms such as DES, RSA, IDEA or AES for text or binary data [5]. These algorithms are difficult to use them in video encryption, large volumes and recompression. Data protection or content protection in encrypted scheme is secure when the cost for breaking is no smaller than one paid for its authorization.

Another method is based on difference expansion (DE) for vacating room in encrypted image in which the difference of each pixel group is expanded, e.g., multiplied by 2, and thus least significance bit(LSBs) of the difference are all-zero and the space created can be used for embedded data.

3. PROBLEM STATEMENTS

It becomes highly desirable to develop data hiding algorithms that work entirely on encoded bitstream in the encrypted domain However; there are some significant challenges for data hiding directly in compressed and encrypted bitstream.

- The first challenge is to determine where and how the bitstream can be modified so that the encrypted bitstream with hidden data is still a compliant compressed bitstream.
- The second challenge is to insure that decrypted videos containing hidden data can still appear to be of high visual fidelity.
- The third challenge is to maintain the file size after encryption and data hiding, which requires that the impact on compression gain is minimal.
- The fourth challenge is that the hidden data can be extracted either from the encrypted video stream or from the decrypted video stream, which is much more applicable in practical applications.
- The fifth challenge is that to reduce the video size making it convenient for cloud storage.

4. PROPOSED SCHEME

In this section, a novel scheme of data hiding in the encrypted version of H.264/AVC videos is presented, which includes four parts i.e., Frame selection, H.264/AVC video encryption, data embedding and data extraction. By analyzing video codec, data hider may embed data with image in the encrypted domain by using codeword substitution technique, without acquaintance of original data. In order to change to different application, data extraction can be done in the encrypted and decrypted domain. Furthermore arithmetic compression is done for efficient transmission of data utilizing minimum bandwidth. Also, the file size of video is strictly protected even after encryption and data embedding. Fig. 1, where the encryption and data embedding are depicted in part (a), and the data extraction and video decryption are shown in part (b).

BLOCK DIAGRAM

Sender **Receiver**

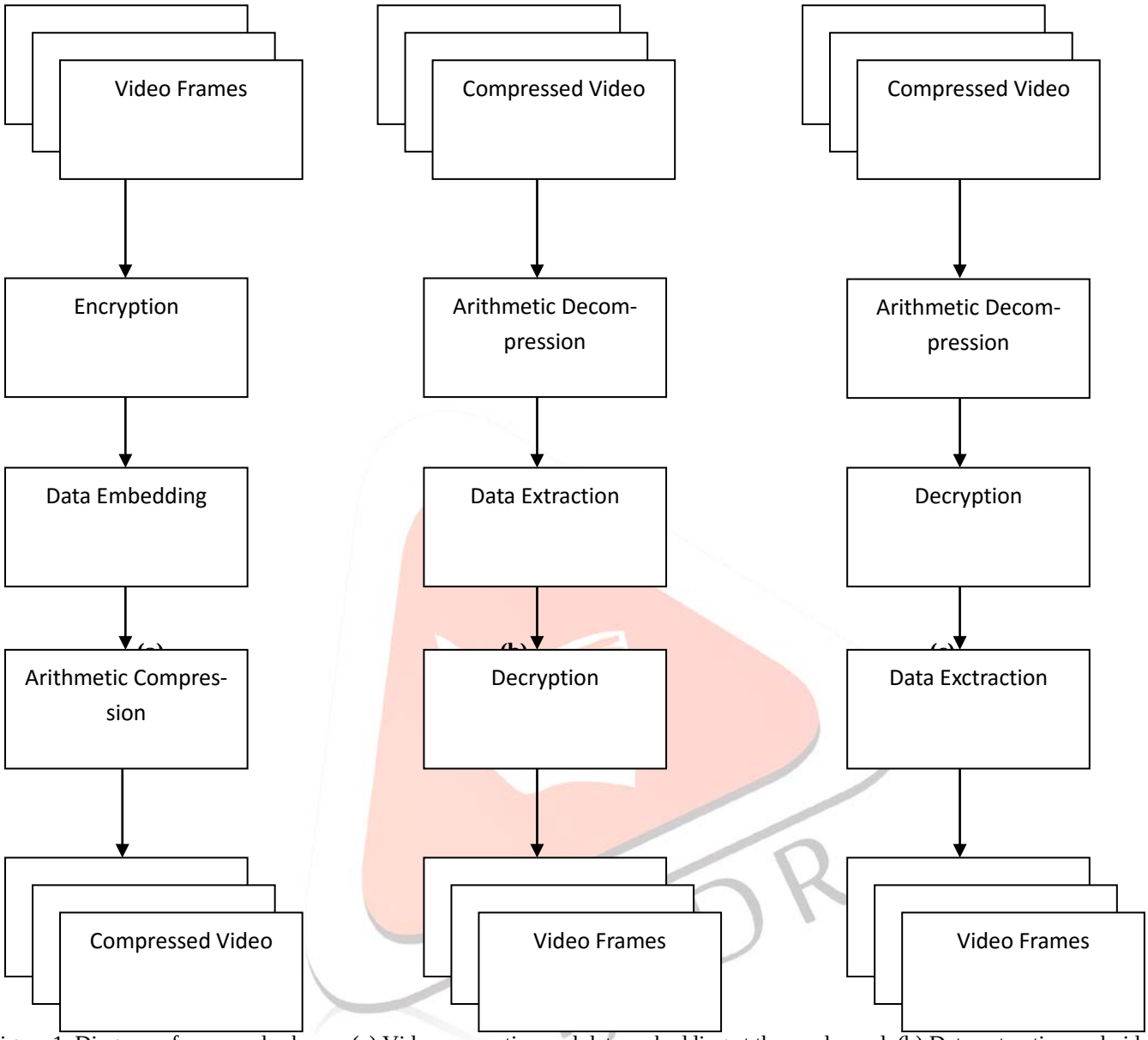

Figure 1. Diagram of proposed scheme. (a) Video encryption and data embedding at the sender end. (b) Data extraction and video display at the receiver end in two scenarios

A. FRAME SELECTION IN ENCRYPTION

Select the video file to hide the Secret image and Data. Then the video part will be converting into n number of frames. In future these frames are used to hide the image and the data.

B. ENCRYPTION OF VIDEO STREAM:

It is not practical to encrypt the whole compressed video bitstream like what the traditional ciphers do because of the following two constraints, i.e., format compliance and computational cost. Alternatively, only a fraction of video data is encrypted to improve the efficiency while still achieving adequate security. The key issue is then how to select the sensitive data to encrypt. The key issue is then how to select the sensitive data to encrypt. According to the analysis given in [13], it is reasonable to encrypt both spatial information (IPMand residual data) and motion information (MVD) during H.264/AVC encoding.

1.) INTRA-PREDICTION MODE (IPM) ENCRYPTION:

The following four types of intra coding are supported, which are denoted as Intra_4 ×4, Intra_16×16, Intra_chroma, and I_PCM [1]. Here, IPMs in the Intra_4×4 andIntra_16×16 blocks are chosen to encrypt. It is prominent at the length of the encrypted codeword is the same as the original one. For the format compliance in the decoding process, the blocks in the first row and/or in the first column of encrypted IPMs should have the decodable value, since not all modes are available along the top and the left borders of each frame due to the lack of acquaintance. If the IPM after encryption is not available for an entire block, then the IPM encryption of this block will be skipped. This further indicates that IPM encryption is not secure enough in some specific locations and should be used in combination with other encrypt ing method. In summary, IPM encryption implies changing the actual mode to another one without violating the semantics and bit stream

mb_type	Name of mb_type	Intra16x16 PredMode	Chroma CBP	Luma CBP	Codeword
1	I_16x16_0_0_0	0	0	0	010
2	I_16x16_1_0_0	1	0	0	011
3	I_16x16_2_0_0	2	0	0	00100
4	I_16x16_3_0_0	3	0	0	00101
5	I_16x16_0_1_0	0	1	0	00110
6	I_16x16_1_1_0	1	1	0	00111
7	I_16x16_2_1_0	2	1	0	0001000
8	I_16x16_3_1_0	3	1	0	0001001
9	I_16x16_0_2_0	0	2	0	0001010
10	I_16x16_1_2_0	1	2	0	0001011
11	I_16x16_2_2_0	2	2	0	0001100
12	I_16x16_3_2_0	3	2	0	0001101
13	I_16x16_0_0_1	0	0	15	0001110
14	I_16x16_1_0_1	1	0	15	0001111
15	I_16x16_2_0_1	2	0	15	000010000
16	I_16x16_3_0_1	3	0	15	000010001
17	I_16x16_0_1_1	0	1	15	000010010
18	I_16x16_1_1_1	1	1	15	000010011
19	I_16x16_2_1_1	2	1	15	000010100
20	I_16x16_3_1_1	3	1	15	000010101

Table 1: Macro block Types for I Slices and Variable Length of Codeword in H.264/AVC

2) MOTION VECTOR DIFFERENCE (MVD) ENCRYPTION

In order to protect both texture information and motion information, not only the IPMs but also the motion vectors should be encrypted. 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 [19] is used to encode MVD. Table 2 shows the values of MVDs and corresponding Exp-Golomb codewords. The last bit of the codeword is encrypted by applying the bitwise XOR operation with a standard stream cipher determined by key. The last bit encryption may change the sign of MVD, but does not affect the length of the codeword and satisfies the format compliance [2]. That corresponding to "2" and "−2" are "00100" and "00101", respectively, which have the same length. It should be verified that when the value of MVD is equal to 0, its corresponding codeword "1" keeps unchanged during the encryption process.

MVD	*code_num*	*codeword*
0	0	1
1	1	010
-1	2	011
2	3	00100
-2	4	00101
3	5	00110
-3	6	00111
4	7	0001000
-4	8	0001001
5	9	0001010
-5	10	0001011
6	11	0001100
-6	12	0001101
7	13	0001110
-7	14	0001111
8	15	000010000
-8	16	000010001
9	17	000010010
-9	18	000010011
…	…	…

Table 2: MVDS AND CORRESPONDING EXP-GOLOMB CODEWORDS

3) RESIDUAL DATA ENCRYPTION

In order to keep high security and sensitive data i.e., the continuing data in both I-frames and Pframes should be encrypted. In this section, a novel method for encrypting the residual data based on the characteristics of codeword substitution is presented in detail.The codeword for each level is made up of aprefix (level_prefix) and a suffix (level_suffix) as

Level codeword = [level_ prefix], [level_ suffix]

Table 3 shows Levels with different suffix Length and corresponding codeword. 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 E_Key5.According to Table III, the last bit encryption may change the sign of Levels, but does not affect the length of the codeword and satisfies the format flexibility. It should be satisfies the format flexibility. It should be verified that when suffix Length is equal to 0, the code words should keep unchanged during the encryption process.

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

Table3: Levels and Corresponding Codeword's

B. DATA EMBEDDING

In the encrypted bitstream of H.264/AVC, the proposed data embedding is accomplished by substituting eligible codewords. the codewords of Levels within P-frames are used for data hiding, while the codewords of Levels in I-frames are remained unchanged. Because I-frame is the first frame in a group of pictures (GOPs), the error occurred in I-frame will be propagated to subsequent P-frames. unchanged. Because I-frame is the first frame in a group of pictures (GOPs), the error occurred in I-frame will be propagated to subsequent P-frames.

On the other hand, the codewords substitution should fulfil the following three limitations. First, the bitstream after codeword substituting must remain syntax compliance so that it can be decoded by standard decoder. Second, to keep the bit-rate unchanged, the substituted codeword should have the same size as the original codeword. The codewords of Levels which suffix Length is 2 or 3 would be divided into two opposite codespaces denoted as C0 and C1 as shown in Fi2. The codewords assigned in C0 and C1are associated with binary hidden information "0" and "1". Suppose the additional data that we want to embed is a binary sequence denoted sB={b(i)|i =1,2,⋯,L,b(i)∈ {0,1}}.

Data hiding is performed directly in encrypted bit-stream through the following steps

Step1. In order to enhance the security, the additional data is encrypted with the chaotic pseudo-random sequence

P = {p(i)|i = 1, 2, · · · , L, p(i) ∈ {0, 1}}[22] to generate the to-be-embedded sequence W = {w(i)|i = 1, 2, · · · , L,w(i) ∈ {0, 1}}. The sequence P is generated by using logistic map with an initial value [22], i.e., the data hiding key. It is very difficult for anyone who does not retain the data hiding key to recover the additional data.

Step2. The codewords of Levels are obtained by parsing the encrypted video bitstream.

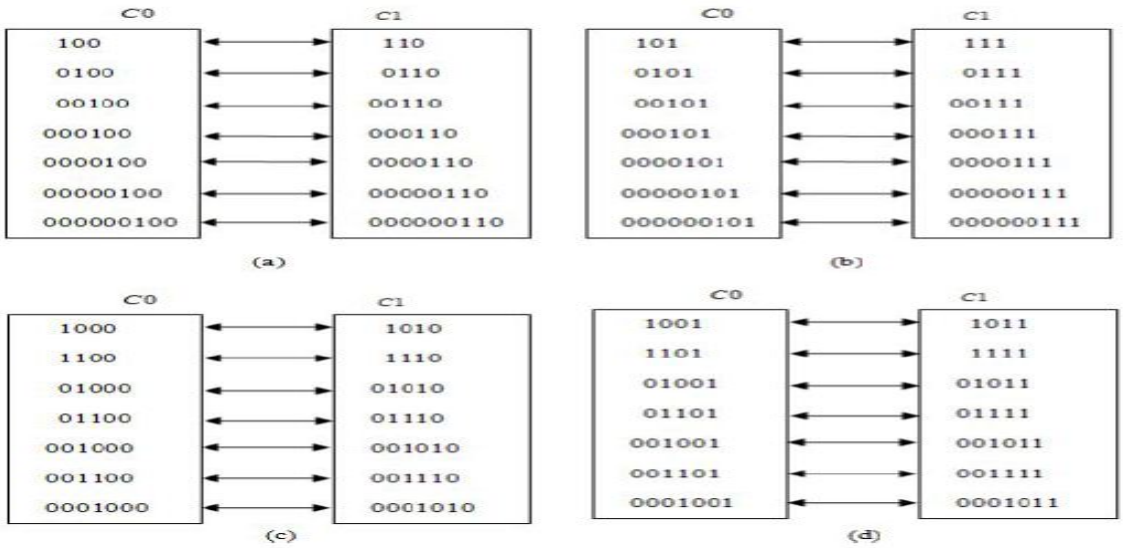

Figure 2: CAVLC codeword mapping. (a) Suffix Length = 2& Level > 0. (b) Suffix Length = 2& Level > 0. (c) Suffix Length = 3& Level > 0. (d) Suffix Length = 3& Level < 0.

Step3. If current codeword belongs to codespaces C0orC1, the to-be-embedded data bit can be embedded by codeword Substituting. Otherwise, the codeword is left unchanged. The detailed procedure of codeword substituting for data hiding is shown in below. For example, when Level is positive 1 and its suffix Length is 3, then its corresponding codeword is "1000" which belongs to C0 as shown in Fig. 2(c). If the data bit "1" needs to be embedded, the codeword "1000" should be replaced with "1010". Otherwise, if the data bit "0" needs to embedded, the codeword "1000" will keep unchanged.

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

Step4. Choose the next codeword and then go to Step3 for data hiding. If there are no more data bits to be embedded, the embedding process is stopped.

Suppose the to-be-embedded data is "1001", the CAVLC codeword of Level parsing from H.264/AVC bitstream is "01 010 00100 00100 0001011 0000100" and the encryption stream is "10111", an example of data embedding based on codeword mapping is shown in Fig 3 codeword mapping is shown in Fig 3

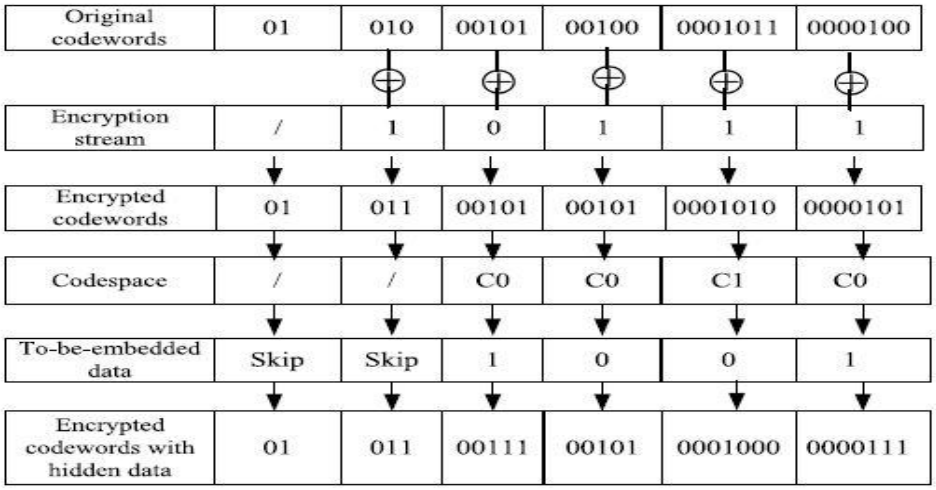

Figure 3: Data Embedding

C. ARITHMETIC COMPRESSION

In this section, a lossless compression technique is used to further reduce the file size. Compression is necessary to reduce the volume of data to be transmitted (text, fax, images) and to reduce the bandwidth required for transmission and to reduce storage requirements of video. Compression possible is because of Properties of human perception and also the redundancy in digital audio, image, and video data. We are using arithmetic compression because it provides highest compression ratio in comparison with RLE or Huffman coding. Image is a rectangular array of pixel values; arithmetic compression which is used in image can be applied for videos because video is nothing but a sequence of images played out at a certain rate where neighbouring sample values are correlated. In digital video, in addition to spatial redundancy, neighbouring images in a video sequence may be similar (temporal redundancy).

Arithmetic coding is more complex and instead of coding each individual symbol in the data it encodes all of the data as a single fraction on the interval [0, 1). Arithmetic coding relies on the symbols' frequencies in the data, which will be seen in the encoding process for which we construct the symbol table which includes each of the individual symbols in the data, their frequency and each symbols range, which we calculate using following equations:

$$L = Lp + R* l$$

$$H = Lp + R* h$$

where H and L are the high and low values, Hp and Lp are the previous high and low values, h and l are the high and low ends of the range from the symbol table, and R is the range in the current row.

E. DATA EXTRACTION

In this scheme, the hidden data can be extracted either in encrypted(scheme 1) or decrypted domain(scheme 2). Data Extraction process is fast and simple.

1) SCHEME I: ENCRYPTED DOMAIN EXTRACTION.

To protect privacy, a database manager (e.g., cloud server) may only get access to the data hiding key and have to manipulate data in encrypted domain. Data extraction in encrypted domain guarantees the feasibility of our scheme in this case. In encrypted domain, as shown in below, encrypted video with hidden data is directly sent to the data extraction module, and the extraction process is given as follows.

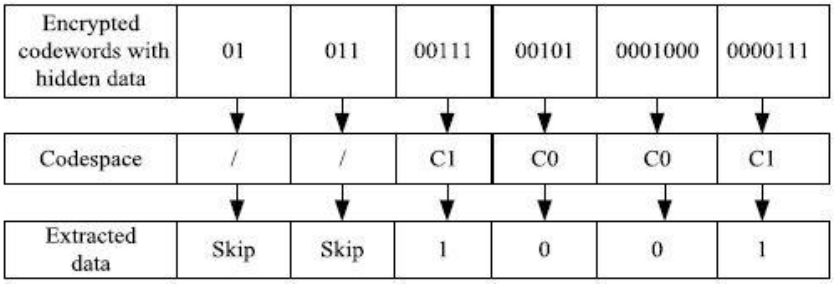

Figure 4: Data Extraction in encrypted domain

***2)* SCHEME II: DECRYPTED DOMAIN EXTRACTION**

In some cases, users want to decrypt the video first and extract the hidden data from the decrypted video. For example, an authorized user, which owned the encryption key, received the encrypted video with hidden data. The received video can be decrypted using the encryption key. That is, the decrypted video still includes the hidden data, which can be used to trace the source of the data. Data extraction in decrypted domain is suitable for this codewords can be decrypted by performing XOR operation with generated encryption streams, and then two XOR operations cancel each other out, which renders the original plaintext. Since the encryption streams depend on the encryption keys, the decryption is possible only for the authorized users. After generating the decrypted codewords with hidden data, the content owner can further extract the hidden information.

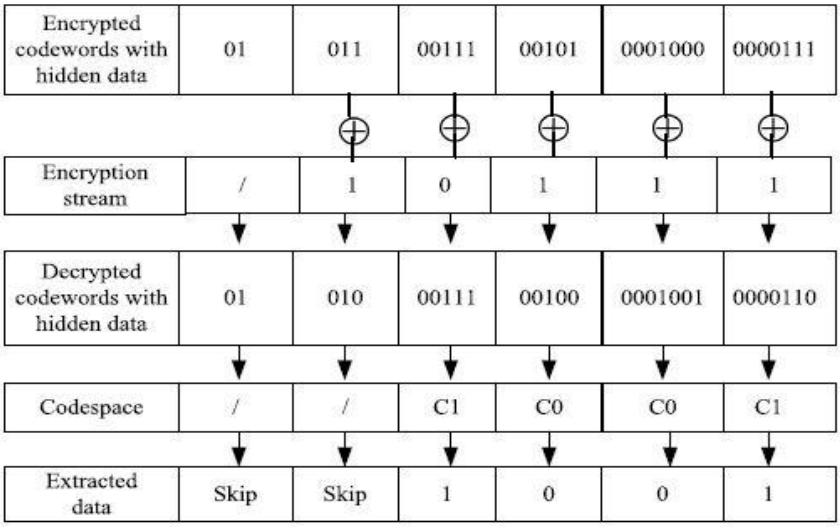

Figure 5: Data Extraction in decrypted domain

THE DECRYPTION AND DATA EXTRACTION'S WHOLE PROCESS CAN BE GIVEN AS FOLLOWS:

Step1: With the encryption keys as given in encryption process, generate encryption streams.

Step2: By parsing the encrypted bit stream, identify the codeword's of IPMs, MVDs, Sign_of_TrailingOnes and Levels. Step3: Because the XOR operation is symmetric, the decryption process is identical to the encryption process. By performing XOR operation with generated encryption streams, the encrypted codeword's can be decrypted and then two XOR operations cancel each other out which renders the original plaintext. The decryption is possible only for the authorized users since the encryption streams depend on the encryption key. The owner can further extract the hidden information only after generating the decrypted codeword's with hidden data.

Step4: The last bit encryption may change the sign of Level according to Levels and Corresponding Codeword's table. The extracted data bit is "0" when the decrypted codeword of Level belongs to code space C0. The extracted data bit is "1" when the decrypted codeword of Level belongs to code space C1.

Step5: According to the data hiding key, generate the same pseudo-random sequence P that was used in embedding process. In order to get the original additional information, the extracted bit sequence should be decrypted.

5. EXPERIMENTAL RESULTS

The proposed data hiding scheme has been implemented with Six well-known standard video sequences (i.e., Stefan, Table, Template, Mobile, Hall, and News) in QCIF format (176 × 144) at the frame rate 30 frames/s are used for simulation. The first 10 frames in each video sequence are used in the experiments. The GOP (Group of Pictures) structure is "IPPPP: one I frame followed four P frames".

A. SECURITY OF ENCRYPTION ALGORITHM

The demonstration is shown in below. It contains an original frame from video, their corresponding encrypted results, followed by encrypted video with hidden data and finally the decrypted video are depicted below. Other frames have a similar effect of encryption. Due to space limitations, we do not list the results of all frames. It should be mentioned that not every video can be degraded to the same extent. The amount of compression taken place can be measured by its compression ratio.

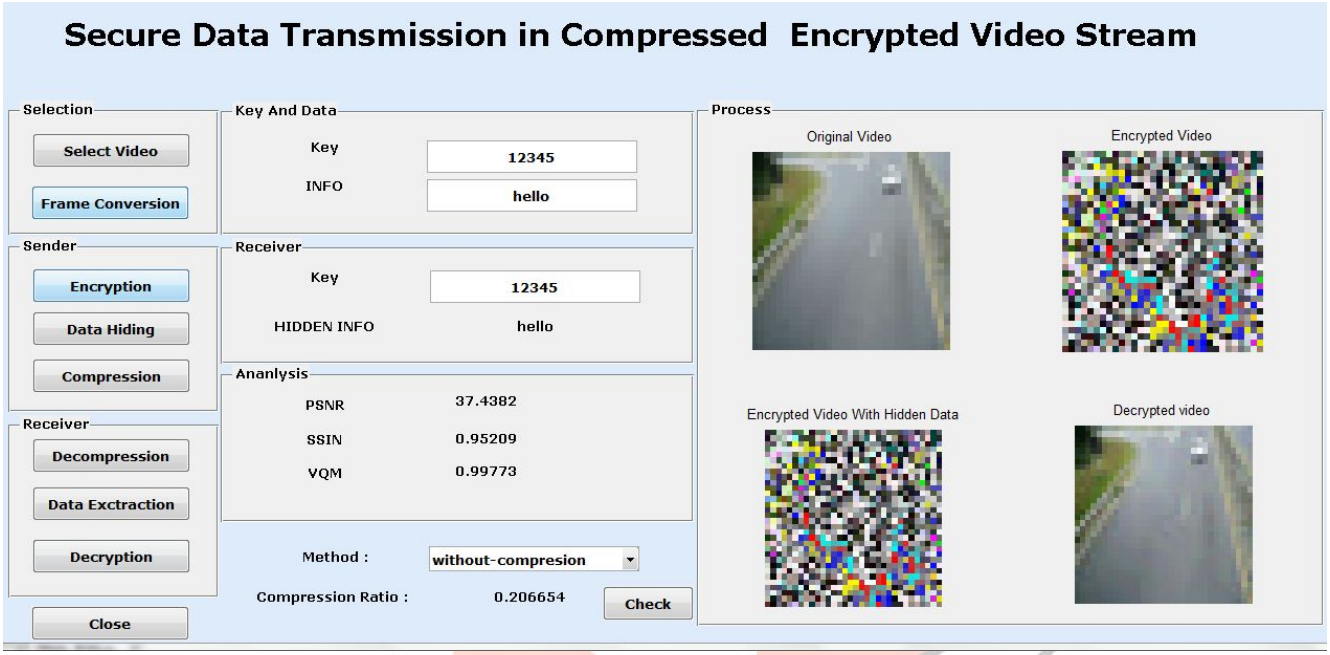

Figure 6: Experimental simulation of video frames without arithmetic compression

B. VISUAL QUALITY OF STEGO VIDEO

Simulation results have demonstrated that we can embed the additional data with a large capacity into P-frames while preserving high visual quality. The encrypted and decrypted video frames with hidden data are shown in simulation results respectively. In the experiments, no visible artefacts have been observed in all of the decrypted video frames with hidden data. Besides subjective observation, PSNR (Peak Signal to Noise Ratio), SSIM (Structural Similarity Index), and VQM (Video Quality Measurement) have been adopted to evaluate the perceptual quality [22]. PSNR is widely used objective video quality metric. However, it does not perfectly correlate with a perceived visual quality due to nonlinear behaviour of human visual system. The SSIM index lies in the range between 0 and 1, where 1 indicates the reference image is identical than the target image. The VQM is another approach to measure video quality that correlates more with the human visual system. In general, the lower VQM value indicates higher perceptual video quality, and zero indicates excellent quality.

C. EFFECT OF ARITHMETIC COMPRESSION

Even after performing arithmetic compression the change in PSNR, SSIM, VQM is quite minute that means compression does not affect the quality of video and also the encryption and data hiding process,(can be seen by comparing fig6 and fig7). With arithmetic compression, the video is compressed up to 80%, also the bandwidth during transmission is saved, thus, facilitating efficient transmission for cloud storage.

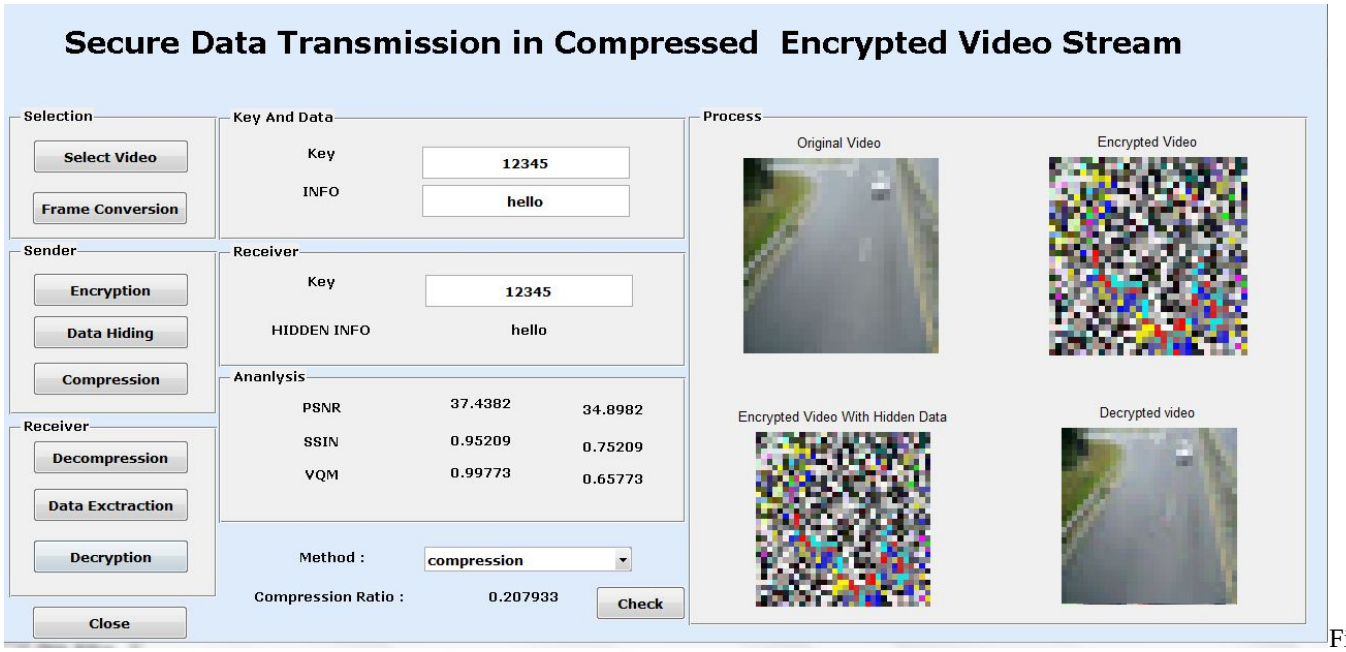

Figure 7: Experimental simulation of video frames with arithmetic compression

6. Conclusion

Because of the privacy-preserving requirements from cloud data management, encrypted media's data hiding is a new topic in the emerging world. To embed additional data in encrypted video stream is presented in this paper and can have video encryption, data embedding and data extraction. Even after encryption and data embedding, the algorithm should preserve the bit-rate correctly. As it is directly performed in the compressed and encrypted domain, it is simple to implement. Using codeword substitution, the content owner should embed the additional data into the encrypted bit stream without the knowledge of original video content. This can preserve the confidentiality of the content completely since data hiding is completed entirely in the encrypted domain. Data extraction can be done either in encrypted or decrypted domain and it can provide two different practical applications. Furthermore, even arithmetic compression is done on the encrypted video making it convenient for cloud storage Experimental results have shown that the proposed encryption and data embedding scheme can preserve file-size, whereas the degradation in video quality caused by data hiding is quite small.

7. References

[[1] W. J. Lu, A. Varna, and M. Wu, "Secure video processing: Problems and challenges," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, Prague, Czech Republic, May 2011, pp. 5856–5859.

[2] B. 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] P. J. Zheng and J. W. Huang, "Walsh-Hadamard transform in the homomorphic encrypted domain and its application in image watermarking," in Proc. 14th Inf. Hiding Conf., Berkeley, CA, USA, 2012, pp. 1–15.

[4] 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.

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

[6] W. Hong, T. S. Chen, and H. Y. Wu, "An improved reversible data hiding in encrypted images using side match," IEEE Signal Process. Lett., vol. 19, no. 4, pp. 199–202, Apr. 2012.

[7] X. P. Zhang, "Separable reversible data hiding in encrypted image," IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 826–832, Apr. 2012.

[8] 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.

[9] 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.

[10] S. 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.
[11] 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.
[12] 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.
[13] S. G. Lian, Z. X. Liu, Z. Ren, and H. L. Wang, "Secure advanced video coding based on selective encryption algorithms," IEEE Trans. Consumer Electron., vol. 52, no. 2, pp. 621–629, May 2006.
[14] Z. 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, May 2011.
[15] M. N. Asghar and M. Ghanbari, "An efficient security system for CABAC bin-strings of H.264/SVC," IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 3, pp. 425–437, Mar. 2013.
[16] T. Stutz and A. Uhl, "A survey of H.264 AVC/SVC encryption," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 3, pp. 325–339, Mar. 2012.
[17] Advanced Video Coding for Generic Audiovisual Services, ITU, Geneva, Switzerland, Mar. 2005.
[18] 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.
[19] I. E. G. Richardson, H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia. Hoboken, NJ, USA: Wiley, 2003.
[20] D. K. Zou and J. A. Bloom, "H.264 stream replacement watermarking with CABAC encoding," in Proc. IEEE ICME, Singapore, Jul. 2010, pp. 117–121.
[21] D. W. Xu and R. D. Wang, "Watermarking in H.264/AVC compressed domain using Exp-Golomb code words mapping," Opt. Eng., vol. 50, no. 9, p. 097402, 2011.
[22] D. W. Xu, R. D. Wang, and J. C. Wang, "Prediction mode modulated data-hiding algorithm for H.264/AVC," J. Real-Time Image Process., vol. 7, no. 4, pp. 205–214, 2012.
[23] T. Shanableh, "Data hiding in MPEG video files using multivariate regression and flexible macroblock ordering," IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 455–464, Apr. 2012.