Image Processing Using Watermarking (Text, Image, Audio, Video)

[1]Prof. Vishal Shinde, [2]Yogesh Ojha, [3]Mitesh Bhanushali, [4]Vaibahv More

[1]Asst. Professor, [2,3,4]BE Student, [1,2,3,4]Comp. Engg. Dept, SSJCET, Asangaon, India.

[1]mailme.vishalshinde@gmail.com, [2]ojhayogesh99@gmail.com, [3]mitbhanushali90@gmail.com, [4]vaibhavmore.20feb@gmail.com

Abstract — **Large demand of various applications requires data to be transmitted in such a manner, that it should remain secure. Data transmission in public communication system is not secure because of interception and Interruption by hackers. The solution for this problem is Steganography, which is the art and science of hiding messages in media files like images, audios, and videos in such a way that no one, apart from the sender and receiver, see the message. Today's data hiding systems uses multimedia files like Texts, Images, Audios and Videos. Embedding secret messages in text and images might prove to be easy but is usually a more difficult in Audio and Video files. Varieties of techniques for embedding information in Texts, Images, Audios and Videos have been established. This paper will present the general techniques of hiding secret information or data using all these technology (Texts, Images, Audios and Videos) in single application using both Cryptography and Steganography techniques.**

Index Terms: Audio steganography, Cryptography, Encryption, Image steganography, LSB, Sequential coding, TripleDES, Text cryptography, Video steganography.

I. INTRODUCTION

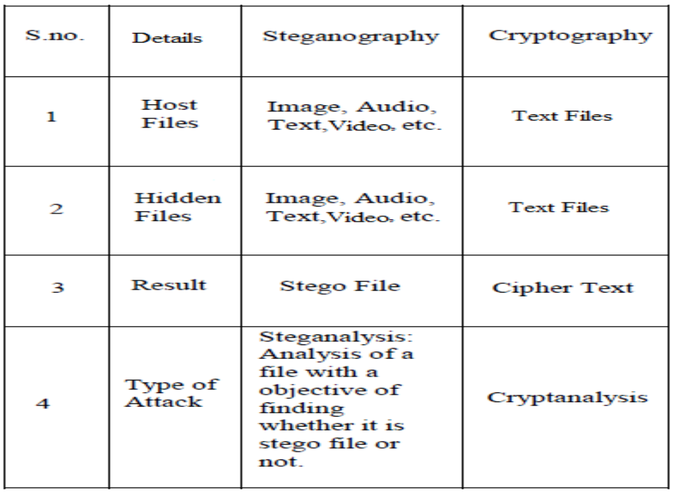
The hackers usually try to hack the important documents of the people or organizations for their benefits. What they do is, they take money from other person or may be from another organization, and hacks the important files or documents of another organizations. Due to such hacker available around the network it is important to have some kind of protection to our data or it is possible to make our data not attractive for the hackers.

1.1 STEGANOGRAPHY

It is the science and art of hiding data's in some cover files like images, audios, videos. It provides a way to hide the user's data in some media files and send it over a network. This makes the data non attractive.

1.2 CRYPTOGRAPHY

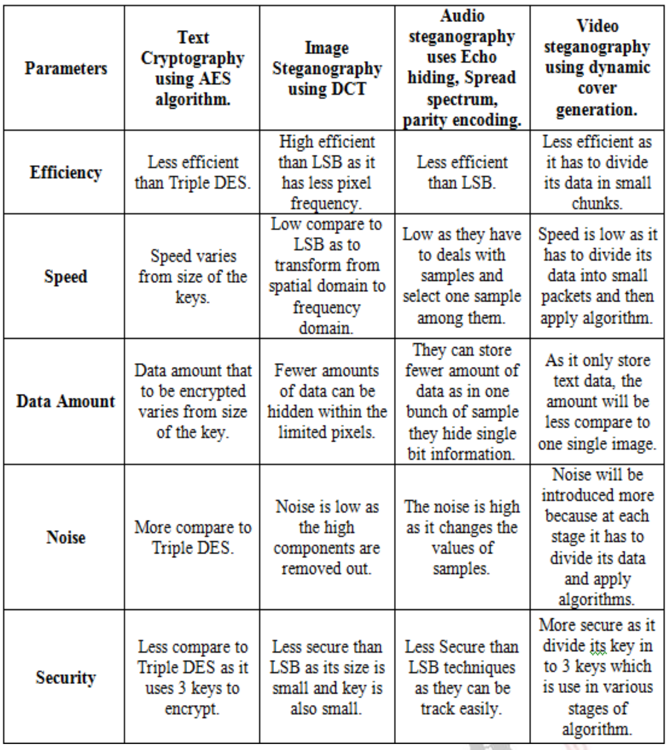
Cryptography encodes data in such a way that nobody can read and understand it. For cryptography key and text files are used to encode the data, which uses the message digest to make it unreadable for anyone. There is some difference in cryptography and steganography technique, in cryptography the hidden message is visible for anyone who sees it, because information is in text form.



S.no.	Details	Steganography	Cryptography
1	Host Files	Image, Audio, Text,Video, etc.	Text Files
2	Hidden Files	Image, Audio, Text,Video, etc.	Text Files
3	Result	Stego File	Cipher Text
4	Type of Attack	Steganalysis: Analysis of a file with a objective of finding whether it is stego file or not.	Cryptanalysis

Table 1.Comparison between steganography and cryptography [6]

II. LITERATURE SURVEY



Parameters	Text Cryptography using AES algorithm.	Image Steganography using DCT	Audio steganography uses Echo hiding, Spread spectrum, parity encoding.	Video steganography using dynamic cover generation.
Efficiency	Less efficient than Triple DES.	High efficient than LSB as it has less pixel frequency.	Less efficient than LSB.	Less efficient as it has to divide its data in small chunks.
Speed	Speed varies from size of the keys.	Low compare to LSB as to transform from spatial domain to frequency domain.	Low as they have to deals with samples and select one sample among them.	Speed is low as it has to divide its data into small packets and then apply algorithm.
Data Amount	Data amount that to be encrypted varies from size of the key.	Fewer amounts of data can be hidden within the limited pixels.	They can store fewer amount of data as in one bunch of sample they hide single bit information.	As it only store text data, the amount will be less compare to one single image.
Noise	More compare to Triple DES.	Noise is low as the high components are removed out.	The noise is high as it changes the values of samples.	Noise will be introduced more because at each stage it has to divide its data and apply algorithms.
Security	Less compare to Triple DES as it uses 3 keys to encrypt.	Less secure than LSB as its size is small and key is also small.	Less Secure than LSB techniques as they can be track easily.	More secure as it divide its key in to 3 keys which is use in various stages of algorithm.

Table 2. Comparison of various Techniques

2.1 Text

Text cryptography is one of the easiest way to hide information in such a way the format get change and unreadable for the user. But one of the problems with it is that the changed format is visible for other who sees it. Also the capacity of data to be hidden is limited. There are various cryptography techniques are available like DES, AES, IDEA, BLOWFISH, etc. which provide some sort of security to the data in terms of the key matters. But still it proves to be insecure. If hacker sees this data then can try brute force attack on it to get the hidden data.

2.2 Image

Images are used to hide data inside it. Images can hide a lot more information compare to text cryptography. Image use steganography techniques to hide data inside the pixels of the images. Image steganography involve various techniques based on their spatial information, their frequency information and some time using hybrid information. Example of Spatial image steganography process is Checksum, basic m-sequence, etc.

LSB steganography is one of the popular techniques to hide data in last bit of each pixel of image with text data. But text data doesn't show much information compare to images, thus it fail to prove better than hiding image inside another image.

2.3 VIDEO

2.3.1 A Novel Video Steganography Technique Using Dynamic Cover Generation

In this technique there is need for data file, key, frames and algorithm. First the keys will be of 32 bits and are divided into 3 parts i.e. 3 keys. 1 part is of 16 bit that is 1 key is of 16 bit length and remaining 2 parts is of 8 bit each that is 2 keys will have 8 bit each. First key will be used at frames division time to divide frames in to two different sets i.e. 2 and 4 sets. And the other 2 will be used at encryption time. This technique divides the chunks of data in two parts, and that two parts will going to divide in other 2 parts till the data is too small to encode and saved. For example if the data file is of 64 bit, than each will be divided into 32-32 bits and these two parts will divide the data into 16-16 bits and so on. Then this small chunk of data is encoded using 2nd and 3rd key and image and algorithm on each side and stores it.

As we know the image carries a lot and lot of information compare to text file [4].

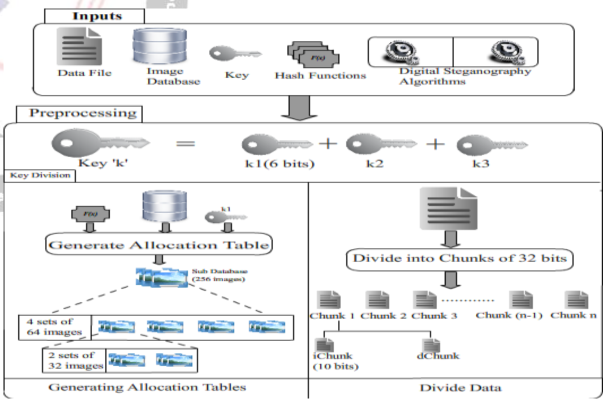


Fig. 2.3.1 Inputs and Preprocessing [4]

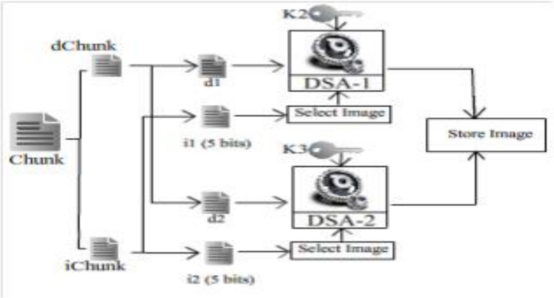


Fig. 2.3.2 Implementation [4]

2.3.2 A Steganography Approach For Sequential Data Encoding And Decoding In Video Images:-

This is another technique which hides text data in to frames of video files. This technique show hiding a bunch of data using a particular pattern from one frames to another frames. As this also use LSB technique the Peak Signal To

Noise ratio will be good compare to normal image hiding. But still it will require a lot of text data to hide a huge amount of data. This technique is fast but only for text inputs. This technique uses one key, text file, and one/ more frames of video files. If the text file is more than 1 or then it might use another frame of the video. Thus for a huge amount of data the process will need a good key combinations [1].

2.4 AUDIO

2.4.1 Parity Coding

Parity coding is one of the audio steganographic techniques. It takes a bunch of sample instead of individual sample. In this bunch of sample take any one sample for encoding. If the message bit is 0, and parity bit is 1, then replace the LSB value of one of the sample with 0. Similarly if the message bit is 1 and parity bit is 1, then in one of the sample of one region replace LSB of that sample with 1. Also, if the message bit is 0 and parity bit is 0, then replace LSB of one of the sample with 0.Thus for all such sample the encoding is to be done. This will take a lot of time if the message is large and also require an audio file of larger length [2]. The Figure below shows the working of parity coding

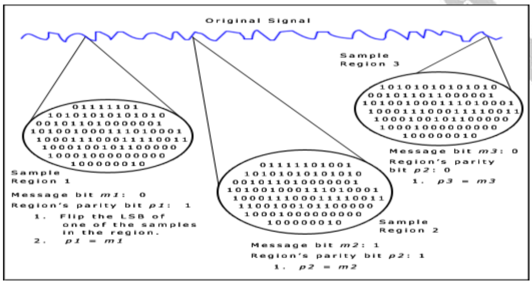


Fig. 2.4.1 Parity coding [2]

2.4.2 Phase Coding

In This technique the messages are replaced in the phases of the signal. In this if the message bit is 0 than the phase change by 90°. Similarly, if the message bit is 1 then the phase changes to -90°. Thus all the message is encoded in this way.

PHASE_CHANGE = +90° if message bit=0
-90° if message bit=1

But this will take lot of time to encrypt all the message bits. Also the signal changes drastically that it can be detected easily as the signal will contain lot of noise.

The below figure shows Phase coding technique

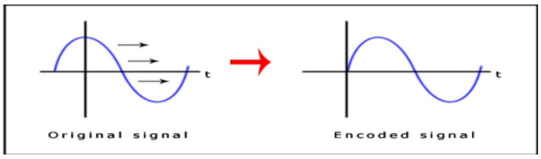


Fig. 2.4.2 Phase coding [2]

2.4.3 Echo Hiding

Echo hiding technique hide data in a wav file by introducing an echo into the discontinuous signal. Echo hiding has advantages that, it provides a high speed and data quantity when compared to other methods. Only one bit of data could be encoded if only one echo is produced from the cover signal. Hence, before the encoding process starts the cover signal is broken down into small blocks. Once the encoding process is done, the blocks are attached again together to create the final signal [2].

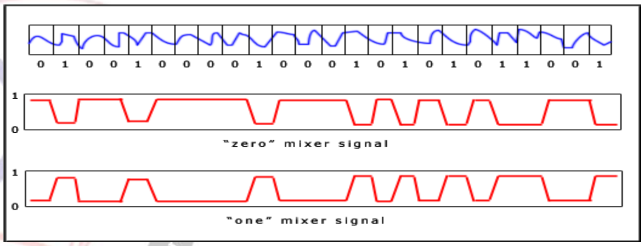


Fig. 2.4.3 Echo hiding [2]

III. PROPOSED SYSTEM

3.1 ARCHITECTURE

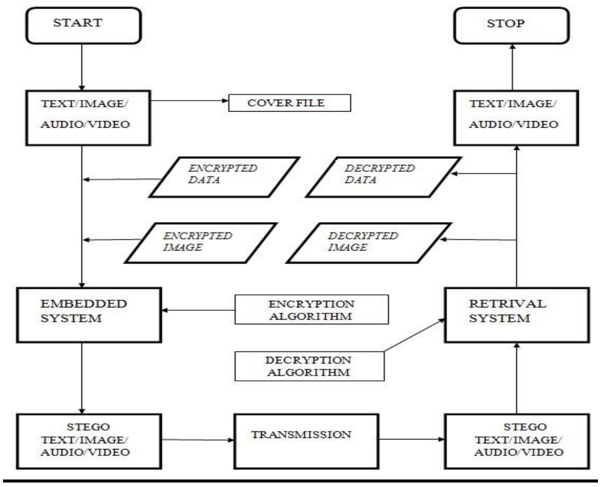


Fig. 3.1 Proposed Architecture

3.2 AUDIO STEGANOGRAPHY

3.2.1 LSB Coding

LSB encoding is very popular technique to hide data in audio signals. In this technique the last bit of each sample value in cover audio file. If the message bit a, then see its ASCII value (ASCII value of A is 65) and computing its binary value as **01000001** then by changing the last bit of the samples with this values as below:

1100101**0**
1000010**1**
0111100**0**
1110010**0**
1010110**0**
1011001**0**
1111010**0**
1011101**1**

Thus by encoding these bits in LSB, there will not be any changes in the original signal. The speed of encoding the values is also high [2].

3.2.2 LSB Algorithm for Audio

Encoding:

1. Input the text message that is to be embedded.
2. Read the audio file and text message in binary.
3. The converted binary audio file is sampled into 8 bit equal size samples.
4. Store audio file in a matrix of a[C][8], where m depends upon size of audio file.
5. Select data reside on Fibonacci index [j] of matrix of audio file.
6. Taking this data of Fibonacci index, select the index value where

   a [j]%2 == 0

   Where j is Fibonacci index of matrix
7. Embed the data in the reverse order in LSB at the selected position.
8. Repeat the step 6 to 8 until the complete text is encoded.

Decoding:

1. Read the audio file in binary form.
2. Select the value at Fibonacci index[j].
3. Retrieve the index value of a[j]%2 == 0
4. Decode the encoded data from the LSB technique
5. Store the binary value of Least Significant Bits.
6. Convert the binary values to decimal to get the ASCII values of secret message.
7. From the ASCII value read the secret message

3.3 VIDEO STEGANOGRAPHY

For video steganography, multiple methods is going to be used. The following are the method which is going to be used:-

3.3.1 Symmetric Encryption

In symmetric encryption the message bit and key file is XORed with each other to produce the encryption value. The same key is used at the sender and receiver side, that's why it is called as symmetric key. At the receiver side the encryption value and the key is then XORed, to produce the message bit. The example is shown below:

Encoding

Message	10101000
Encryption key	00001111
Encryption Value	10100111

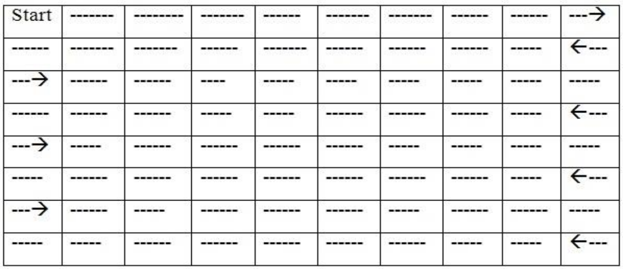
Decoding

Encrypted Data	10100111
Encryption key	00001111
Recovered Message	10101000

3.3.2 Sequential Encoding

In sequential encoding the coding starts at one position and follow a certain sequence of encoding the data. The message being encoded is place in the last bit of the pixels and this encryption follows a specific pattern of encoding till the end of the message bit. The following is the process of hiding data in the frames of the video file [3].

a. Message data are encoded-decoded from some starting point.
b. Message data is then encoded-decoded in unvarying pattern.



Start	------	-------	------	-----	------	------	-----	-----	---→
-----	------	------	-----	------	-----	-----	-----	----	←---
---→	-----	-----	----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----	-----	-----	←---
---→	-----	-----	-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----	-----	-----	←---
---→	-----	-----	-----	-----	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----	-----	-----	-----	←---

Fig. 3.3.2 Sequential Encryption

3.3.3 LSB Techniques

In normal LSB technique the data is hidden in the last bit of the images, but a color image consist of 3 colors i.e. RGB (Red, Green, and Blue). These three colors are of 8 bit each thus can be used to hide 1 bit in each of this colors last bit. Due to this the data can be hidden is also high compared to other techniques. Also this will not affect the quality of images of the video file. The following shows this technique in detail:
For example:

Red Green Blue
Pixel value (11010101, 11110101, 01010011)
Encoded value (1101010**0**, 1111010**1**, 0101001**0**)
Message 010

This algorithm has been referred from [3].

3.3.4 *ALGORITHM*

A) *Encoding Algorithm:-*

C:-Carrier Video Stream
C_f:-Set of Frame of Carrier Video Stream
H:-Hidden Image.
H_f:-Set of Images.
HT:-Stego Video Stream
HT_f:-Set of Frame of Stego Video Stream
HT:-Reconstructed Video Stream
HT_f:-Set of Frames of Reconstructed Stream
K: - Key

Input: - Carrier Video C (W, H, N_c),Hidden Image H(W, H, N_h),Key.
Output: - Stego Video HT (W, H, N)
Algorithm: - Stego Video (C, H, K)

Where C and H are the carrier Video and hidden Image with height H and Width W, Number of Frames N_c and N_h attributes and Symmetric encryption key.

```
{
Extract Frames of carrier video $C_f$ and Secret Image $H_f$ respectively.
$C_f$= Extract_Frames(C)
$H_f$ = Images (H)
1.      N = $N_c$
2.      For k=1 to N  //Work for each frames of videos C and H.
{
i.      Read Images $H_f$[k] //One frame from each set
         Message=$H_f$(k)
ii.     Represent Images $H_f$(k) in integer.
iii.    Prepare header for size of Images and add it to beginning of the frame.
         Header = size (Message)
        //In integer total 8 bits 4-4 to each Height and Width.
        New_Message = Add(Header, Message)
iv.     For each Used Images, Perform Encryption with Key k
        Encrypt_Message = XOR (New_Message, k)
v.      Read Frame $C_f$(k)
        Cover = $C_f$(k)
vi.     Embed secret frame under Cover frame's LSB using sequential encoding with RGBBGRRGB with predefined pattern.
        //R.G.B means each bit stored in Red, Green, Blue band of next pixel.
        Stego_Message = Embed (Cover, Encrypt_Message)
vii.    Generate the Stego Frame
        $HT_f$(k) = Stego_Message
}
1.      Add the Stego Frames set $HT_k$ to form video with proper frames rate and compression.
}
```

B) Decoding Algorithm

Input: - Stego_VideoHT(W, H, N), Key.
Output: - Hidden/Secret Images HT (W, H, N_h)
Algorithm: -Extract_Video (HT, K)

```
        //Where HT is the Hidden Images with Height H and Width W, Number of Images N attributes
{
1.      Extract frames set $HT_f$ from Stego video.
        $HT_f$= Extract frames (HT)
2.      Nh= N
3.      For k=1 to N //Work for frame set of video HT
{
i.      Read frame $HT_f$[k]  //One frame for each set $HT_f$.
            Stego_Message = $HT_f$(k)
ii.     Extract Header from frames which was added at the time of encoding from LSB of stego frame following sequential decoding with RGBBGRRGB predefined pattern
        Header = LSB (Stego_Message) //Only LSB //of 8 bytes of Steg0_Message
        Secret_Message = LSB (Stego_Message) //LSB of whole Stego_Message except First 8 bytes.
iii.    Perform Decryption of Header and Secret_Message with Key k
            Header = XOR (Header, k)
            Secret_Message = XOR (Secret_Message, k)
iv.     Reconstruct the Hidden Images from Secret_Message.
            HT(k) = Reconstruct (Secret_Message)
        }
        }
```

3.3.5 *MATHMATICAL MODEL*

Validity and Imperceptibility are the important parameter. Validity lies with the similarity between recovered secret frames and the original secret frames. Imperceptibility depends on the similarity of carrier frame and original frame. The calculation of Peak Signal to Noise Ratio

(PSNR) and also Root Mean Square Error (RMSE) values. The root mean square value should be smaller and peak signal to noise ratio value should be higher to show more similarity between two images.

$$PSNR = \frac{10\log_2 * H * W * 255 * 255}{AV}$$

Where $AV = \sum_{x=1}^{M}\sum_{y=1}^{N} |G(x,y),F(x,y)|^2$

$$RMSE = \sqrt{\frac{AV}{(W-H)}}$$

Where F(x, y) is Original Carrier image video frame. G(x, y) is embedded carrier image video fame. W and H represent width and height of image.

IV. EXPECTED RESULT

4.1 Texts

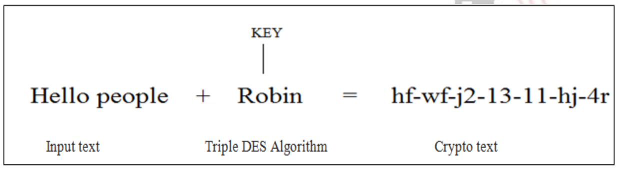


Fig. 4.1.1 Text Encryption

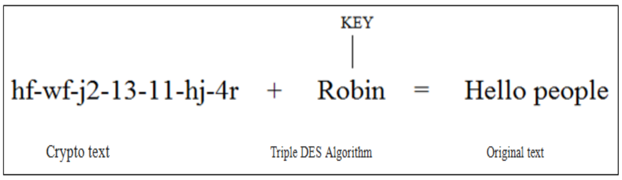


Fig. 4.1.2 Text Decryption

4.2 Image

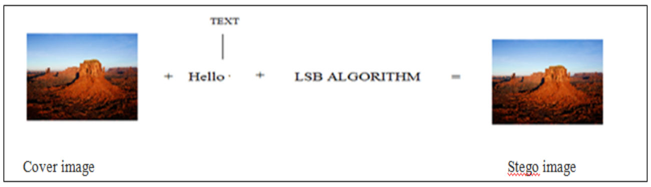


Fig. 4.2.1 Image Encryption

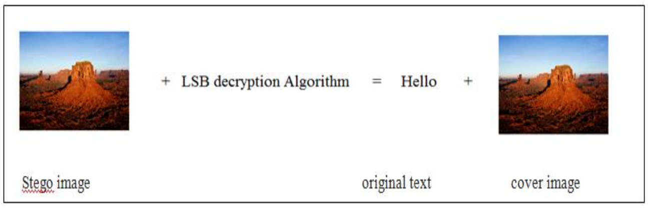


Fig. 4.2.2 Image Decryption

4.3 Audio

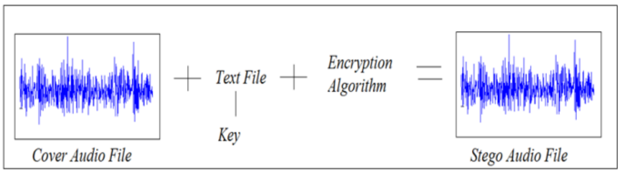


Fig. 4.3.1 Audio Encryption

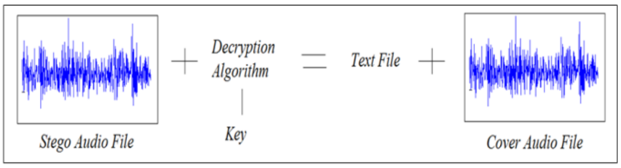


Fig. 4.3.2 Audio Decryption

4.4 VIDEO

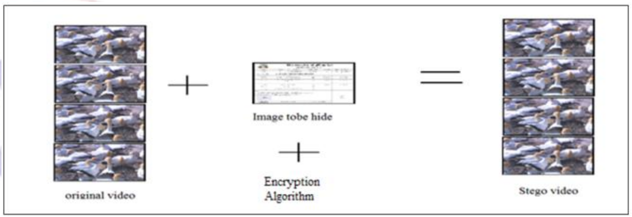


Fig. 4.4.1 Video Encryption

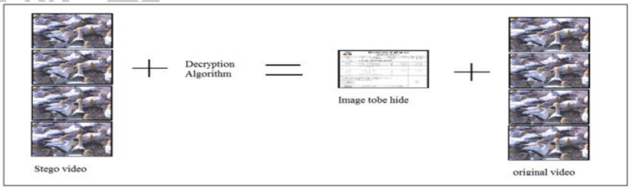


Fig. 4.4.2 Videos Decryption

V. CONCLUSION

Though there are several methods to provide security to data's, this paper shows a mixture of various techniques to hide data in single application. This paper has text to text cryptography which can be used to change the format of data and very useful to transfer keys. Another technique is use to hide data inside an image and also image inside image. This paper shows audio steganography in which data is used to hide in wave files. Video is the set of images which is arranged in very short time period to make it as a continuous picture. This paper shows the hiding of image inside this video frames. Usually in image to image steganography, it is hardly possible to get the original image as it is. But using video frames the whole image can be encoded and decoded as it is. This paper proposed various algorithms like Triple DES Algorithm for text to text Cryptography, LSB for the Image and Audio steganography and LSB, Symmetric and Sequential encoding of image in Video steganography.

REFERENCES

[1] Mritha Ramalingam and Nor Ashidi Mat Isa, *“A steganography approach for sequential data Encoding and decoding in video images”,* 2014 International Conference on Computer, Control, Informatics and Its Applications.

[2] Jayaram , Ranganatha H , and Anupama H**,***“Information Hiding Using Audio Steganography – A Survey”*, The International Journal of Multimedia & Its Applications (IJMA) Vol.3, No.3, August 2011.

[3] Pooja Yadav, Nishchol Mishra, and Sanjeev Sharma*,“ A Secure Video Steganography with Encryption Based on LSB Technique ”,* 2013 IEEE International Conference on Computational Intelligence and Computing Research.

[4] Vivek Sampat, Kapil Dave, Jigar Madia, and Parag Toprani*,“ A Novel Video Steganography Technique using Dynamic Cover Generation”*, National Conference on Advancement of Technologies – Information Systems & Computer Networks (ISCON – 2012) Proceedings published in International Journal of Computer Applications® (IJCA).

[5] Manoranjan Kr Sinha ,Dr. Rajesh Rai, Prof. G. Kumar, “*Literature Survey on Digital Watermarking,”* (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (5) , 2014, 6538-6542.

[6] Vijay Kumar Sharma, Vishal Shrivastava, “*A Steganography Algorithm For Hiding Image In Image By Improved LSB Substitution By Minimize Detection”,* Journal of Theoretical and Applied Information Technology 15th February 2012. Vol. 36 No.1.

[7] Abdelfatah A. Tamimi, Ayman M. Abdalla, Omaima Al-Allaf, *“Hiding an Image inside another Image using Variable-Rate Steganography”,* (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No. 10, 2013