ISSN: 2321-7782 (Online)
Volume 2, Issue 1, January 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Paper
Available online at: www.ijarcsms.com

Quality Assessment of Precodec Video Protection

R. N. Mandavgane[1]
Dept. of Electronics & Telecommunication Engg
Bapurao Deshmukh engg. college, Sevagram Wardha
Nagpur - India

Dr. N. G. Bawane[2]
Principal
S. B. Jain engineering college
Nagpur - India

Abstract: ***Video authentication is very important in today's multimedia communication. For transmission of video, security is most vital. Encryption is one of the methods of protecting the video. Many experiments are going on for the security of video files. In this paper, a novel authentication scheme is used for video security. Here, first the YUV video file is converted to BMP files frame by frame. These BMP files generated are then modified by applying some logical operations. After modification, these BMP files are again converted back to a YUV file. This YUV file generated is not corrupted as the headers of the BMP files are retained. This modified YUV file is very different from the original YUV file, depending upon the logical operation used for modifying the BMP files and is used as an input to video CODEC for video compression. Again at the output of CODEC the conversion procedure is applied in reverse manner to get the original video. The Video quality is tested using the PSNR and fantastic results were obtained.***

Keywords: BMP files, logical operation, codec, PSNR.

I. INTRODUCTION

With the development of Internet and Computer technologies, multimedia data is widely used in communication. Communication can be in the form of audio as well as video or both. To keep the communication of video secure, it is protected before transmission. There must be some scheme which will protect the video from getting it intercepted by the unauthorized users. Video encryption is one of the methods on this issue. Many encryption algorithms such as DES, RSA, IDEA or AES are used by researchers. Most of these algorithms are for the text or binary data. These algorithms cannot be used for video encryption as volume of video data is very large.

Video encryption algorithms have been developed based on video CODECs in different research papers. In [1] [2] the raw data is encrypted directly. In [1] scan methodology is used to rearrange the pixel and the values of pixels have been changed. In [2] the Boolean matrix theory is used. The uncompressed video signal has been encrypted here. The encoding of .YUV file to .264 file gets affected a little as regards PSNR if the .YUV file is tampered with, which we've done here. There are some algorithms which use compressed data directly as in [3] for MPEG streams and are suitable for video storage. Other algorithms combine encryption process with compression process in which the sign of the DCT coefficients and motion vectors are encrypted [4] & [7]. Some algorithms use the encryption with variable length code as in [5] , [8].These algorithms are related to MPEG 1or 2 CODEC. In [9] DCT coefficients are partially permuted.

The concept of parameter sets are introduced in H.264 bit stream coding. As compared to MPEG codec, AVC produces the bit-stream with many more parameters. A Sequence parameter set describes the parameters for complete video sequence. The parameters such as profile and level, the size of the video sequence and maximum number of reference frames are main parameters of sequence parameter set which either depends on the video frame or on the programming. Hence the size of this set is not fixed. The next parameter set is the picture parameter set. This set contains the common parameters that may apply to a sequence or subset of coded frames, such as entropy coding type, number of active reference pictures and initialization

parameters. While encrypting the bit stream these parameters must be kept intact as decoder requires he information for decoding the video files. Then the frames are coded as one or more slices each contains slice header. Slice header have different parameter, e.g slice type depends on the I,P,bslices is one of them. This parameter is coded by Ex-Golomb code. Hence the size of this parameter is no fixed. Again this header also cannot be disturbed. Hence we have to look further for encrypting the video bit stream. The next bits determine the macroblocks of frames. In CAVLC encoding, parameters as the number of coefficients, trailing ones (coeff_token), the sign of each trailing ones, the levels of the remaining non-zero coefficients, the total number of zeros before the last coefficient and each run of zeros for luma and chroma are encoded respectively. Here the parameters which can be encrypted are the signs of trailing ones and levels of the remaining nonzero coefficients. From the above description as these parameters are discussed only for baseline profile, the generalized encryption program for the AVC bit stream is not that much of simple because this bitstream changes with format, size of video etc.

The AVC video stream is very parameter sensitive stream. It is not possible to encrypt parameters randomly. Residue data and motion vector differences in intra macroblock and intra prediction mode are encrypted in [6]. Encryption scheme along with AVC coding is presented and analyzed. Here it is essential to have a deep knowledge of AVC bit stream and AVC encoding for securing the video stream. The computing complexity depends on the data volumes to be encrypted, the cost of sub-key generation and the cost of the stream cipher.

The highlight of this work is that, dealing directly with the highly cryptic .264 file (encoded file) is avoided. The .264 alongwith the information file trace_enc.txt form a very complex combination which is very difficult to decipher and making any changes in the .264 file can be very dangerous.

In section 2, proposed encryption scheme along with the block diagram is given in detail. In section 3 encryption/decryption procedure is stated along with the figures and the flow chart. In section 4 the comparison of PSNRs of different video sequences for QCIF and CIF formats is shown and in section 5 a conclusion is drawn from it.

II. Proposed Encryption Scheme

In this scheme, we have first of all converted the YUV video file into bitmap files screen by screen. The bitmap files are the very easy to manipulate as compared to AVC bitstream. Any operation can be done on the binary files and the video can be encrypted. Now this encrypted video is again converted into YUV file. This new YUV file is then passed through H.264AVC. Again at the decoder end YUV video is converted back to bitmap file and reverse operation is applied on the bitmap files and then these bitmap files are again converted back to a YUV file to get the original video.

For that, we want to discuss the structure of a BMP file. The first 14 bytes are the header. The next 40 bytes are information bytes, followed by the actual data. These first 54 bytes should not be touched as the BMP file ceases to be recognized as a true BMP file. Fig 1 shows this structure.

14 bytes header	40 bytes information	Actual data (Pixel wise info)

Fig1 Structure of BMP file

Fig 2 below shows the block diagram for the above proposed scheme.

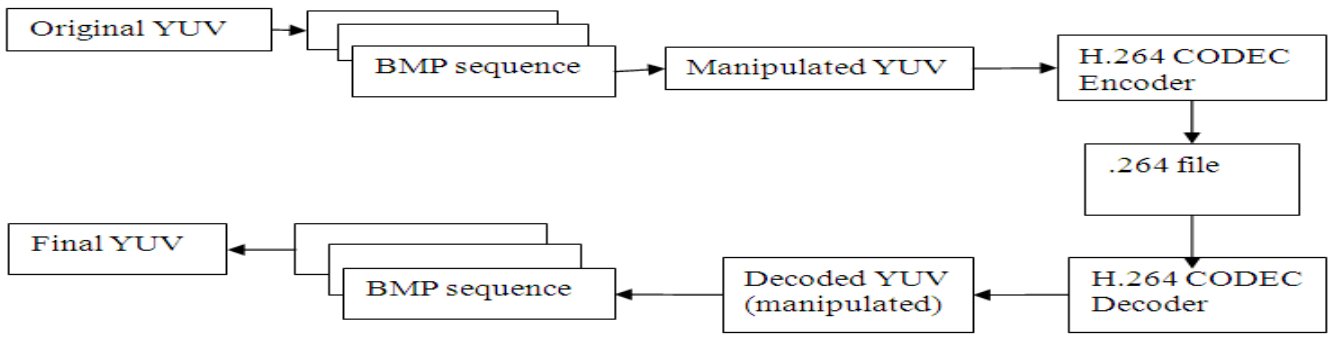

Fig 2 Block Diagram

For doing this process the tools required are YUV convertor which was downloaded from the internet, whereas, MATLAB software was used for encrypting the bitmap files. BMP file structure is simple as compared to AVC bitstream output file (.264 files). To manage the AVC bitstream file for modification is very tough task as the .264 files are very cryptic. Even if a single bit is altered non judiciously from the .264 file, the file ceases to be a true .264 file and decoding back to YUV is not possible. As mentioned earlier, the data section can only be encrypted. Where as in AVC, there are sequence parameter set, picture parameter set and slice header which are not fix sized. Also not every bit can be modified but we have to see the encoding tables for applying the encryption to these bits. This job is very difficult. To encrypt it at the time of encoding means we have to change the basic CODEC programming. As compared to this bitmap file can be used for encryption which is easier although the exercise of YUV to BMP to YUV conversions are involved in this process, at the cost of reduced PSNR. However, going by human eye sensitivity, not much appreciable change is observed.

III. Encryption/Decryption Procedure

Bitmap is encrypted using a software program. This program does not change the first 54 byte as it is the header size of the bitmap. The remaining bytes till the end of the files can be operated upon with some mathematical operation. The algorithm used for encryption in this case is flipping some bytes, i.e., reversing their order in the file for a set of bytes, sequentially progressing forward till the end of file

This operation is performed on all the BMP files as a sequence of frames and later, clubbed together to form a YUV file. The original YUV image and the YUV image after the encryption are shown in fig 3 and fig 4 below for the QCIF foreman Sequence.

Fig 3 original Image

Fig 4 Image after passed through MATLAB Program for two factors

This encrypted image is totally different from the original. Here the flow diagram for the encryption algorithm is given below in fig 5.

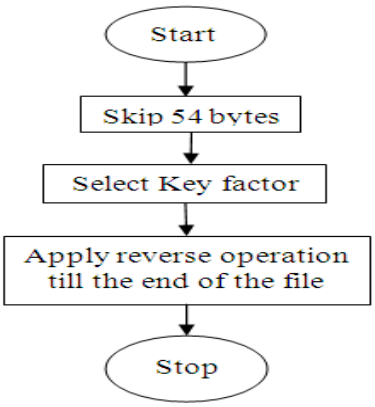

Fig 5 Encryption Algorithm

These encrypted videos are then passed through the sensitive AVC codec for the compression. If the compression is not required then the video can be sent in encrypted form and received at the other end which gives the original image with 100 % PSNR. At the decoder end these video file are again decrypted with a reverse algorithm. The video frames after the decoder and the decrypted video frames are shown in fig. 6 and fig 7 as below.

Fig. 6 Decoded image for factor one

Fig. 7 Image after passed through the MATLAB program

IV. COMPARISON

In this section, a comparison is made between the PSNRs for the two scenarios, one, for no encryption of the input files and secondly, for the encrypted file. Both encrypted and non encrypted video files are passed through the AVC CODEC and the final images after the decoder are obtained. These files are compared with each other as regards their PSNRs. Following are the results of a sequence we obtained through this work. The video sequences used are QCIF (Quarter Common Intermediate Format) and CIF with 30 frames /sec. The following results for the different sequences are obtained as Y PSNR as shown in fig. 8. It is observed that the Y PSNRs for the encrypted video for different YUV sequences are not more than the Y PSNRS of the original sequences but the difference between the two PSNRs is also very less. If the increment in number of bytes dealt with in a loop is increased the PSNRs get affected very little. As compared to Y (luma) PSNR the U and V (chroma) PSNRs get affected much more. This is because; a prediction block is formed from the previously encoded and reconstructed blocks. If the prediction of previously encoded and reconstructed block is done for the encrypted video, the next block shows more error as compared to prediction of the non encrypted blocks. The encryption for the video file is done by moving two pixels simultaneously because it gives better results as compared to encryption by moving single pixel. The results can be improved if the encryption is done by moving the blocks of 2x2 or 4x4 pixels as the encoder luma prediction block is formed by considering 4x4 blocks.

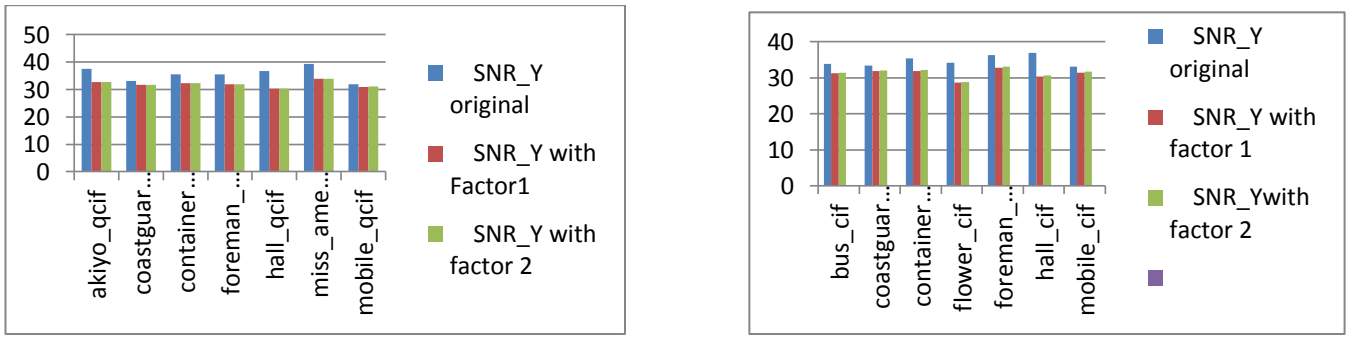

Fig 8 PSNR of Y for the QCIF and CIF sequences for two different factors

V. CONCLUSION

It can be concluded that if the videos are encrypted before passing them through the CODEC, its PSNR is not up to the mark. But this is at the cost of gaining security of the video. If we want a very high security, we get it at the cost of decreased value in PSNRs. These types of algorithms are applicable in intelligence applications wherein despite the reduced PSNR the information received is not much distorted so as not to be unintelligible. The most important thing I wish to highlight is that, one doesn't have to deal with the highly cryptic .264 (encoded) file. As mentioned earlier in 'INTRODUCTION', The trace_enc.txt which gives details of the encoded file (.264 file) generated by the encoder is very large and highly complex. Moreover, in h.264/SVC encoding, the problems get even more acute because of the various layers involved in the process. By the method we applied, we can get over this complex problem without bothering about .264 file.

References

1. S. Maniccam, and G. Nikolaos, "Image and video encryption using SCAN patterns," Pattern Recognition Vol. 37, No. 4, pp. 725-737, 2004.

2. X. Yi, C. Tan, C. Siew, and S. Rahman, "Fast encryption for multimedia," IEEE Transactions on Consumer Electronics, Feb. 2001, Vol. 47, No. 1, page(s): 101 – 107.
3. L. Qao, and K. Nahrstedt, "A new algorithm for MPEG video encryption," In Proceeding of the First International Conference on Imaging Science, Systems and Technology (CISST'97), pp. 21-29, Las Vegas, Nevada, July 1997.
4. C. Shi, S. Wang, and B. Bhargava, "MPEG video encryption in real-time using secret key cryptography," In Proc. of PDPTA'99,(Las Vegas, Nevada), pp. 2822-2828, 1999.
5. C. Wu, C. Jay Kuo, "Fast encryption methods for audiovisual data confidentiality," In SPIE International Symposia on Information Technologies 2000, Proceedings of SPIE Vol. 4209, pp. 284-295, (Boston, MA, USA), Nov. 2000.
6. S. Lian, Z. Liu, Z. Ren, and H. Wang, "Secure Advanced Video Coding Based on Selective Encryption Algorithms" In IEEE Transactions on Consumer Electronics, Vol. 52, No. 2,pp 612-629, MAY 2006.
7. F. Chiaraluce, L. Ciccarelli, E. Gambi, P. Pierleoni, and M. Reginelli, "A New Chaotic Algorithm for Video Encryption," IEEE Transactions on Consumer Electronics, Vol.48, No.4, pp. 838-844, Nov. 2002.
8. M. Kankanhalli, and T. Guan, "Compressed-domain scrambler/descrambler for digital video," IEEE Transactions on Consumer Electronics, May 2002, Vol. 48, No. 2, page(s): 356 – 365.
9. W. Zeng, and S. Lei, "Efficient frequency domain selective scrambling of digital video," IEEE Trans, Multimedia, Vol. 5, No. 1, pp. 118-129, 2003.
10. Iain E. Richardson, "The H.264 Advance Video Compression Standard" Wiley Second Edition
11. Martin Fiedler, "Implementation of a basic H.264/AVC Decoder" Chemnitz University of Technology, June 1, 2004.

