IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 10, October 2017

Enhanced Steganography in Encoded Video Streams using Multi Motion Vector Difference Model

Ms. Nanthini.D[1], Mrs. Sathiya Priya.T[2]

M. Phil Full Time Scholar Computer Science, Shri Sakthi Kailassh Women's College, Salem, India[1]

Assistant Professor, Department of Computer Science, Shri Sakthi Kailassh Women's College, Salem, India[2]

Abstract: In this paper analysis a digital video sometimes are stored and processed in an encrypted format to maintain privacy and security. For the purpose of content notation, it is necessary to perform data hiding in these encrypted videos. In this way, data hiding in encrypted domain without decryption preserves the confidentiality of the content. In addition, it is more efficient without decryption followed by data hiding and re-encryption. This project proposes a novel scheme of data hiding directly in the encrypted version of AVI video stream, which includes the following three parts, i.e., AVI video encryption, data embedding, and data extraction. Furthermore, video file size is strictly preserved even after encryption and data embedding. Experimental results have demonstrated the feasibility and efficiency of the proposed scheme.

Keywords: Data Hiding, Stream Extraction, Steganography, Image Processing, DES, E-LSB Representation.

I. INTRODUCTION

In recent years, signal processing in the encrypted domain has attracted considerable research interest. As an effective and popular means for privacy protection, encryption converts the ordinary signal into unintelligible data, so that the traditional signal processing usually takes place before encryption or after decryption. However, in some scenarios that a content owner does not trust the processing service provider, the ability to manipulate the encrypted data when keeping the plain content unrevealed is desired. For instance, when the secret data to be transmitted are encrypted, a channel provider without any knowledge of the cryptographic key may tend to compress the encrypted data due to the limited channel resource.

This paper proposes a novel scheme for classic data hiding in encrypted images or video files. In the first phase, a content owner encrypts the original uncompressed image /video using an encryption key. Then, a data-hider may replace the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate some additional data. With an encrypted image or video containing additional data, if a receiver has the data-hiding key, he can extract the additional data though he does not know the image content. If the receiver has the encryption key, he can decrypt the received data to obtain an image similar to the original one, but cannot extract the additional data. If the receiver has both the data-hiding key and the encryption key, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image when the amount of additional data is not too large.

II. LITERATURE SURVEY

In this paper [1], an enhanced watermarking scheme is presented. Compared with Solanki et al.'s scheme, the enhanced scheme increases effective watermarking capacity, avoids additional overhead and overcomes an inherent flaw that watermarking capacity depends on the probability distribution of input watermark sequence. Based on the security requirements of buyer–seller watermarking protocols, a new watermarking scheme in the encrypted domain with flexible watermarking capacity is proposed. It improves the robustness of watermark sequence against image compressions and enables image tampering detection. Watermark extraction is blind, which employs the same threshold criterion and secret keys as watermark embedding. Experimental results demonstrate that the enhanced watermarking scheme eliminates the drawbacks of Solanki et al.'s scheme and that the proposed watermarking scheme in the encrypted domain out-performs Kuribayashi and Tanaka's scheme.

Rini.J et al [6] describe the secure and authenticated discrete reversible data hiding in cipher images deals with security and authentication. In the first phase, a content owner encrypts the original uncompressed image using an encryption key. Then, a data hider may compress the least significant bits of the encrypted image using a data hiding key to create a sparse space to accommodate some additional data. With an encrypted image containing additional data, if a receiver

DOI10.17148/IJARCCE.2017.61029

has the data hiding key, receiver can extract the additional data though receiver does not know the image content. If the receiver has the encryption key, can decrypt the received data to obtain an image similar to the original one. If the receiver has both the data hiding key and the encryption key, can extract the additional data and recover the original content.

In this paper [7], we propose a novel method by reserving room before encryption with a traditional RDH algorithm, and thus it is easy for the data hider to reversibly embed data inthe encrypted image. This method provides improved PSNR ratio and recovers image with its original quality. Divya h. T et al [3] describe the data hacking is very challenging problem in today's internet world. There are number of techniques to secure the data. So, the data hiding in the encrypted image comes into the picture, but occurrence of distortion at the time of data extraction is a main problem. So Reversible Data Hiding (RDH) in encrypted image is used. With this method original cover can be recovered.

This paper [4] mainly focus on watermarking of compressed-encrypted JPEG2000 images, where the encryption refers to the ciphering of complete JPEG2000 compressed stream except headers and marker segments, which are left in plaintext for format compliance.D.Prabhakar et al [4] describe the IMAGE watermarking, which is finding more and more support as a possible solution for the protection of intellectual property rights. To this aim, many techniques have been proposed in the literature over the last few years, and many commercial products are already available. It is possible to state that the most important features a watermarking technique to be used for IPR protection should exhibit are unobtrusiveness and robustness.

Robust quantization index modulation (QIM) based watermarking technique, which embeds the watermark in the encrypted domain. In the technique proposed in [11], the addition or subtraction of a watermark bit to a sample is based on the value of quantized plaintext sample. However, in our algorithm, the watermark embedder does not have access to the plain text values. They have only compressed-encrypted content. Also the watermark embedders do not have the key to un-encrypt and get the plain text compressed values. Thus, watermarking in compressed-encrypted domain using the technique proposed in [11] is very challenging.

IV. VIDEO ENCRYPTION METHODOLOGY

A novel scheme of data hiding in the encrypted version of AVI videos is presented, which includes three parts, i.e., AVI video encryption, data embedding and data extraction. The content owner encrypts the original AVI video stream using standard stream ciphers with encryption keys to produce an encrypted video stream. Then, the data-hider (e.g., a cloud server) can embed the additional data into the encrypted video stream by using codeword substituting method, without knowing the original video content. At the receiver end, the hidden data extraction can be accomplished either in encrypted or in decrypted version.

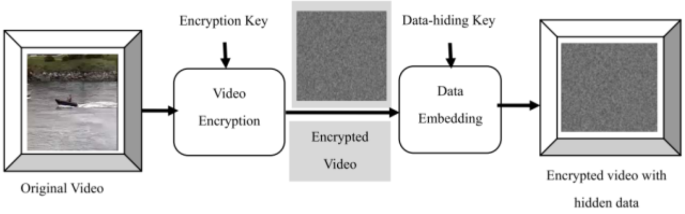


Fig 2.1 Video encryption and data embedding at the sender end.

A. ENCRYPTION MODEL

Video encryption often requires that the scheme be time efficient to meet the requirement of real time and format compliance. It is not practical to encrypt the whole compressed video bit stream 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.

In this study, an AVI video encryption scheme with good performance including security, efficiency, and format compliance is proposed. By analyzing the property of AVI codec, three sensitive parts (i.e., IPMs, MVDs, and residual coefficients) are encrypted with stream ciphers. Compared with the proposed encryption algorithm is performed not during AVI encoding but in the AVI compressed domain. In this case, the bitstream will be modified directly. Selective encryption in the AVI compressed domain has been already presented on context-adaptive variable length coding (CAVLC) and context-adaptive binary arithmetic coding (CABAC). In this study, improved and enhanced the previous proposed approach by encrypting more syntax elements. We encrypt the codewords of IPMs, the codewords of MVDs, and the codewords of residual coefficients. The encrypted bitstream is still AVI compliant and can be decoded by any standard-compliant AVI decoder, but the encrypted video data is treated completely different compared to plaintext video data.

DOI10.17148/IJARCCE.2017.61029

- **Intra-Prediction Mode (IPM) Encryption:**

According to AVI 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. Here, IPMs in the Intra_4 × 4 and Intra_16 × 16 blocks are chosen to encrypt. Four intra prediction modes (IPMs) are available in the Intra_16 × 16. 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).

- **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 AVI , motion vector prediction is further performed on the motion vectors, which yields MVD. In AVI baseline profile, Exp-Golomb entropy coding is used to encode MVD. The codeword of Exp-Golomb is constructed as[M zeros] [I N FO], where I N FO is an M-bit field carrying information.

- **Residual Data Encryption**:

In order to keep high security, another type of sensitive data, i.e., the residual data in both I-frames and P-frames should be encrypted. In AVI baseline profile, CAVLC entropy coding is used to encode the quantized coefficients of a residual block. Each CAVLC codeword can be expressed as the following format:

{Coef f token, Sign of T railingOnes, Level, T otal zeros, Run bef ore}

B. DATA EMBEDDING MODEL

In the encrypted bitstream of AVI , the proposed data embedding is accomplished by substituting eligible codewords of Levels. Since the sign of Levels are encrypted, data hiding should not affect the sign of Levels. Besides, the codewords substitution should satisfy 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.
- Third, data hiding does cause visual degradation but the impact should be kept to minimum. That is, the embedded data after video decryption has to be invisible to a human observer.

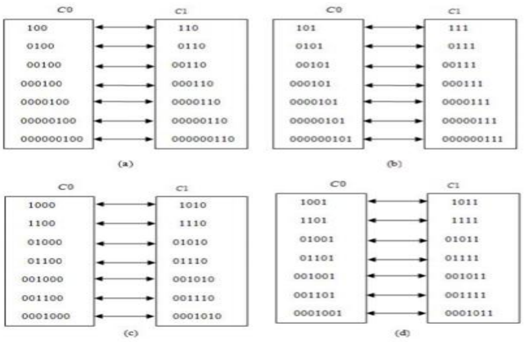
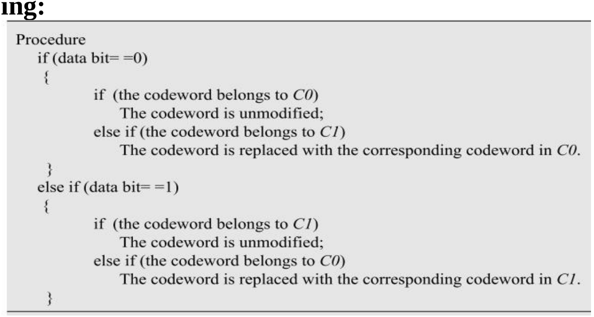


Fig. 4.2. CAVLC codeword mapping. (a) su f f ix Length = 2& Level > 0. (b) su f f ix Length = 2& Level > 0. (c) su f f ix Length = 3& Level > 0. (d) su f f ix Length = 3& Level < 0

So the value of Level corresponding to the substituted codeword should keep close to the value of Level corresponding to the original codeword. In addition, 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

Procedure of Codeword Mapping:



```
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.
    }
```

DOI10.17148/IJARCCE.2017.61029

C. DATA EXTRACTION MODEL

Scheme I: Enhanced 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.

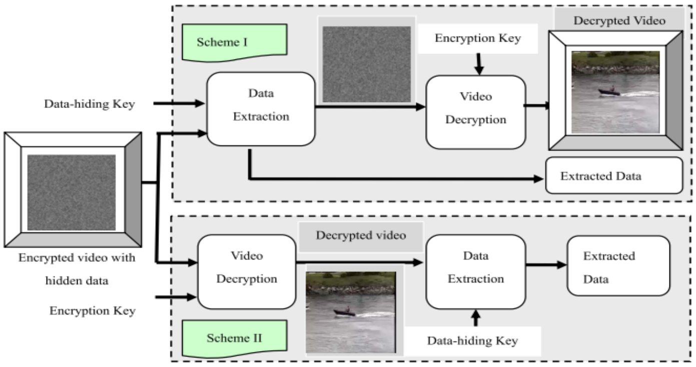


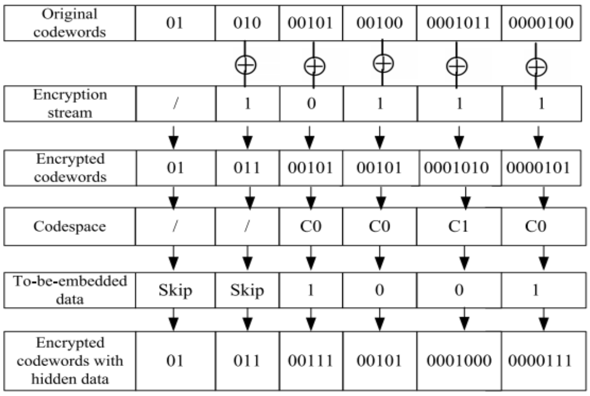
Fig 4.3 Data extraction and video display at the receiver end in two scenarios.

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 by parsing the encrypted bitstream.
- **Step2**: If the codeword belongs to codespace C0, the extracted data bit is “0”. If the codeword belongs to codespace C1, the extracted data bit is “1”.
- **Step3**: According to the data hiding key, the same chaotic pseudo-random sequence P that was used in the embedding process can be generated. Then the extracted bit sequence could be decrypted by using P to get the original additional information. Since the whole process is entirely operated in encrypted domain, it effectively avoids the leakage of original video content.

Scheme II: Enhanced Decrypted Domain Extraction.

In scheme I, both embedding and extraction of the data are performed in encrypted domain. However, 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 case.



Original codewords	01	010	00101	00100	0001011	0000100
Encryption stream	/	1	0	1	1	1
Encrypted codewords	01	011	00101	00101	0001010	0000101
Codespace	/	/	C0	C0	C1	C0
To-be-embedded data	Skip	Skip	1	0	0	1
Encrypted codewords with hidden data	01	011	00111	00101	0001000	0000111

Fig 4.4 Data embedding Model

Step1:
Generate encryption streams with the encryption keys as given in encryption process.
Step2:
The codewords of IPMs, MVDs, Sign_of_TrailingOnes and Levels are identified by parsing the encrypted bitstream.
Step3:
The decryption process is identical to the encryption process, since XOR operation is symmetric. The encrypted 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.

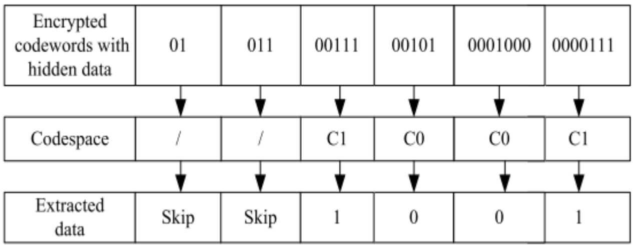


Fig 4.5 Data extraction in encrypted domain model

Step4:
The last bit encryption may change the sign of Level. Encrypted codeword and the original codeword are still in the same codespaces. If the decrypted codeword of Level belongs to codespace C0, the extracted data bit is “0”. If the decrypted codeword of Level belongs to codespace C1, the extracted data bit is “1”.

Step5:
Generate the same pseudo-random sequence P that was used in embedding process according to the data hiding key. The extracted bit sequence should be decrypted to get the original additional information.

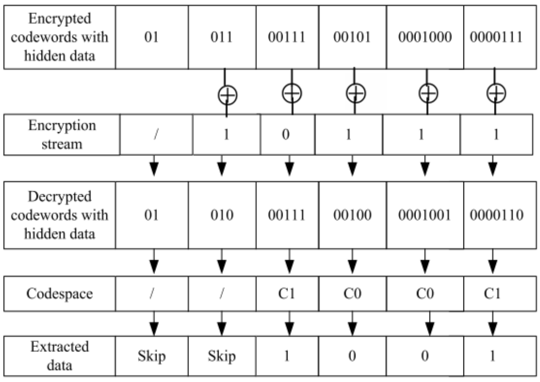


Fig 4.6 Data extraction in decrypted domain

V. EXPERIMENTAL RESULTS

The following **Table 5.1** describes experimental result for proposed system algorithms. The table input text size for DES and 3DES algorithms for encryption execution time details are shown.

The following **Fig 5.1** describes experimental result for proposed system algorithms. The table input text size for DES and 3DES algorithms for encryption execution time details are shown

The following **Table 5.2** describes experimental result for proposed system algorithms. The table input text size for DES and 3DES algorithms for decryption execution time details are shown

Table1 5.1: Execution Time (Milliseconds) of Encryption of Different Text data size (DES & 3DES)

Input Text Size (Bytes)	**DES**	**3DES**
75	21	57
96	32	55
112	54	81
286	97	173
359	188	198
600	198	202
951	391	327
5345	1399	1149
Throughput (MB/sec)	**3.01**	**2.8**

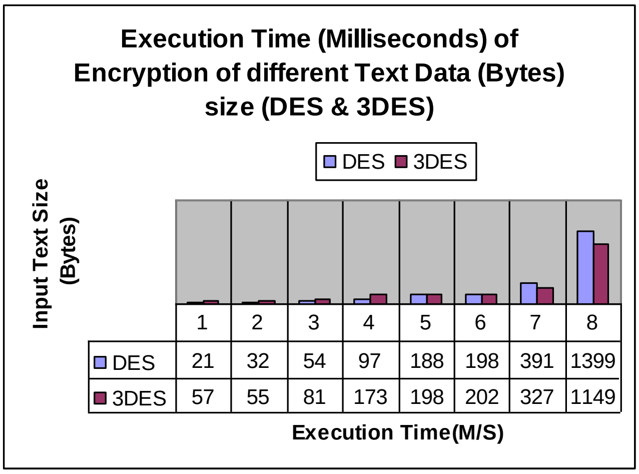


Fig 5.1: Execution Time (Milliseconds) of Encryption of Different Text Data Size (DES and 3DES)

Table 5.2: Execution Time (Milliseconds) of Decryption of Different Text data size (DES & 3DES)

Input Text Size (Bytes)	**DES**	**3DES**
75	25	62
96	34	59
112	56	84
286	102	176
359	192	203
600	205	210
951	402	333
5345	1401	1152
Throughput (MB/sec)	**3.03**	**2.84**

The following Fig 5.2 describes experimental result for proposed system algorithms. The table input text size for DES and 3DES algorithms for decryption execution time details are shown.

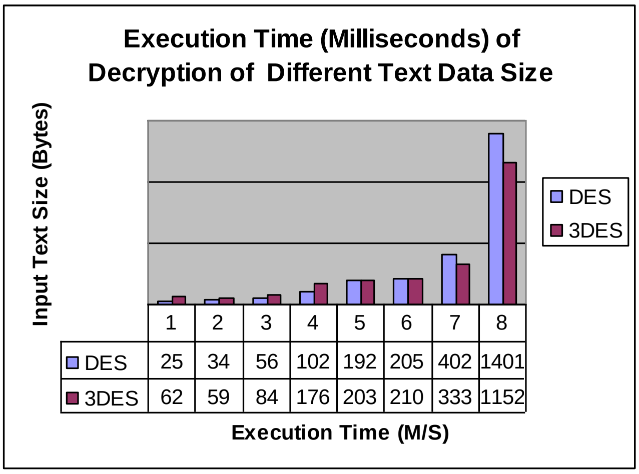


Fig 5.2: Execution Time (Milliseconds) of Decryption of Different Text Data Size (DES and 3DES)

VI. CONCLUSION

The reversible data hiding in encrypted image is investigated. Most of the work on reversible data hiding focuses on the data embedding/extracting on the plain spatial domain. But, in some applications, an inferior assistant or a channel administrator hopes to append some additional message, such as the origin information, image notation or authentication data, within the encrypted image though he does not know the original image content. And it is also hopeful that the original content should be recovered without any error after image decryption and message extraction at receiver side. A content owner encrypts the original image using an encryption key, and a data-hider can embed additional data into the encrypted image using a data-hiding key though he does not know the original content. With an encrypted image containing additional data, a receiver may first decrypt it according to the encryption key, and then extract the embedded data and recover the original image according to the data-hiding key. In the scheme, the data extraction is not separable from the content decryption. In future additional encryption techniques applied in data transfer.

REFERENCES

[1] W. J. Lu, A. Varna, and M. Wu, "Secure video processing: Problems andchallenges," 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 homo- morphic 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 encryp- tion," 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

DOI10.17148/IJARCCE.2017.61029