 ISSN: 2321-8169
3970 - 3975

Steganography on Multi-media Elements

Vipin Deoli, Vassudev Dhempo
A.S.M's Institute of Management & Computer Studies
Plot C-4, Wagle Industrial Estate, Near Mulund Check Naka,
Opp. to Aplab, Thane (W) – 400604 India.
anidhempo2304@hgmail.com
A.S.M's Institute of Management & Computer Studies
Plot C-4, Wagle Industrial Estate, Near Mulund Check Naka,
Opp. to Aplab, Thane (W) – 400604 India.
2010vipindeoli@gmail.com

***Abstract*—**Stegnography is method to make the communication of secret data hiding through a media. A perfect Steganographic in media enables to embedding data in an media, robust and secure way and then extracting it by authorized people. In this paper we introduces a new approach for Least Significant Bit (LSB) based on image and audio steganography that increases the existing LSB substitution techniques to improve the security level of hidden information by introducing secret key technique. The new security method hides secret information within the LSB using secret key combined with LSB to protect data from unauthorized users. In original LSB methods, secret data is present to a specific position of selected media. Because of this, knowing the extracting methods, anyone can extract the hidden information. In our method, we use secret key which is embedded with LSB and the according to selected media the data will be hidden therefore, it is difficult to extract the hidden information knowing the retrieval methods.

Keywords: *Stegnography, Key, LSB algorithm, Media*

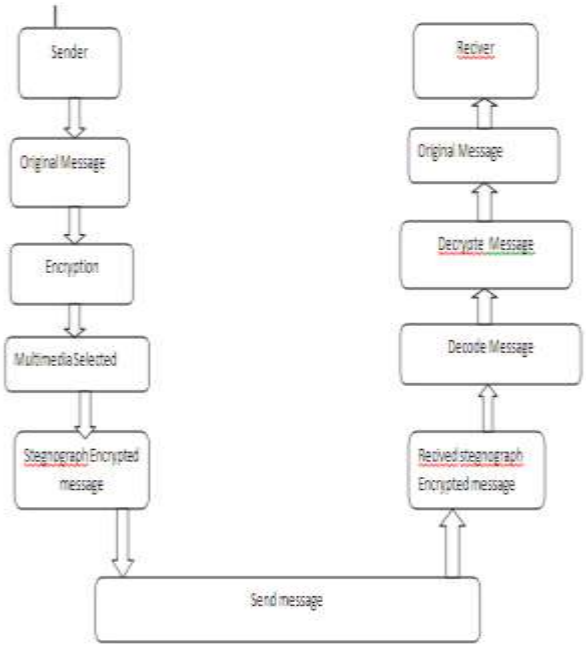
*****

I. INTRODUCTION

The drastic evolution in connectivity the communication is moving to another dimension therefore we require exceptional way of security on computer networks [3]. The importance of network security is increasing as there is increase in data exchange on internet.[3] Maintaining data integrity and confidentiality is necessary to protect against unauthenticated users.For avoiding secret message extracted during transmission over network there are two methods encryption and stegnography.Encryption is a method of encoding the message using a secret key shared between authorized user, The user having valid key will only decode the original message. Another method is stegnography,the message is wrapped into a media so that only authenticated person will retrieve the data from media. To make a steganographic communication more safer to communicate the secret data will be encrypted before being hidden in the media .In such way Cryptography and Steganography can be merged together to increase confidentiality.

Steganography algorithms can be categorized bin to many defining properties. The importance of these are Transparency, Capacity and Robustness. Level of extracting data we included key as authentication clue for accessibility of authorized data [10]. The provided key will be embedded in LSB algorithm of selected media and the hide the secret message to be transferred .This will increase the level of confidentiality though the method of extracting data from LSB is known to attacker but the variation made in algorithm using key will not lead to extract the secret message

II. ARCHITECHTURE DIAGRAM



III. ALGORITHM

Basics of LSB:

Before understanding LSB we should how to work on bytes. Byte arithmetic is different than normal arithmetic.

Bytes: individually as integers and as arrays

Bit Operations: Logical AND (&), OR(|) and how they work

Images: BufferedImage specifically

 ISSN: 2321-8169
3970 - 3975

ImageIO: how image files are opened and saved
Graphics2D: accessing user space image properties
Raster: specifically WritableRaster allows access to the buffer
DataBufferByte: Buffer used with BufferedImage
Bytes:
Bytes are the elementary data source of most applications, and many programmers will [i]never[/i] use them in any source code, but that is beside the point. A byte is made of bits, 1s and 0s, 8 of them to be exact. And the 8 0s and 1s have a decimal value, it is simply a case of transforming the binary (base 2) into decimal (base 10).
Value by position: 128 64 32 16 8 4 2 1 (and all positions with a 1 are added together)
Examples:
00000000 = 0
00000010 = 2
00000111 = 7
00001011 = 11
And so on…

A byte can be transformed from an int in java by simple casting:
[il]Byte b = (byte)7;[/il]
Most classes in java have a method for returning the byte[] of an object, either as a section of the object or the entire object.
String Class Example:

String w = "William";
Byte[] b = w.getBytes();

Where b[0] will now contain the ascii value for 'W' 87 if printed. Though it is good to remember that although it appears as an int, when displayed, it is in fact a byte, which is stored as 8 bits, in this case: 01010111.

Bit Operations:
There are simple operations which most computer users have either heard of, or even used:
AND:
The AND(&) bit operator, will AND 2 bytes together. The same rules apply as when using true and false values, where 1 = true, and 0 = false. If both bytes have a 1 in the same position, then the result for that position is a 1, otherwise the result is a 0.

Example:
01010111 = 87
01100101 = 101
01000101 = 69

[il]Byte b = 87 & 101; //69: 01000101[/il]

OR:
The OR(|) bit operator, will OR 2 bytes together. The same rules as with AND where 1 = true, and 0 = false, only when using OR, as long as one of the bits in the position is a 1, then the result is a 1. Only if both bits are 0, is the result a 0.

Example:
01010111 = 87
01100101 = 101
01110111 = 119

[il]Byte b = 87 | 101; //119: 01110111[/il]

On top of these basic operations, we can also shift bits:
Left Shift:
An important thing to remember when left shifting bits, is if the first bit is not a 1, a single left shift will essentially double the value. What actually happens, is a 0 is added on the right hand side of the bits, then the far left bit is removed thus leaving a new set of 8 bits. Also, when shifting in Java, a number of positions to shift must also be supplied. If the value is greater than 1, the process is simply repeated that many times each time beginning with the result of the previous shift. Thus any value will become 0 if shifted 8 times.

Examples:
(single shift)
01010111 = 87
<< 1
10101110 = 174

(double shift)
01010111 = 87
<< 2
01011100 = 92

Byte b1 = 87 << 1; //174: 10101110
Byte b2 = 87 << 2; //92: 01011100

Right Shift:
A right shift is the opposite of a left shift in the sense that a 0 is added to the left side of the bits, and the far right bit is removed, once again leaving a set of 8 bits.

Examples:
(single shift)
01010111 = 87
>>> 1
00101011 = 43

(double shift)

 ISSN: 2321-8169
3970 - 3975

01010111 = 87
>>> 2
00010101 = 21

byte b1 = 87 >>> 1; //43: 00101011
byte b2 = 87 >>> 2; //21: 00010101

These are the bit and byte operations which are used to effectively create this steganography application, I will provide some more complex examples, breaking down the steps of adding the data to the image, a little later.

BufferedImage:
A bufferedImage is something to be comfortable with when dealing with images. They are easily used with the newly introduced ImageIO class of Java 1.5.0 as well as containing methods for accessing the raster and buffer of the image, which makes image editing much easier. The basic actions for creating a new image are:

BufferedImage img = new BufferedImage(int, int, int);

File file = new File(String);
BufferedImage img = ImageIO.read(file);

ImageIO:
A useful class to handle IO operations on images. This class has much to offer, but as far as this program is concerned, the read() and write() methods will be sufficient.

Graphics2D:
A class which has been around for a long time as far as Java is concerned, and allows access to some of the more in depth aspects of graphics/images. Allows for creating editable areas in a new image or an image which already exists. As well as allowing a way to reach the renderable area of the image. This class also allows for an easy switch from image space to user space, which is necessary when modifying or reading certain bytes of an image.

WritableRaster:
This by definition is the process of rendering an image pixel by pixel, which comes in handy when you need to access the bytes of an image, that are representing pixels. WritableRaster is a sub-class of Raster itself, which has methods to access the buffer of an image more directly.

DataBufferByte:
The form of a byte[] buffer for an image.

Existing LSB ALgorithm:

Least significant bit (LSB) insertion is the most widely efficient.[3]The name of the method implies that; the least significant bits of the cover-image are manipulated so that they form the embedded information. The following example shows how the letter A can be hidden in the first eight bytes of three pixels in a 24-bit image.
Pixels: (00100111 11101001 11001000) (00100111 11001000 11101001) (11001000 00100111 11101001)
A: 10000001
Result: (00100111 11101000 11001000) (00100110 11001000 11101000) (11001000 00100111 11101001)
The three underlined bits are the only three bits that were actually changed. LSB insertion requires on average that only few bits in an image be changed. Since the 8-bit letter A only requires eight bytes to hide it in, the ninth byte of the three pixels can be used to hide the next character of the hiddenmessage.
A slight modification of this technique allows for embedding the message in two or more of the least significant bits per byte. This increases the capacity of hidden information on cover-object, but the cover-object degrades more statistically, and it is more noticeable. Other changes on this technique include assuring that statistical changes in the image do not occur. Some intelligent software also checks for areas that are made up of one solid color. Modifications in these pixels are then avoided because slight changes would cause noticeable variations in the area.

Advantages and Diadvantages of LSB Insertion:
Major advantage of the LSB algorithm is it is quick and easy[4]. There has also been steganography software developed which work around LSB color alterations via palettemanipulation[4]
LSB insertion also works well with gray-scale images. A slight variation of this technique allows for embedding the message in two or more of the least significant bits per byte. This increases the hidden information capacity[4].
But disadvantage is it is a very common technique. If a hacker knows LSB technique ,he can easily hack and get information.

Proposed Technique:
The Proposed Technique in our paper involves a key concept.The key is converted into a binary form.i.e.1's and 0's.If while embedding the teext message into media file 1 is encountered of the key embedd or else if 0 is encountered jump and next byte is encoded with the message.

Algorithm of Proposed Technique
Step 1: Covert the Media,message and key into binary,
Step 2:Read the binary of key.

 ISSN: 2321-8169
3970 - 3975

Step 3:If 1 is encountered embed message into medai file.
Step 4:Else 0 is encountered do not embed but jump

Audio LSB algorithm:
Least significant bit (LSB) coding is the most acceptable way to embed information in a digital audio file[1]. By replacing the least significant bit of each sampling point with a binary message, LSB coding allows for a huge amount of data to be encoded. In LSB coding, the ideal data transmission rate is 1 kbps per 1 kHz. In some implementations of LSB coding, however, the two least significant bits of a sample are replaced with two message bits. This increases the amount of data that can be encoded but also enhances the amount of resulting noise in the audio file as well. To extract a secret message from an LSB encoded sound file, the receiver needs access to the sequence of sample indices used in the embedding process. Normally, the length of the secret message to be encoded is smaller than the total number of samples in a sound file. One must decide then on how to choose the subset of samples that will contain the secret message and communicate that decision to the receiver. One trivial technique is to start at the beginning of the sound file and perform LSB coding until the message has been completely embedded, leaving the remaining samples unchanged [6]. This creates a security problem, however in that the first part of the sound file will have different statistical properties than the second part of the sound file that was not modified. One solution to this problem is to pad the secret message with random bits so that the length of the message is equal to the total number of samples.

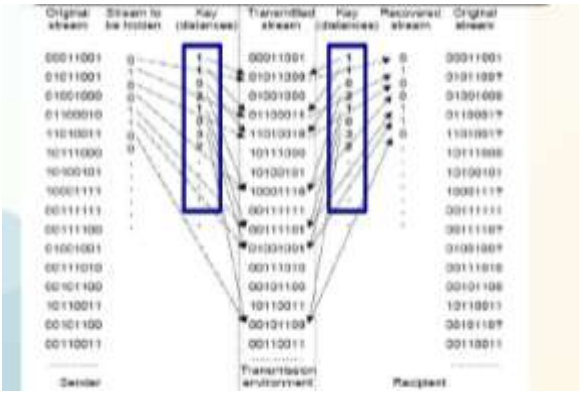
Disadvantages of LSB in audio:

There are two main disadvantages associated with the use of methods like LSB coding. The human ear is very sensitive and can often detect even the slightest bit of noise introduced into a sound file, Second disadvantage however, is that this is not robust. If a sound file embedded with a secret message using either LSB coding was resample, the embedded information would be lost. Robustness can be improved somewhat by using a redundancy technique while encoding the secret message. However, redundancy techniques reduce data transmission rate significantly.

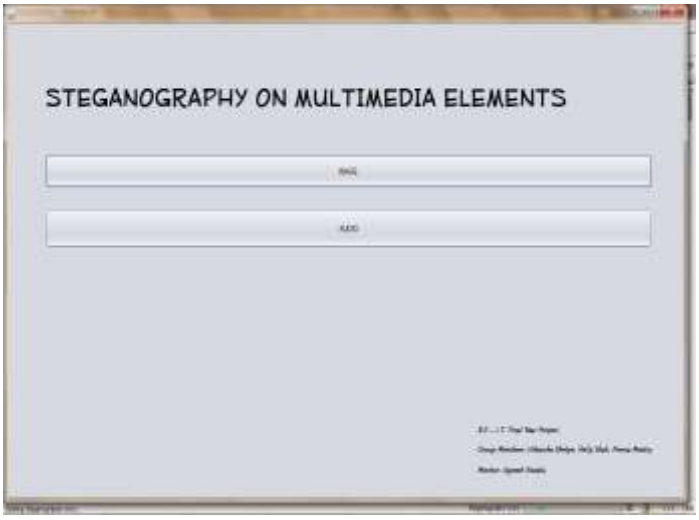
Proposed technique for Audio
Just as the proposed Image Steganography audio is quite similar. Its architecture diagram is as given below.

Algorithm steps are as follows:

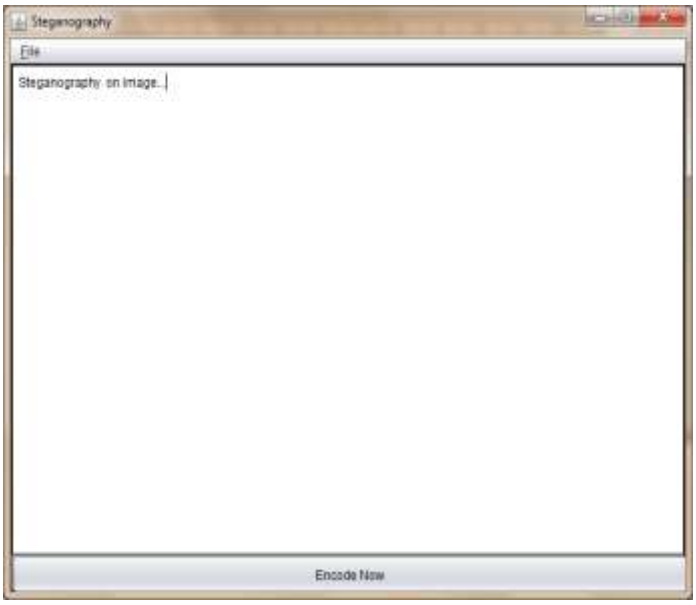
1.Select original stream of cover media.
2.Select the stream to be hidden(text message).
3.Put the key.
4.Convert key into binary
5.If 0 is encountered jump, else substitute.



IV.SCREEN SHOTS FOR IMAGE LSB ALGORITHM.



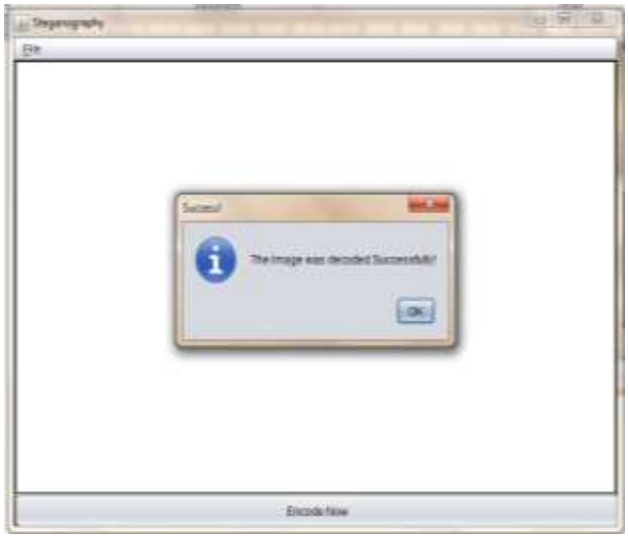
i.Open the running project homepage



ii.Enter messgae to encode

 ISSN: 2321-8169
3970 - 3975

iii.Browse the file to Use as cover and encoded image



iv.Decode the image

v.Audio Steganography

v.Encypting Audio

vi.Decrpyting Audio

V.FUTURE SCOPE:

In today's digitised world where we have better and safer ways to transfer data from one workstation to another,we have to be more careful as well.Just as a coin has two sides,similarly digitisation is to be handled carefully[7].Hacking is acommon term used these days.It can be done consciously or uncounsciously,but hackers on a rage these days.In such sensitive areas like military and classifieds we have to send data carefully[9].Thus Steganography is useful in such areas.It can be used in combination with other cryptography algorithms to send data more confidentially.

VI.CONCLUSION

In this paper we have introduced a new method of steganography using media image and audio. This system makes the sharing of data more efficient and in secured manner to share data on network and send to destination in safe way. Thus in proposed system we used key method for making system more safe for use, because of this though the hacker knows the method of decrypting data using LSB algorithm will not able to decrypt data and take advantage form it and it deals with both the media of stegnography .As increase in use of communication through digital networks the methods for security must be evolved beyond the limits. So

this system makes the further way to make better method for securing data.

REFERENCES

[1] Padmashree G, Venugopala P S" Audio Stegnography and Cryptography: Using LSB algorithm at 4th and 5th LSB layers" International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 4, October 2012 .

[2] Ankit Chadha, Neha Satam, Rakshak Sood, Dattatray Bade" An Efficient Method for Image and Audio Steganography using Least Significant Bit (LSB) Substitution" Volume 77– No.13, September 2013.

[3] Yogendra Kumar Jain, R. R. Ahirwal" A Novel Image Steganography Method With Adaptive Number of Least Significant Bits Modification Based on Private Stego-Keys"

[4] Marghny Mohamed1, Fadwa Al-Afari and Mohamed Bamatraf." Data Hiding by LSB Substitution Using Genetic Optimal Key-Permutation". International Arab Journal of e-Technology, Vol. 2, No. 1, January 2011

[5] Prof. Samir Kumar Bandyopadhyay,Sarthak Parui."A Method for Public Key Method of Steganography "International Journal of Computer Applications (0975 – 8887) Volume 6– No.3, September 2010.

[6] K.P.Adhiya Swati A. Patil."Hiding Text in Audio Using LSB Based Steganography".Information and Knowledge Management www.iiste.org ISSN 2224-5758 (Paper) ISSN 2224-896X (Online) Vol 2, No.3, 2012

[7] Namita Tiwari,Dr.Madhu Shandilya "Evaluation of Various LSB based Methods of Image Steganography on GIF File Format".International Journal of Computer Applications (0975 – 8887) Volume 6– No.2, September 2010.

[8] Ashwini Mane,Gajanan Galshetwar,Amutha Jeyakumar "DATA HIDING TECHNIQUE: AUDIO STEGANOGRAPHY USING LSB TECHNIQUE ".Vol. 2, Issue 3, May-Jun 2012, pp.1123-1125

[9] Gunjan Nehru1, Puja Dhar "A Detailed look of Audio Steganography Techniques using 402 Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved. LSB and Genetic Algorithm Approach". IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, January 2012.

[10] Jayeeta Majumder,Sweta Mangal "An Overview of Image Steganography using LSB Technique".Applications with International Journal of Computer Applications (NCACSA 2012) Proceedings published in International Journal of Computer Applications® (IJCA).

[11] Gurmeet Kaur and Aarti Kochhar "A Steganography Implementation based on LSB & DCT"International Journal for Science and EmergingTechnologies with Latest Trends 4(1): 35-41 (2012) .