Video And Data Encryption Using AES Encryption For Secure Transmission

K.Kavuthami
Student, Department of CSE
Dhanalakshmi Srinivasan college of engineering and technology, Chennai.
kavuthami21@gmail.com

N.Thulasi, Research scholar,
St.Peter's university, Chennai.
AssistantProfessor, Dhanalakshmi Srinivasan college of engineering and technology, Chennai.
nara.thulasi@gmail.com

Dr.K.Thirunadana Sikamani,
Professor and Head,
St.Peter's college of engineering and technology, Chennai.

Abstract— In This paper, video and data is encrypted using AES algorithm for secure transmission. Steganography is used for embedding messages into video or image to transfer a large amount of secret message. Huffman coding is a technique for lossless data compression and reduction of file size. Huffman coding is used to encode images in a video file. Then the encoded embedded file is encrypted using AES encryption. The process is repeated until the search is done all over the frame. It is decrypted only when the user knows the key. The experimental result shows that the degradation of visual quality is less. The performance of the process is measured by calculating Peak signal to noise ratio, Mean square error rate and Bit error rate.

Index Terms— *AES Encryption, Data Hiding, Encoding, Embedding data, Steganography, zigzag.*

1. Introduction

Image processing refers to convert an image into digital form and perform some operations, to get an enhanced image or to extract some useful information and it is a type of signal dispensation in which input is video, image, photograph and output may be image or characteristics associated with that image. Encryption is the process of encoding messages in such a way that only authorized parties can only read it.

Encryption doesn't prevent hacking but it reduces the likelihood that the hacker will be able to read the data that is encrypted. H.264 or MPEG-4 Part 10, Advanced Video Coding (MPEG-4 AVC) is a video compression format that is currently one of the most commonly used formats for the recording messages; compression of video and distribution of video content. H.264 is used for lossy compression in strict mathematical sense; the amount of loss may sometimes be reduced. It is also possible to create truly lossless encodings and encryption of videos.

Data hiding is a technique for embedding information or secret messages into covers for example image, audio, and video files used for integrity authentication, covert communication. Most data hiding methods include embedding messages into the cover media to generate the media by only modifying the least significant the least significant part of the cover and thus ensure perceptual transparency.

2. Related Works

In previous work, the encryption, data embedding and decryption of videos are processed.

In [1] recently, more and more attention is paid to reversible data hiding (RDH) in encrypted images; since it maintains the Excellent property in which the original cover can be losslessly recovered after embedded data is extracted while protecting the image content is in confidentiality. Previous methods embed data by reversibly vacating room from the encrypted images, which may cause some errors on data extraction and/or image restoration. A novel method by reserving room before encryption with a traditional RDH algorithm, which is easy for the data hider to reversibly embed data in the encrypted image. The proposed method implies achieve real reversibility of data extraction and image recovery is free from any error. Experiments show that the novel method can embed more than 10 times as large payloads for the same image quality as the previous methods, for example PSNR dB.

In [2], Digital communication has become an essential part of infrastructure now-a-days and the advancement in the field of digital communication in other sense has hiked the fear of getting the data snooped at the time of sending it from the sender to receiver. Information Security is becoming an inseparable part of digital Communication. The Steganography approach uses the flexible macro block ordering feature of H.264/AVC to hide message bits. The message to be hidden is encrypted using AES algorithm. The message is hidden twice and predicted using the transitive, reflexive, symmetric properties. The proposed method here evades the copyright infringement and makes the stego video immune to steganalysis.

In [7], an improved version of Zhang's reversible data hiding method in encrypted images. The work partition in which encrypted image into blocks and each block carries one bit by flipping three LSBs of a set of predefined pixels of image. The data extraction and image recovery can be examined by the block smoothness. Zhang's work does not fully exploit the pixels in calculating the smoothness of each block and does not consider the €very good encryption results focusing towards the security against statistical attacks.

In [14], Reversible watermarking enables the embedding of useful information in a host signal without any loss of host information. Ian's difference-expansion technique is a high-capacity, reversible method for data embedding. However, the method suffers from undesirable distortion at low embedding capacities and lack of capacity control due to the need for embedding a location map. We propose a histogram shifting technique as an alternative to embedding the location map. The proposed technique improves the distortion performance at low embedding capacities and mitigates the capacity control problem. We also propose a reversible data-embedding technique called prediction-error expansion. This new technique better exploits the correlation inherent in the neighborhood of a pixel than the difference-expansion scheme. Prediction-error expansion and histogram shifting combine to form an effective method for data embedding. The experimental results for many standard test images show that prediction-error expansion doubles the maximum embedding capacity when compared to difference expansion. There is also a significant improvement in the quality of the watermarked image, especially at moderate embedding capacities.

3. Proposed System

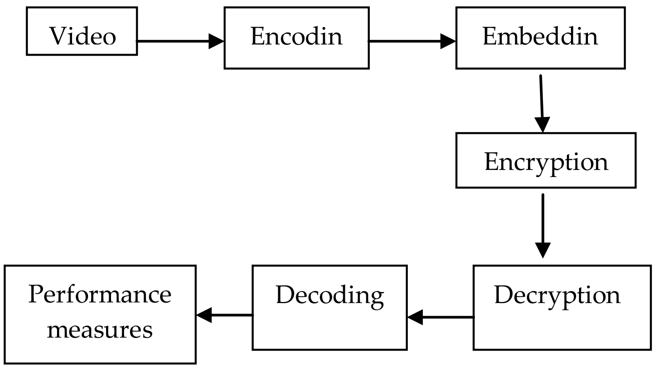


Fig 1.1 System architecture

The videos were converted into frames. The frames were first encrypted by employing Huffman encoding technique. The key is generated and based on the key generated the image region were divided and the pixels values were modified according to the key generated based on the cipher exchanging process in stream cipher. The original data that is to be hided in the encrypted videos were then embedded by using code word substitution technique. The code word generated for the message is placed in the original image pixels so that the information is hided in the video frames. The images were encrypted with the help of AES and zigzag encryption methods. In the receiver side the video the original message is extracted and then the video is decrypted to get the original video. The performance of the process is measured by the calculation of the Bit Error Rate value.

3.1 Encoding

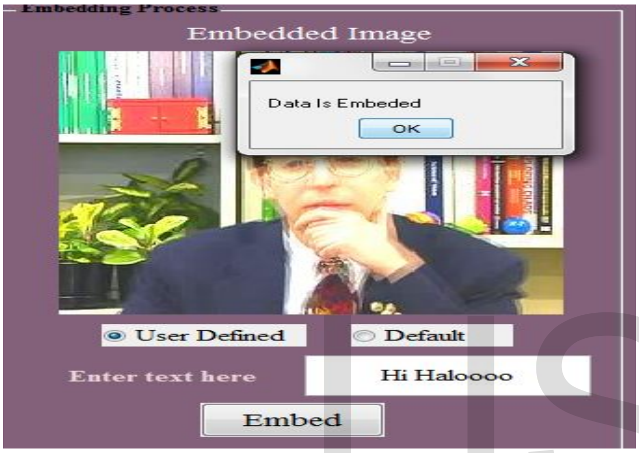
If the symbols are sorted by probability, there is a linear-time (O(n)) method to create a Huffman tree. Huffman tree is created by using two trees, in which first one containing the initial weights (along with pointers to the associated leaves), and combined weights (along with pointers to the trees) being put in the back of second queue, the lowest weight is always kept at the front of one of the two queues Start with as many leaves as there are symbols. Enqueue all leaf nodes into the first queue (by probability in increasing order so that the least likely item is in the head of the queue).While there is more than one node in the queues: Dequeue the two nodes with the lowest weight by examining the fronts of both queues. Create a new internal node, with the two just-removed nodes as children (either node can be either child) and the sum of their weights as the new weight .Enqueue the new node into the rear of the second queue. The remaining node is the root node; the tree has now been generated. Preened the Huffman tree, bit by bit, to the output stream for encoding. For example, assuming that the value 0 represents a parent node and 1 which represents a leaf node, whenever the latter is encountered the Huffman tree building routine simply reads the next 8 bits to determine the character value of that specified leaf. The process continues recursively until the last leaf node is reached to that process; the Huffman tree will thus be faithfully reconstructed.

Frames are Encoded

Fig 1.2 Encoding

3.2. Embedding

An embedded file refers to any type of multimedia file that might be inserted, or embedded into the Web page or video. This includes files like graphics and sound files. The pixels in the image are shifted and the data is embedded in the block using some embedding function. The embedding will convert the input data into number and it replaces the block in the image pixels with it. After embedding the area where the data is hided is mapped. The secret message is embedded into a video file. The videos are converted into frames and encoded, and then data is embedded. Shifting the pixels of each image of a video file which is encoded. In LSB, the embedded messages are saved.



Data Is Embeded

● User Defined ○ Default

Enter text here Hi Haloooo

Fig 1.3 Embedding

3.3 Encryption

The Advanced Encryption Standard or AES is a symmetric block cipher to protect classified information and is implemented to encrypt sensitive data. AES is the usage of the two different keys for encryption and decryption. The key is added to the frames than the row values were shifted and the columns were mixed. In the decryption process also the same shift row and mixed column technique were employed. The message will be more secure when compared to the other techniques. AES is based on a design principle known as a substitution-permutation network, combination of both substitution and permutation and fast in both software and hardware components. AES operates on a 4×4 column-major order Shift rows cyclically shifts the bytes in each row by a certain offset. In the Mix Columns step, the four bytes of each column of the state are combined using an invertible linear transformation. In the AddRoundKey step, the sub key is combined Transform it into the final output of cipher text.To Encrypt, Decrypt the image pixel, 7 Bits is the input and output which communicates with the 128 bit AES algorithm. We have to shift the 7 bit input to 128 bit register and feed it to the AES Encryption algorithm to get the Encrypted 128 bit data and from the 128bit register shift 7 bit data to Image per clock to get the encrypted image. To get back the original image feed the Encrypted image to the AES Decryption algorithm. In AES, the key size is calculated by image size.

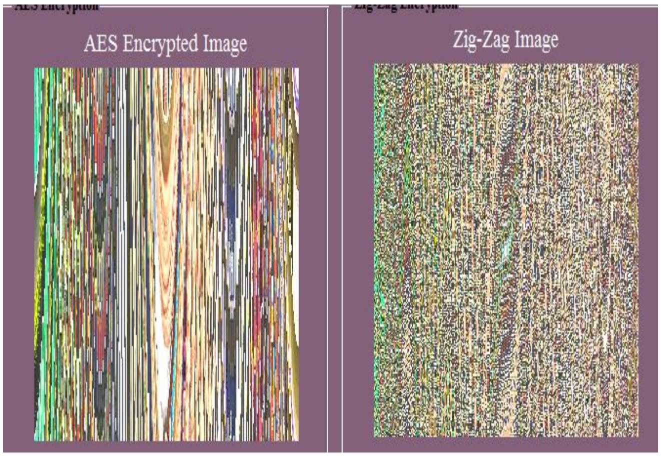


Fig 1.4 Encryption

3.4 Decryption

For AES Decryption, the same encryption process occurs simply in reverse order. The decryption block, the encryption parameters are the input text, the key and the output text is same as the encryption input. In decryption the key schedule is same in the encryption; the only operations we need to implement are the Inverse sub Bytes, shift Rows and mix Columns, while addRoundKey stays the same for the process.

The zig zag is reversed in order to obtain the AES encrypted pixels. If the zigzag transposition is done row wise, then the pixels were read in zigzag fashion based on the digits in the key. The AES encryption is reversed in order to obtain the Huffman encoded embedded image. While decryption using AES the steps were reversed.

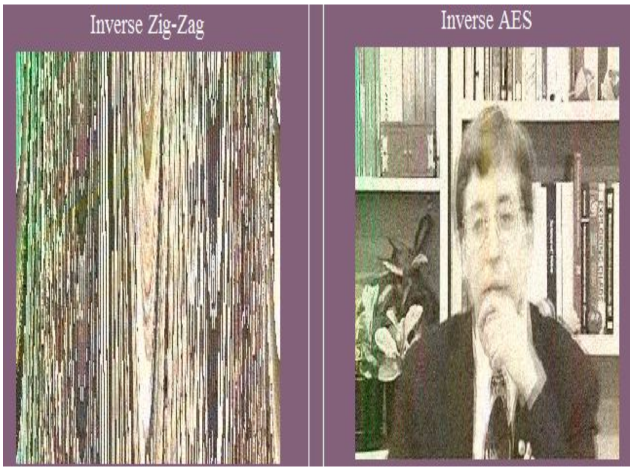


Fig 1.5. Decryption

3.5 Performance measures

The Bits per Pixel value is calculated which gives the performance of our algorithm. The result shows that the embedding rate of the proposed algorithm is high. The cover video frame is also retrieved with minimum distortion. The PSNR value and the MSE value indicate the accurate retrieval of the encoding video frames.

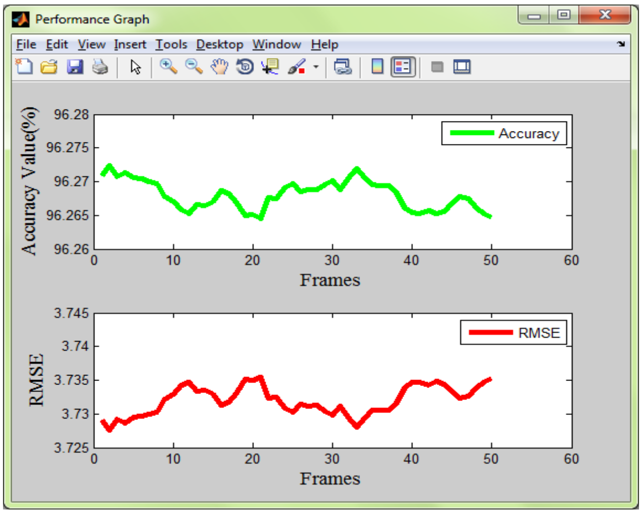


Fig 1.6 graph

*F.*Comparison between the existing system and proposed system performances

In existing system, the videos which are converted to frames are encrypted and data embedded in a single file. Using stream cipher for encryption, codeword substitution for data embedding. The accuracy and root mean square error rate of frames after decryption is not clear. The percentage of accuracy and RMSE is shown in fig.1.6 and fig 1.8.

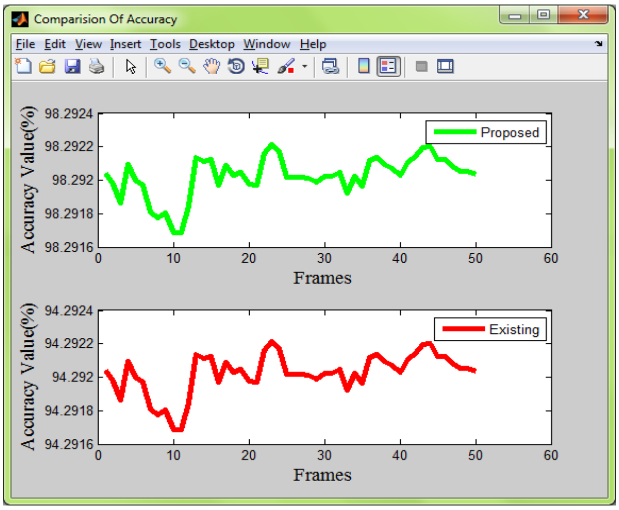


Fig 1.7.Accuracy

In Proposed system, the videos which are converted to frames are encrypted and data embedded in a single file. Using AES, zigzag for encryption and decryption. The accuracy and root mean square error rate of frames after decryption is clear. The percentage of accuracy and RMSE is shown in fig.1.6 and fig 1.8.

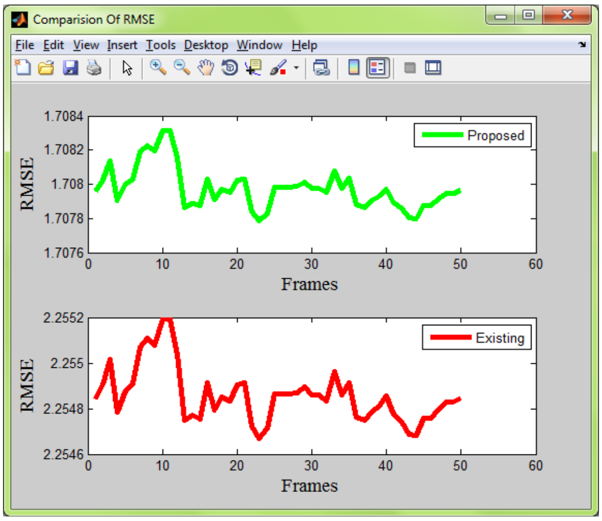


Fig 1.8.RMSE

4.CONCLUSION

In this paper, the proposed method for transmitting the secured message and video. A data hiding process that is combined with encoding and encryption is proposed. The secret informations were placed in the images The encryption is then employed using two different types of encryption in order to improve the performance. The retrieval process is done by reversing the same process that were used for encryption, embedding and encoding. The performance of the process is finally measured which indicates that the proposed method is more efficient and secure compared to the existing work. The performance of AES Encryption shows the frames are in sufficient display. The experimental result shows that the video and secret data can be encoded and decoded. Therefore, we conclude that using AES is sufficient way for secure transmission of videos and messages. In future work, we will pay our attention to transmitting the secret file to one place to another place.

5 .FUTURE SCOPE

The enhancement can be done by encrypting the secret information using Advanced Encryption Standard encryption method. In the existing work the secret message is hided as it is in the image. Now in order to improve the performance further the secret information is encrypted. The encryption process increases the security of the secrete message.

REFERENCES

[1] 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.

[2] Lathikanandini. M, Suresh.J,”Steganography in mpeg video files using macroblocks”, International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970) Volume-3 Number-1 Issue-8 March-2013.

[3] 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.

[4] Weiming Zhang, Biao Chen, and Nenghai Yu,” Improving Various Reversible Data Hiding Schemes Via Optimal Codes for Binary Covers”,IEEE trans on image processing, vol. 21, no. 6, june 2012 .

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] M. N. Asgar 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.

[13] Diljith M. Thodi and Jeffrey J. Rodriguez,” Expansion Embedding Techniques for Reversible Watermarking”, IEEE Transactions on image processing, vol. 16, no. 3, march 2007.

[14] X. Wang, X. Li, B. Yang, and Z. Guo, “Efficient generalized integer transform for reversible watermarking,” *IEEE Signal Process. Lett.*,vol. 17, no. 6, pp. 567–570, Jun. 2010.

[15] Y. Hu, H. K. Lee, and J. Li, “DE-based reversible data hiding with improved overflow location map,” IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 2, pp. 250–260, Feb. 2009.

[16] W. Hong, T. S. Chen, and C. W. Shin, “Reversible data hiding for high quality images using modification of prediction errors,” J. Syst. Softw.,vol. 82, no. 11, pp. 1833–1842, Nov. 2009.

[17] 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.

[18] I. E. G. Richardson, H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia. Hoboken, NJ, USA: Wiley,2003.

[19] 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.

[20] 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.

[21] 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.

[22] 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.