

Secret Data Hiding in Encrypted Compressed Video Bitstreams for Privacy Info Protection

Sagar Rajkumar Somani¹, P.S. Malge²

P.G. Student¹, Associate Professor²

Department of Electronics Engineering

Walchand Institute of Technology, Solapur, Maharashtra, India

Corresponding Author: er.sagarsomani@gmail.com¹

Abstract

Digital video and secret data sometimes needs to be stored and processed in an encrypted format to maintain security and privacy. Data hiding approach is necessary to perform in these encrypted videos for the purpose of content notation and tampering detection. 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. Here, data hiding directly in the encrypted version of H.264/AVC video stream is approached, which includes the following three parts, i.e., H.264/AVC video encryption, data embedding, and data extraction. By analyzing the property of H.264/AVC codec, the code words of intra prediction modes, the code words of motion vector differences, and the code words of residual coefficients are encrypted with stream ciphers. Then, a data hider may embed additional data in the encrypted domain by using bits replacement technique, without knowing the original video content. Chaos crypto system is used here to encrypt/decrypt secret text data before/after data embedding/extraction. In order to adapt different application scenarios, data extraction is done either in the encrypted domain or in the decrypted domain.

Keywords: Encryption, chaos, bit substitution, secret, decryption

I. INTRODUCTION

A novel scheme of data hiding in the encrypted version of H.264/AVC videos is presented, which includes three parts, i.e., H.264/AVC video encryption, data embedding and data extraction[2]. The content owner encrypts the original H.264/AVC video stream using standard stream ciphers with encryption keys to produce an encrypted video stream. Then, the data-hider can embed the additional data into the encrypted video stream by using bit 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[5].

The compression/decompression cycle is time-consuming and hampers real-time implementation. Besides, encryption and watermark embedding would lead to increasing the bit-rate of H.264/AVC bit stream [7]. Therefore, it becomes highly desirable to develop data hiding algorithms that work entirely on encoded bit stream in the encrypted domain. However, there are some significant challenges for data hiding directly in compressed and encrypted bit stream [3]. The first challenge is to determine where and how the bit stream can be modified so that the encrypted bit stream with hidden data is still a compliant

compressed bit stream. 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.

The fourth challenge is that the hidden data can be extracted either from the encrypted video stream or from the decrypted video stream, which provides two different practical applications. In contrast to the existing technologies discussed above, the proposed scheme can achieve excellent performance in the following three different prospects

- The data hiding is performed directly in encrypted H.264/AVC video bit stream.
- The scheme can ensure both the format compliance and the strict file size preservation.
- The 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. Although few methods have been proposed to embed data into H.264/AVC bit stream directly, however, these methods cannot be implemented in the encrypted

domain.[3] In the encrypted bit stream of H.264/AVC, the proposed data embedding is accomplished by bit substitution method. Besides, the bits substitution should satisfy the following three limitations [9]. First, the bit stream after bit substituting must remain syntax compliance so that it can be decoded by standard decoder. Second, to keep the bit-rate unchanged, the bits substituted should have the same size as the original

bitstream. Third, data hiding does cause visual degradation but the impact should be kept to minimum [6]. That is, the embedded data after video decryption has to be invisible to a human observer. As the (GOP) group of pictures structure is “IPPPP: one I frame followed by four P frames”. 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 [7].

II. METHODOLOGY

2.1 Block Diagram

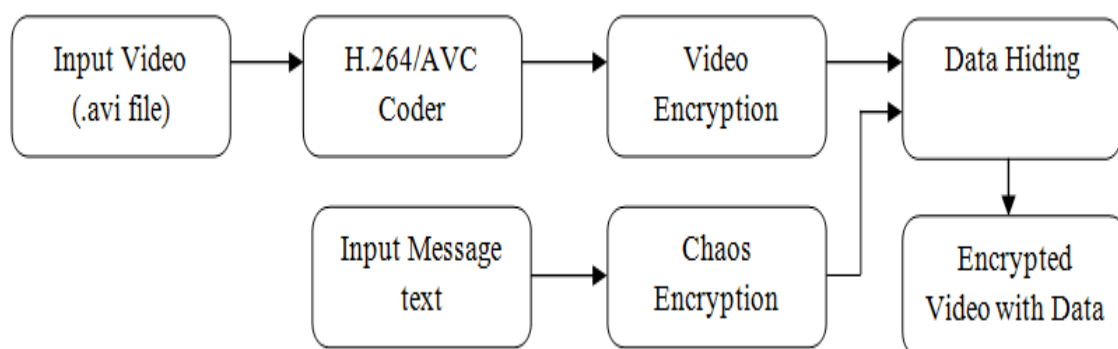


Figure 2.1 Video Encryption and Data Hiding

2.1.1 Input Video:

The encoding process starts with selecting the video for data hiding. Video can be selected from any location of the computer. This video is divided into multiple frames. Each of these frames is an image. The frame is encrypted using any

one of the convenient encryption mechanisms.

2.1.2 H.264/AVC coder:

In H.264/AVC coder converts video into a compressed format by setting appropriate macroblock size and quantization respectively.

2.1.3 Video Encryption:

The Advanced Encryption Standard uses XOR on single byte. XOR is used in the key schedule. The last bit of the bitstream is encrypted by applying the bitwise XOR operation with a standard stream cipher.

2.1.4 Input Message:

Input the text data which is to be hided in to an image. The text may be either numerical, alphabets or combination of both. The data which is given as input is converted into its corresponding binary form.

2.1.5 Chaos Encryption

This method is one of the advanced encryption standard to encrypt the secret text data for secure transmission.

It encrypts the text data's with encryption key value generated from chaotic sequence with threshold function by bit XOR operation. It is very useful to transmit the secret data through unsecure channel securely which prevents data hacking.

2.1.6 Data hiding

Data Hiding is the process of secretly embedding information inside a data source without changing its perceptual quality.

Data Hiding is the art and science of writing hidden messages in such a way that no one apart from the sender and intended recipient even realizes there is a hidden message. *See Figure 2.2*

2.2.1 Encrypted Video with Data

To protect privacy, a data manager may only get access to the data hiding key and have to manipulate data in encrypted video.

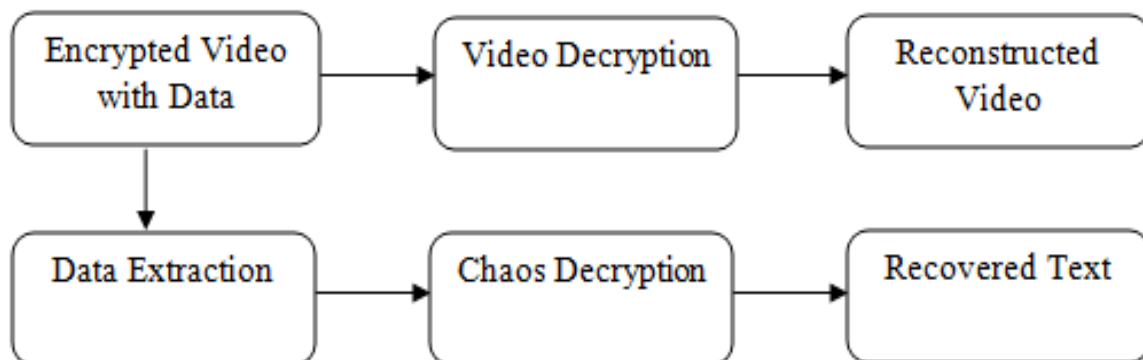


Figure 2.2 Data Extraction and Video Decryption

2.2.2 Data extraction

In encrypted video guarantees the feasibility in this case. Encrypted video with hidden data is directly sent to the data extraction module.

2.2.3 Video Decryption & Reconstruction

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. Thus the original frames are reconstructed.

Therefore, an algorithm is developed to hide the secret data in compressed video bit stream and to achieve better performance in terms of computation efficiency, high data security and video quality after decryption. The parameters such as Mean Square Error, PSNR, Correlation and SSIM will be evaluated to measure its efficiency.

III. RESULTS

The proposed data hiding scheme has been implemented in H.264/AVC. Before data hiding in encrypted compressed video

signal the secret text data was encrypted using chaos crypto system.

Then this encrypted secret text data was embed with the encrypted compressed video signal by bits replacement technique. The user has been given choice to choose the number of frames to process for data hiding.

Then, in order to adapt different application scenarios, data extraction is done either in encrypted domain or in decrypted domain.

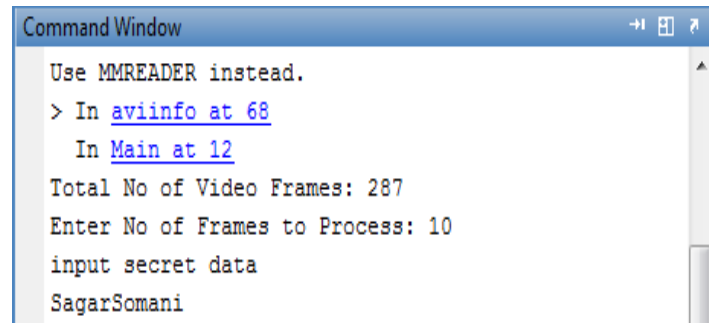
So, the proposed data hiding algorithm has been developed which will hide the secret text data in the encrypted compressed video signal, which preserves the confidentiality of the content.

The parameters such as mean square error (MSE), peak signal to noise ratio (PSNR), correlation and structural similarity index (SSIM) are evaluated to measure its efficiency.



Fig a. input video

Fig a. shows input video where user can choose .avi files which are to be processed.



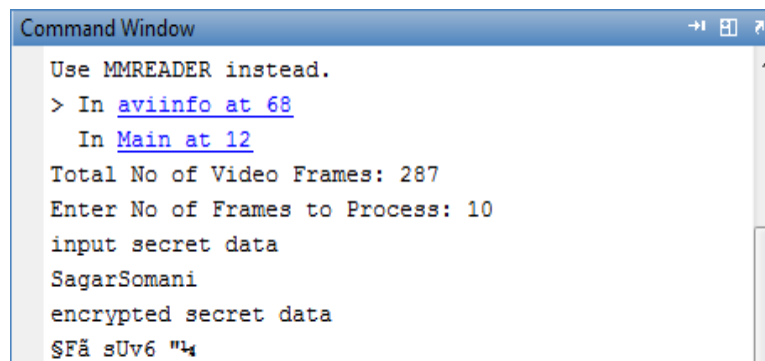
```

Command Window
Use MMREADER instead.
> In aviinfo at 68
   In Main at 12
Total No of Video Frames: 287
Enter No of Frames to Process: 10
input secret data
SagarSomani
  
```

Fig b. Frame count and input secret data

Fig b. shows frame count and secret data where when user selects .avi files to process, it shows total number of frames it contains and also user can select number

of frames to be processed and then the user selects the input secret text data and in this example we are taking input secret data as Sagar Somani.



```

Command Window
Use MMREADER instead.
> In aviinfo at 68
   In Main at 12
Total No of Video Frames: 287
Enter No of Frames to Process: 10
input secret data
SagarSomani
encrypted secret data
$Fä sUv6 "14
  
```

Fig c. Encrypted secret data

Fig c. shows encrypted secret data where the secret data file selected by user is encrypted by using chaos crypto system.



Fig d. Encrypted frames with secret data

Fig d. shows encrypted frames with secret data where the number of frames selected to process by user is embed with the input

secret data using bits substitution technique to prevent the confidentiality of the content.



Fig e. Reconstructed frames

Fig e. shows reconstructed frames where the receiver will receive the frames with the secret data and when user decrypts the data by using decryption key, user gets the

reconstructed frames back which were chosen to process and also the secret data send by the receiver.

```

Command Window
Use MMREADER instead.
> In aviinfo at 68
   In Main at 12
Total No of Video Frames: 287
Enter No of Frames to Process: 10
input secret data
SagarSomani
encrypted secret data
$Fã sUv6 "14
Secret text was hidden at compressed I frame
Raw Input File Size(in Bytes) :
    253440

Compressed File Size(in Bytes) :
    70928

Compression Ratio :
    3.5732

SagarSomani
Recovered Secret text:
SagarSomani
Mean Square Error :
    0.4236

Peak Signal to Noise Ratio(dB) :
    51.8615

Correlation :
    0.9998
    
```

Fig f. Recovered secret data and parameter calculation

Fig f. shows recovered secret data and parameter calculation where secret data send by sender to receiver is received after decryption and the parameters such as mean square error, peak signal to noise ratio, correlation and structural similarity index are evaluated to measure its efficiency.

Below table shows result of parameters calculated for secret data hiding in different videos for different quantization parameter(QP) values.

Six different videos are used with different frame size and same text data was embed into the videos and parameters named mean squared error (MSE), peak signal to noise ratio(PSNR), correlation, structural similarity index (SSIM) are calculated by setting different values of quantization parameter(QP).It has been observed that when quantization parameter(QP) is set to 15, better results are obtained.

Table 3.1 Result analysis for different Quantization Parameter (QP) values.

Video	QP	MSE	PSNR	Correlation	SSIM
Ball Video (50)	15	0.2943	53.4559	0.9995	0.9932
	24	1.2224	47.2692	0.9993	0.9775
	28	1.9969	45.1350	0.9993	0.9699
	32	3.4976	42.6971	0.9991	0.9607
Inp2 (127)	15	0.2132	54.9211	0.9994	0.9967
	24	1.2756	47.1807	0.9990	0.9814
	28	2.4429	44.4432	0.9985	0.9685
	32	4.7701	41.6955	0.9975	0.9524
Inp (30)	15	0.5104	51.0547	0.9993	0.9964
	24	3.1283	43.1789	0.9983	0.9820
	28	6.7720	39.8240	0.9972	0.9650
	32	14.8095	36.4269	0.9942	0.9323
Inp2(3) (287)	15	0.5039	51.1090	0.9992	0.9940
	24	2.7700	43.7077	0.9985	0.9736
	28	5.5127	40.7195	0.9975	0.9570
	32	10.8521	37.7783	0.9955	0.9344
Inp2 (60)	15	0.4565	51.5399	0.9993	0.9922
	24	2.3251	44.4709	0.9988	0.9668
	28	4.4832	41.6204	0.9977	0.9419
	32	8.7039	38.7414	0.9970	0.9035
Video2 (73)	15	0.4923	51.2129	0.9991	0.9923
	24	2.3831	44.3605	0.9978	0.9661
	28	4.3884	41.7091	0.9963	0.9415
	32	8.7162	38.7307	0.9935	0.8973

Setting quantization parameter value (QP) to a high value means that more coefficient are set to zeros, resulting in high compression at the expense of poor decoded image quality. Setting quantization parameter (QP) to a low value means that more non-zero coefficients remain after quantization, resulting in better decoded image quality but lower compression.

CONCLUSION

Thus an algorithm to embed additional data in encrypted H.264/AVC bit stream is presented, which consists of video encryption, data embedding and data extraction phases. The algorithm preserves the bit-rate exactly, even after encryption and data embedding, and is simple to implement as it is directly performed in the compressed and encrypted domain. Since data hiding is completed entirely in the encrypted domain, our method preserves the confidentiality of the content completely. With an encrypted video containing hidden data, data extraction can be carried out either in encrypted or decrypted domain, which provides two different practical applications. The proposed scheme, viz. encryption and data embedding preserves file-size, whereas the degradation in video quality caused by data hiding is quite small.

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) 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. Lett.*, vol. 19, no. 4, pp. 199–202, Apr. 2012.
- 6) X. P. Zhang, "Separable reversible data hiding in encrypted image," *IEEE*

-
- Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 826–832, Apr. 2012
- 7) 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.
- 8) X. P. Zhang, “Separable reversible data hiding in encrypted image,” IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 826–832, Apr. 2012.
- 9) 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.
- 10) Dawen Xu, Rangding Wang, and Yun Q. Shi, “Data Hiding in Encrypted H.264/AVC
- 11) Video Streams by Codeword Substitution,” IEEE Transaction on information forensics and security, vol. 9, no. 4, April 2014.