



IJITCE

ISSN 2347- 3657

International Journal of

Information Technology & Computer Engineering

www.ijitce.com



Email : ijitce.editor@gmail.com or editor@ijitce.com

SECURING ALL MEDIA DATA USING CRYPTOGRAPHY AND STEGANOGRAPHY TECHNIQUES

RAYA PAVANKUMAR¹

Assistant Professor, Department of CSE, ABR College of Engineering and Technology
Kanigiri, Prakasam, Andhra Pradesh

KUPPALLA UTTEJ KUMAR², NARSI REDDY YAGGONU³, BOTTUMANCHI RAJKUMAR⁴, KUDUMU GURUPRAKASH⁵, KOLLIPARA KISHORBABU⁶

^{2,3,4,5,6} IV B.Tech. Students, Department of CSE (Data Science), ABR College of Engineering and Technology

Kanigiri, Prakasam, Andhra Pradesh

1

2

3

ABSTRACT

The rapid development of technology Made it less difficult to send the information correct and rapid to the destination. The essential factor of knowledge technological know-how and communication is the protection of the information. This safety will also be achieved via steganography. Steganography is artwork and science of invisible communication. This paper customarily specializes in audio and video steganography. Video Steganography hides secret knowledge within a video and audio steganography deals with hiding secret knowledge within audio. The key data is first compressed, encrypted and then embed into the cover frames in this sort of approach that eight bits of the secret data are divided into 2, 2, 2, 2 after which embedded into the RGB (red, green, Blue) pixel values of the cover frames respectively and the remaining 2 bits are inserted within the subsequent pixel of canopy frame and many others. The proposed method is when compared with current LSB (Least giant 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 system of hiding secret data within a service in invisible manner. It is derived from a Greek word steganos, which means that covered or secret, and graphy (writing or drawing) [1]. The medium the place the key knowledge is hidden is referred to as as quilt medium. The duvet medium may also be picture, video or an audio file. Hiding the key knowledge within the quilt medium results in stego. The stego file includes both quilt medium and secret information. The steganography takes expertise over cryptography. In cryptography by looking at the data itself hacker is aware of that it has been encrypted, so by means of doing a little cryptanalysis he can effortlessly get the key information, however in steganography, the hacker couldn't identify that, a secret information has been embedded as it allows for invisible communication.

Steganography is the science of writing hidden data inside the duvet media in this sort of approach that no person besides the sender and the supposed recipient can realise that there is a hidden information. This field has been brought into the general public attention with the aid of intelligence corporations and information media. Now a day's cryptography is combined with steganography to provide brought protection. Whilst cryptography presents privacy, Steganography is intended to provide secrecy. Steganography works to mask the very existence of the message. Applications of Steganography fluctuate from navy, industrial applications to copyright and intellectual Property Rights (IPR). The messages will also be despatched and bought securely with the aid of utilizing lossless steganography approaches [2]. Frequently, steganography was situated on hiding secret understanding within the snapshot files. But present day work suggests that there has been developing interest among research fraternity in making use of steganographic tactics to video files as good [3], [4].

The abilities of utilizing video documents in hiding understanding is the introduced safety in opposition to the assault of hacker as a result of the relative complexity of the constitution of video compared to picture records. The potential within the process is that the amount of knowledge (payload) that may be embedded is extra in LSB tactics. However many of the LSB strategies are inclined to attacks are described in [5], [6]. The audio steganography additionally performs a excellent function in delivering protection. This makes research fraternity inquisitive about designing new approaches. The outcome obtained are determined to be giant and inspiring.

2. PROPOSED SYSTEM

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

<https://doi.org/10.62647/ijitce.2025.v13.i2.pp93-97>

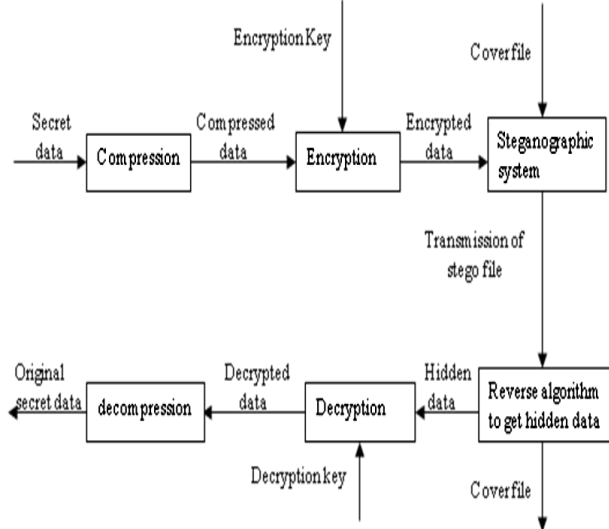


Fig 1: Overview of the proposed system

The cover file is broken down into multiple frames and then the proposed LSB based technique is applied 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 Information compression and decompression. The LZW (It has been due to the fact identify on the title of the authors: Jacob Ziv, Abraham Lempel and Terry Welch) a dictionary-headquartered compression algorithm that maintains an express dictionary [7]. The code phrases output with the aid of the algorithm consist of two factors: an index regarding the longest matching dictionary entry and the first non-matching symbol. Furthermore to outputting the codeword for storage or transmission, the algorithm additionally adds the index and image pair to the dictionary. When an emblem that not but in the dictionary is encountered, the codeword has the index worth 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 customary text file from the compressed file. It takes the movement of codes output from the compression algorithm, and uses them to precisely recreate the input. The LZW algorithm is efficient for the reason that it does now not ought to cross the dictionary to the decompression code. The dictionary can also be constructed exactly because it used to be in the course of compression by using using the enter circulate. This is possible on the grounds that the compression algorithm invariably outputs the string and persona components of a code earlier than it makes use of it in the output circulate.

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 approach of changing the plain text (normal text) into cipher text (unreadable structure) with the aid of a key. The encryption manner used in the proposed system is AES (advanced Encryption regular) [8]. It's uneven-key algorithm, that means that the identical key's used for both encrypting and decrypting the info.

AES is established on a design precept referred to as a substitution-permutation network, and is speedy 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 identify of the normal is modified to AES-128; AES-192 or AES-256 respectively. AES operates on a four×four column-important order matrix of bytes, termed the state. The key measurement used for an

<https://doi.org/10.62647/ijitce.2025.v13.i2.pp93-97>

AES cipher specifies the quantity of repetitions of transformation rounds that convert the plaintext into cipher textual content.

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.

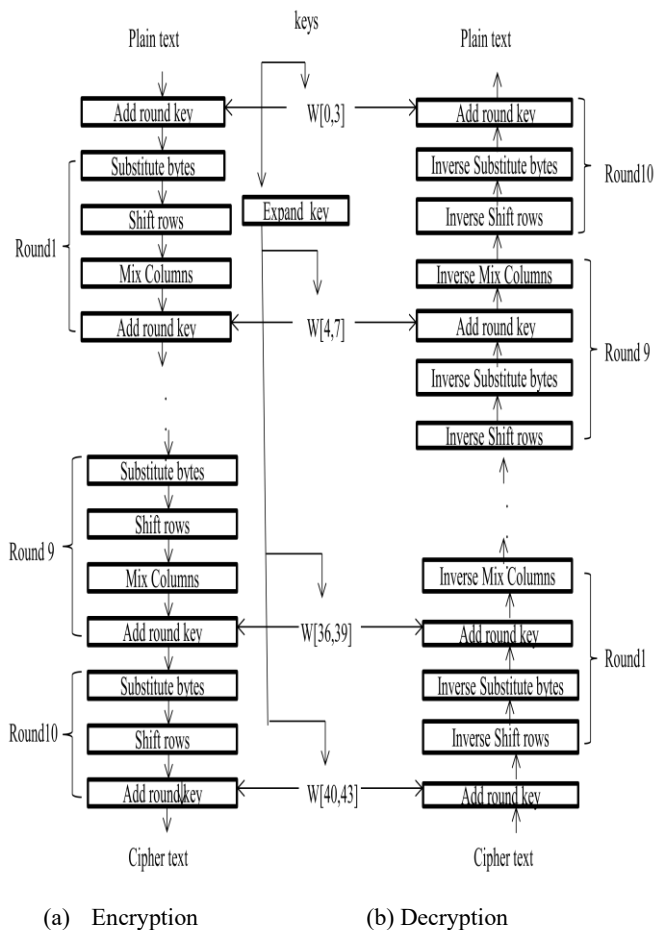


Fig 2: Overall structure of AES algorithm

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: InverseSubstitutebytes—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: InverseSubstitutebytes

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

<https://doi.org/10.62647/ijitce.2025.v13.i2.pp93-97>

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.

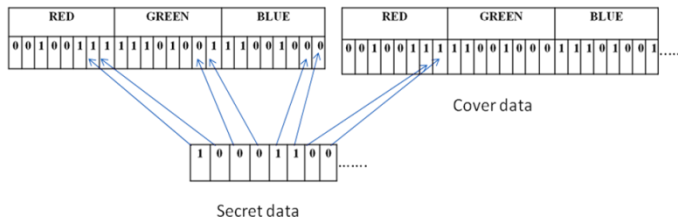


Fig 3: Embedding process

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

Byte1 Byte2 Byte3 Byte4
00100110 11101000 11001011 00100100

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

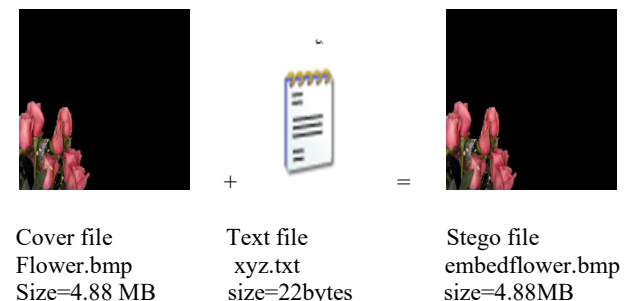
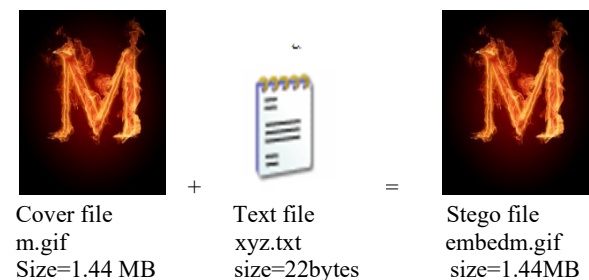
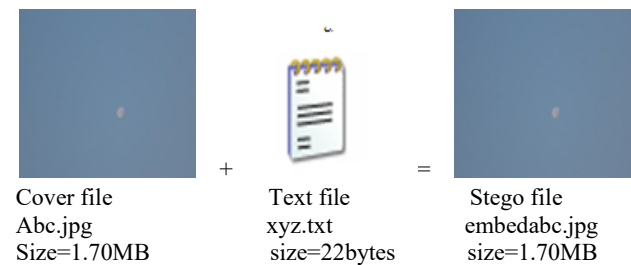
When the system is executed GUI (Graphical User Interface) is displayed. The snapshot of the main window is as shown in figure 4. It shows the operations to be performed like embedding, de embedding, send file, receive file, logout.

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.

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.

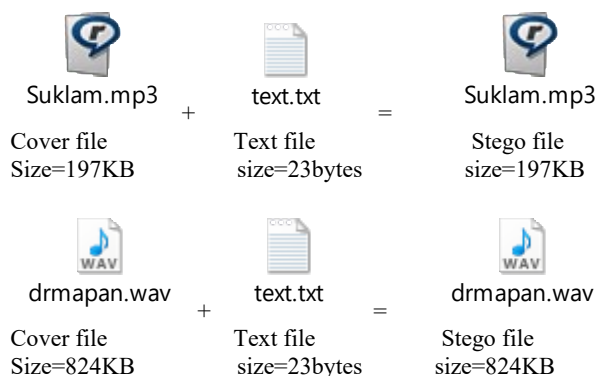
The proposed method is applied on different cover files and the results are given below:

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

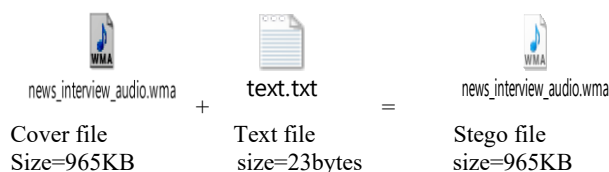
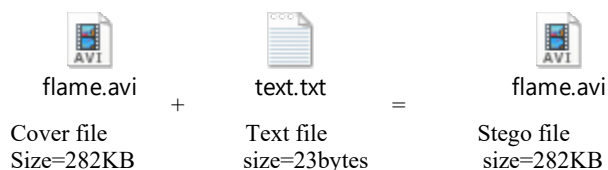
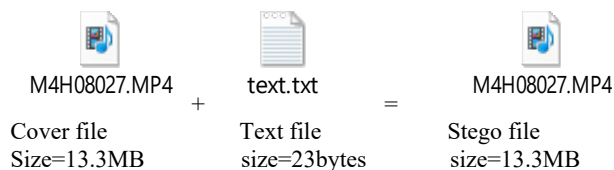


Case 2: Cover medium=audio (mp3,wav)
Secret data=text file

<https://doi.org/10.62647/ijitce.2025.v13.i2.pp93-97>



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



4. CONCLUSION

The proposed Procedure is utilized on exceptional file codecs of snapshot, audio and video files. In the proposed technique secret message is first compressed, encrypted after which embed in cover file with the support of steganographic system. It could increase the confidentiality of expertise. Efficiency evaluation of the proposed method after assessment with improved LSB method is fairly encouraging. It may be further increased 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.