A Novel Steganographic System for Data Hiding in Video/Audio

M.Jyotheeswari
M.Tech Student
Department of CSE
K.S.R.M. College of Engineering
Kadapa, Y.S.R. District., A.P. (India)

V. Lokeswara Reddy
Associate professor
Department of CSE
K.S.R.M. College of Engineering
Kadapa, Y.S.R. District., A.P. (India)

ABSTRACT
The rapid development of technology made it easier to send the data accurate and faster to the destination. The most important factor of information technology and communication is the security of the information. This security can be achieved through steganography. Steganography is art and science of invisible communication. This paper mainly focuses on audio and video steganography. Video Steganography hides secret data within a video and audio steganography deals with hiding secret data within audio. The secret data is first compressed, encrypted and then embed into the cover frames in such a way that eight bits of the secret data are divided into 2, 2, 2, 2 and then embedded into the RGB (Red, Green, Blue) pixel values of the cover frames respectively and the remaining 2 bits are inserted in the next pixel of cover frame and so on. The proposed technique is compared with existing LSB (Least Significant Bit) based steganography and the results are found to be encouraging.

Keywords
Video Steganography, Audio steganography, cover frame, secret data, LSB.

1. INTRODUCTION
Steganography is a process of hiding secret data within a carrier in invisible manner. It is derived from a Greek word steganos, which means covered or secret, and graphy (writing or drawing) [1]. The medium where the secret data is hidden is called as cover medium. The cover medium can be image, video or an audio file. Hiding the secret data inside the cover medium results in stego.The stego file contains both cover medium and secret data. The steganography takes advantage over cryptography. In cryptography by looking at the data itself hacker knows that it has been encrypted, so by doing some cryptanalysis he can easily get the secret data, but in steganography, the hacker couldn't identify that, a secret data has been embedded as it allows invisible communication.

Steganography is the science of writing hidden data inside the cover media in such a way that no one except the sender and the intended recipient can realize that there is a hidden data. This subject has been brought into the public attention by intelligence agencies and news media. Now a day's cryptography is combined with steganography to provide added security. While cryptography provides privacy, Steganography is intended to provide secrecy. Steganography works to mask the very existence of the message. Applications of Steganography vary from military, industrial applications to copyright and Intellectual Property Rights (IPR). The messages can be sent and received securely by using lossless steganography techniques [2]. Traditionally, steganography was based on hiding secret information in the image files. But modern work suggests that there has been growing interest among research fraternity in applying steganographic techniques to video files as well [3], [4].

The advantage of using video files in hiding information is the added security against the attack of hacker due to the relative complexity of the structure of video compared to image files. The advantage in the method is that the amount of data (payload) that can be embedded is more in LSB techniques. However most of the LSB techniques are prone to attacks are described in [5], [6]. The audio steganography also plays a good role in providing security. This makes research fraternity interested in designing new methods. The results obtained are found to be significant and encouraging.

2. PROPOSED SYSTEM
The overview of the proposed system is shown in figure 1.

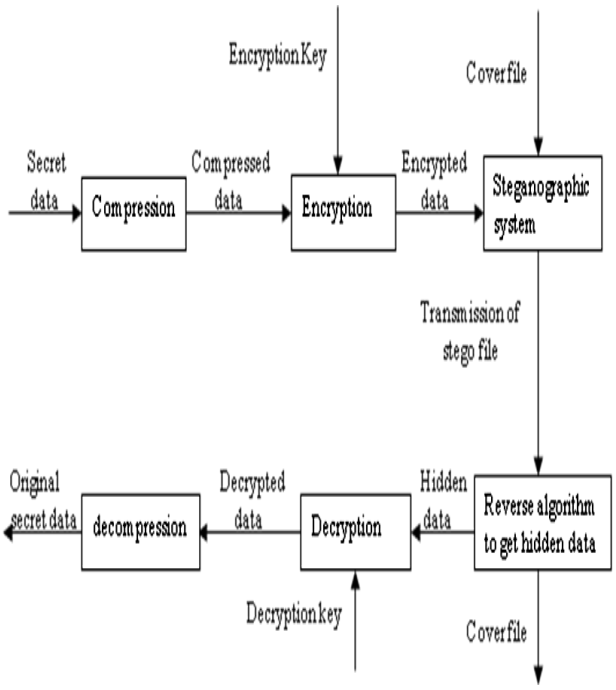


Fig 1: Overview of the proposed system

The cover file is broken down into multiple frames and then the proposed LSB based technique is 6applied to hide the secret data in the cover file frames. The size of the secret data does not matter as it can be compressed and then embedded in multiple frames of audio or video. The proposed system provides the added security for the secret data by encrypting it. So to embed a secret data within the cover file uses the following three steps:

(1) Compress the secret data
(2) Encrypt the compressed data
(3) The encrypted data is then embedding in the cover media.

Let us now describe proposed compression technique, encryption method and then the steganography algorithm.

2.1 LZW Technique

This is an algorithm for lossless data compression and decompression. The LZW (It has been given that name on the name of the authors: Jacob Ziv, Abraham Lempel and Terry Wellch) a dictionary-based compression algorithm that maintains an explicit dictionary [7]. The code words output by the algorithm consist of two elements: an index referring to the longest matching dictionary entry and the first non-matching symbol. In addition to outputting the codeword for storage or transmission, the algorithm also adds the index and symbol pair to the dictionary. When a symbol that not yet in the dictionary is encountered, the codeword has the index value 0 and add it to dictionary.

2.1.1 Compression algorithm

A quick examination of the algorithm shows that LZW is always trying to output codes for strings that are already known. And each time a new code is output, a new string is added to the string table.

Input: Text file
Output: Compressed file

Step 1: Start
Step 2: Initialize 'S' as an empty string
Step 3: While there is still data to be read continue
Step 3.1: Read the text file character by character and store it in the variable 'ch'
Step 3.2: If S+ch already presents in the dictionary then
Step3.2.1: S=S+ch
Else
Step3.2.2: Encode S to output file
Step3.2.3: Add S+ch to the dictionary
Step3.2.4: Assign S=ch
End while loop
Step4: End

2.1.2 Decompression algorithm

The decompression algorithm is used to get back the original text file from the compressed file. It takes the stream of codes output from the compression algorithm, and uses them to exactly recreate the input. The LZW algorithm is efficient because it does not need to pass the dictionary to the decompression code. The dictionary can be built exactly as it was during compression by using the input stream. This is possible because the compression algorithm always outputs the string and character components of a code before it uses it in the output stream.

Input: Compressed file
Output: Original text file

Step 1: Start
Step 2: Read OLD_CODE
Step 3: output OLD_CODE
Step 4: While there is still data to be read continue
Step 4.1: Read NEW_CODE
Step 4.2: S = get the translation of NEW_CODE
Step 4.3: output S
Step 4.4: ch = get first character in S
Step 4.5: Add OLD_CODE + ch to translation table
Step 4.6: OLD_CODE = NEW_CODE
End while loop
Step 5: End

2.2 AES Encryption/Decryption algorithm

The secret data is encrypted before embedding. Encryption is a process of converting the plain text (original text) into cipher text (unreadable format) with the help of a key. The encryption technique used in the proposed system is AES (Advanced Encryption Standard) [8]. It is asymmetric-key algorithm, meaning that the same key is used for both encrypting and decrypting the data.

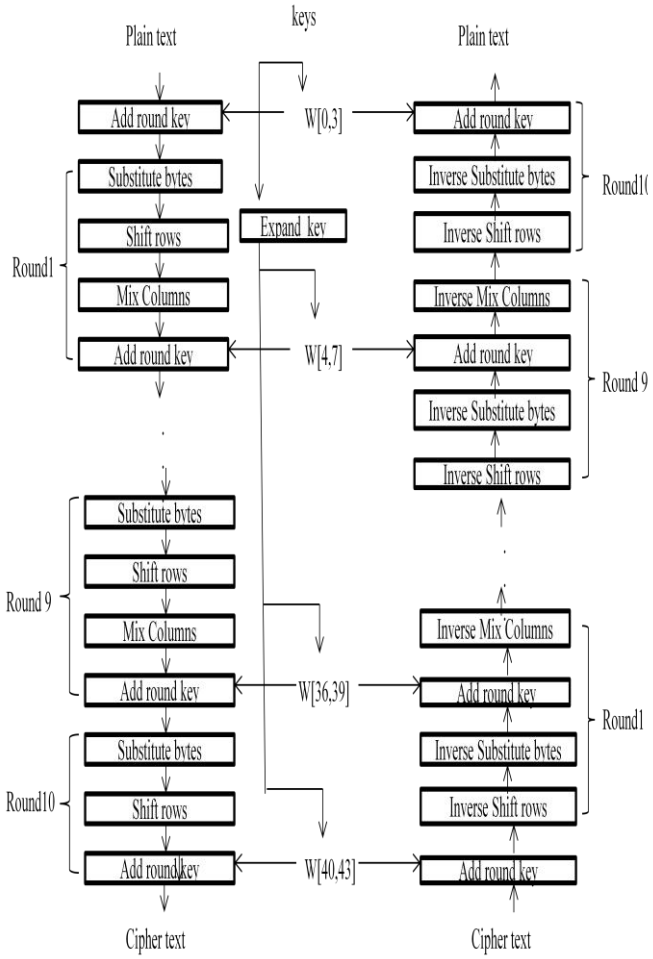
AES is based on a design principle known as a substitution-permutation network, and is fast in both software and hardware. It has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. Depending on which version is used, the name of the standard is modified to AES-128; AES-192 or AES-256 respectively.AES operates on a 4×4 column-major order matrix of bytes, termed the state. The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the plaintext into cipher text. The number of cycles of repetition is as follows:

10 cycles of repetition for 128-bit key.

12 cycles of repetition for 192-bit key.

14 cycles of repetition for 256-bit key.

The overall structure of the AES algorithm is shown in figure 2. It uses the key size of 128 bit, so there will be 10 cycles of repetition. The input is a single 128 bit block both for decryption and encryption and is known as the 'in' matrix. This block is copied into a 'state' array which is modified at each stage of the algorithm and then copied to an output matrix. The algorithm begins with an Add round key stage followed by 9 rounds of four stages and a tenth round of three stages. The tenth round at encryption/decryption simply leaves out mix columns/inverse mix columns stage. This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of its counterpart in the encryption algorithm.



(a) Encryption (b) Decryption

Fig 2: Overall structure of AES algorithm

2.2.1 Encryption algorithm

The encryption algorithm has the following steps:

Step 1: Key Expansion—round keys are derived from the cipher key using Rijndael's key schedule.
Step 2: Initial Round
- Step 2.1: AddRoundKey—each byte of the state is combined with the round key using bitwise xor.

Step 3: Rounds
- Step 3.1: SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table(s-box).
- Step 3.2: ShiftRows—a transposition step which performs circular left shift operation.
- Step 3.3: Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
- Step 3.4: Add Round Key

Step 4: Final Round (no Mix Columns)
- Step 4.1: SubBytes
- Step 4.2: ShiftRows
- Step 4.3: AddRoundKey

2.2.2 Decryption algorithm

The decryption is a process of converting the cipher text into plain text (original text).The decryption algorithm has the following steps:

Step 1: AddRoundKey—each byte of the state is combined with the round key using bitwise xor.

Step 2: Rounds
- Step 2.1: InverseShiftRows—a transposition step which performs circular right shift operation.
- Step 2.2: InverseSubsitutebytes— a substitution step where each byte is replaced with another according to a lookup table (inverse s-box).
- Step 2.3: AddRoundKey
- Step 2.4: InverseMixColumns—is a reverse operation of MixColumns

Step 3: Final Round (no InverseMixColumns)
- Step 3.1: InverseShiftRows
- Step 3.2: InverseSubsitutebytes
- Step 3.3: AddRoundKey

2.3 Embedding/De-Embedding Process

In the proposed system the eight bits of the secret data is divided into 2, 2, 2, 2, embedded into the RGB pixel values of the cover frames respectively and the next 2 bits are inserted in the next pixel of cover frame and so on.

2.3.1 Embedding process

It is a process of hiding the secret data within the carrier file. If the cover frame bytes are as follows:

Byte1	Byte2	Byte3	Byte4
00100111	11101001	11001000	00100111

Byte 5	Byte 6 …….
11001000	11101001 …..

And a byte of secret data to be hidden is 10001100 then the embedding process is shown in figure 3.

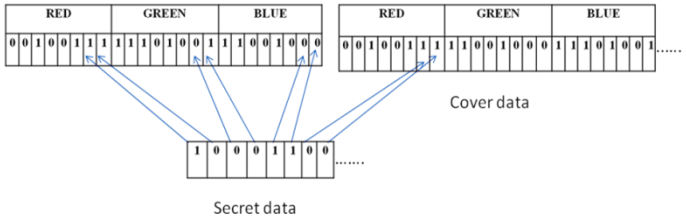


Fig 3: Embedding process

After hiding, the cover frame bytes will be as follows:

Byte1	Byte2	Byte3	Byte4
0010011**0**	1110100**0**	1100100**11**	0010010**0**

Byte 5	Byte 6…..
11001000	11101001 …..

2.3.2 De-Embedding process:

It is a process of extracting the hidden data from stego file (contains both cover file and secret file).Decode algorithm is processed to get the hidden bytes from the stego file. The

extraction process takes the stego file and the outputs the text file.

2.3.3 Encoding algorithm at sender side

Step 1: Select the secret data to be hidden.
Step 2: Compress and encrypt the secret data.
Step 3: Select video or audio in which the secret is to be embedded.
Step 4: Embed the secret information.
Step 5: Transfer file to the receiver.

2.3.4 Decoding algorithm at receiver side

Step 1: Receive stego file from the server.
Step 2: Extract the hidden data.
Step 3: Decompress the secret data
Step 4: Decrypt with the help of key.
Step 5: Get original secret data.

3. EXPERIMENTAL WORK

When the system is executed GUI (Graphical User Interface) is displayed. The snapshot of the main window is as shown in figure 4. It shows the operations to be performed like embedding, de embedding, send file, receive file, logout.

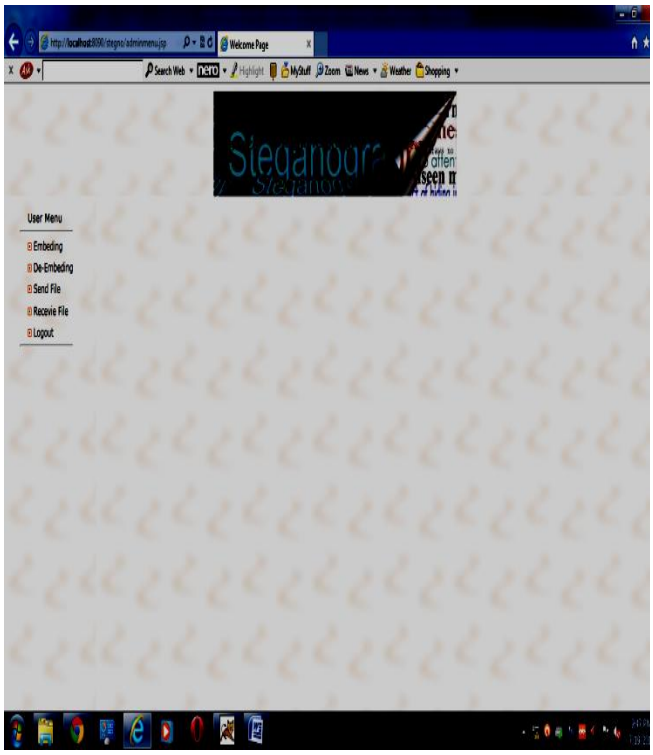


Fig 4: Snapshot of main window

The Snapshot of embed window is shown in figure 5.To embed data click on 'embedding'. The Embed window provides a provision for choosing cover medium (video, audio, image) and it also provides an option for selecting secret message file. The secret message file is text. Select the compression technique (LZW) and then mark the AES check box. Finally press submit button to embed the message in cover file.

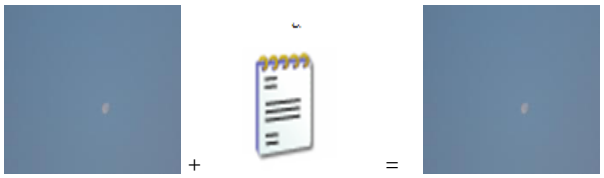
Fig 5: Snapshot of embedding window

The snapshot of de embed window is shown in figure 6.To get the original secret data click on 'deembedding'. The de embed window provides a provision for choosing stego file. Select the compression technique (LZW) and then mark the AES check box. Finally press submit button to get the secret data.

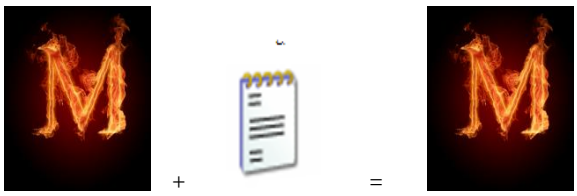
Fig 6: Snapshot of deembedding window

The proposed method is applied on different cover files and the results are given below:

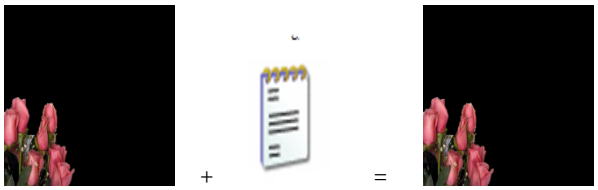
Case 1: Cover medium=image (jpg, gif, png, tif, bmp)
Secret data=text file



Cover file Abc.jpg Size=1.70MB | Text file xyz.txt size=22bytes | Stego file embedabc.jpg size=1.70MB



Cover file m.gif Size=1.44 MB | Text file xyz.txt size=22bytes | Stego file embedm.gif size=1.44MB



Cover file Flower.bmp Size=4.88 MB | Text file size=22bytes | Stego file xyz.txt embedflower.bmp size=4.88MB

Case 2: Cover medium=audio (mp3,wav)
Secret data=text file

Cover file Suklam.mp3 Size=197KB | Text file text.txt size=23bytes | Stego file Suklam.mp3 size=197KB

Cover file drmapan.wav Size=824KB | Text file text.txt size=23bytes | Stego file drmapan.wav size=824KB

Case 3: Cover medium=audio (mp4, avi, wmv)
Secret data=text file

Cover file M4H08027.MP4 Size=13.3MB | Text file text.txt size=23bytes | Stego file M4H08027.MP4 size=13.3MB

Cover file flame.avi Size=282KB | Text file text.txt size=23bytes | Stego file flame.avi size=282KB

Cover file news_interview_audio.wma Size=965KB | Text file text.txt size=23bytes | Stego file news_interview_audio.wma size=965KB

4. CONCLUSION

The proposed technique is applied on different file formats of image, audio and video files. In the proposed technique secret message is first compressed, encrypted and then embed in cover file with the help of steganographic system. It can enhance the confidentiality of information. Performance analysis of the proposed technique after comparison with enhanced LSB technique is quite encouraging. It can be further extended with multi file embedding.

5. REFERENCES

[1] E. Cole and R.D. Krutz, Hiding in Plain Sight: Steganography and the Art of Covert Communication, Wiley Publishing, Inc., ISBN 0-471-44449-9, 2003.

[2] Stefan Katzenbeisser and Fabien A. P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House Books, ISBN 1-58053-035-4, 1999.

[3] D. Stanescu, M. Stratulat, B. Ciubotaru, D Chiciudean, R. Cioarga and M. Micea, Embedding Data in Video Stream using Steganography, in 4th International Symposium on Applied Computational Intelligence and Informatics, SACI-2001, pp. 241-244, IEEE, 2007.

[4] Feng Pan, Li Xiang, Xiao-Yuan Yang and Yao Guo, Video Steganography using Motion Vector and Linear Block Codes, in IEEE 978-1-4244-6055-7/10/, pp. 592-595,2010.

[5] A. Westfield, and A. Pfitzmann, Attacks on Steganographic Systems, in Proceedings of 3rd Info.Hiding Workshop, Dresden, Germany, Sept. 28−Oct. 1, pp. 61-75, 1999.

[6] J. Fridrich, R. Du, and L, Meng, Steganalysis of LSB Encoding in Color Images, in Proceedings of ICME 2000, Jul.-Aug. 2000, N.Y., USA.

[7] ZIV, J., AND LEMPEL, A. A universal algorithm for sequential data compression. IEEE Transactions on Information Theory 23, pp.337–343, 1977.

[8] "Distinguisher and Related-Key Attack on the Full AES-256". Advances in Cryptology – CRYPTO 2009. Springer Berlin / Heidelberg. pp. 231–249.

AUTHORS BIOGRAPHY

M.Jyotheeswari did her B.Tech (CSE) from JNTUA, Anantapur in the year 2011. She is pursuing her M.Tech (CSE) from JNTUA Anantapur, Andrapradesh. She is currently doing her M.Tech in K.S.R.M College of Engineering, Kadapa.

V. Lokeswara Reddy did his M.Tech (CSE) from SRM University, Chennai in the year 2005. He did his M.C.A from S.V. University, Tirupati in the year 2000. He is pursuing his Ph.D from JNTUA, Anantapur. He has a total of 12 years of experience in teaching. Currently he is working as Associate Professor at K.S.R.M College of Engineering, Kadapa. He has presented 9 papers in International, National Conferences and published 9 papers in International journals.