INTERNATIONAL JOURNAL OF ENHANCED RESEARCH IN SCIENCE TECHNOLOGY & ENGINEERING
VOL. 2 ISSUE 3, MARCH-2013 ISSN NO: 2319-7463

Image/message Steganography using LSB insertion Algorithms for DRM system

Chandra Mohan Reddy Sivappagari
Assistant Professor of ECE, JNTUA College of Engineering,
Pulivendula, Andhra Pradesh, INDIA
email2cmr@gmail.com

Abstract: Digital Rights Management (DRM) system is the effective scheme for digital transactions. The main disadvantage of DRM is that once the content is decrypted, it can be easily copied using widely available utilities. It can be removed by using protection system like steganography. Steganography is the art of communicating in a way which hides a secret message into the main information or cover. The most common approaches to information hiding in images are in time domain, Least Significant Bit (LSB) insertion, in frequency domain using Fast Fourier Transform, in spatial domain using Wavelets. In this paper, the hiding of information of images and messages are implemented in time domain using LSB insertion method. The encoding and decoding algorithms are developed and successfully implemented in MATLAB graphical user interface. LSB insertion method in image Steganography works effectively for 24-bit–map (BMP) and PNG image formats. The proposed algorithms are simple to implement and gives better results.

Keywords: Digital rights management, Steganography, Intellectual property right, LSB insertion method.

I. INTRODUCTION

With the advent of World Wide Web and the internet technology, it is easy to access the digital content, copy righted material and even redistribute illegally by the users. So the mechanism to protect the digital content from electronic data theft, called Digital Rights Managements (DRM) is highly required in the present day world. DRM has started to emerge as a field on its own in the middle of 1990's when the internet began to commercialize. Digital Rights Management (DRM) poses one of the greatest challenges for content communities in this digital age [1]. DRM is a mechanism used to control access to copyrighted material. DRM system is the effective schemes for digital transactions.

A. Digital rights management principles:

The principles of DRM are explained by Frank Hartung et. al. [2]. The DRM pillar model is shown in Fig. 1. Like a cryptographic system, any DRM system is as strong as its weakest component. It now has been widely realized that DRM is a required core ingredient of multimedia distribution, which is why several standards bodies are active in that area and complement the available proprietary solutions. They either define the whole DRM system, or interfaces and application programming interfaces (APIs). Some important standardization groups and odies that have been working on DRM systems are the International Organization for Standardization (ISO) MPEG, Secure Digital Music Initiative (SDMI), DVD/Copy Protection Technical Working Group (CPTWG), *C, Open Platform Initiative for Multimedia Access (OPIMA), Digital Video Broadcasting (DVB), Digital Audio-Visual Council (DAVIC), Bluetooth Special Interest Group, and TV anytime.

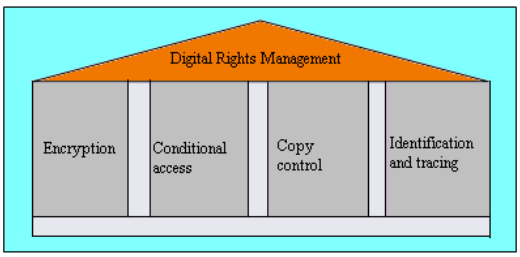

Fig. 1: DRM pillar model

A DRM system enables the secure exchange of intellectual property, such as copyright protected music, video, or text, in digital form over the Internet or other electronic media, such as CDs, removable disks, or mobile networks. Digital content in real world applications include multimedia information (audio and video), text, images etc. The digital era is a blessing and a curse at the same time. The main problem with any digital content is that anyone can make exact replica of the original without any quality degradation make the commercial value of pirated content. DRM find applications in various fields like digital broadcasting, digital library managements, copy protection, copy right protection, fingerprinting of multimedia data, authentication of digital content, etc. Industries latest technique is to authenticate during content creation, developing software/hardware tools for DRM.

Latest techniques so developed are in support to existing legislation related to Intellectual Property Rights (IPR) protection laws which protect both authors as well as consumers. This paper investigates the solution to the existing problems in the technology and the tools for DRM by developing new steganography algorithm which is not available in the literature.

II. STEGANOGRAPHY

Steganogrphy is the art / science /study / work of communicating in a way which hides a secret message into the main information or cover. Various steganography terminologies is given by B.Pfitzmann [3]. The model of steganography and cryptography are given in Fig. 2 and Fig. 3 respectively. The term steganography [4, 5] means "cover writing" whereas cryptography means "secret writing". Cryptography is the study of methods of sending messages in distinct form so that only the intended recipients can remove the disguise and read the message. The message we want to send is called plain text and disguised message is called cipher text. The process of converting a plain text to a cipher text is called enciphering or encryption, and the reverse process is called deciphering or decryption. Encryption protects contents during the transmission of the data from the sender to receiver. However, after receipt and subsequent decryption, the data is no longer protected and is the clear. Steganography hides messages in plain sight rather than encrypting the message; it is embedded in the data (that has to be protected) and doesn't require secret transmission. The message is carried inside data. Steganography is therefore broader than cryptography.

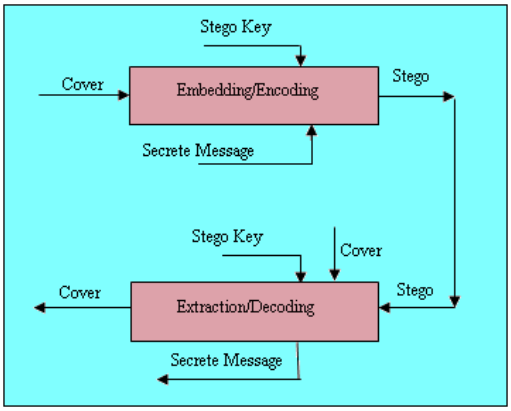

Fig.2 Steganography

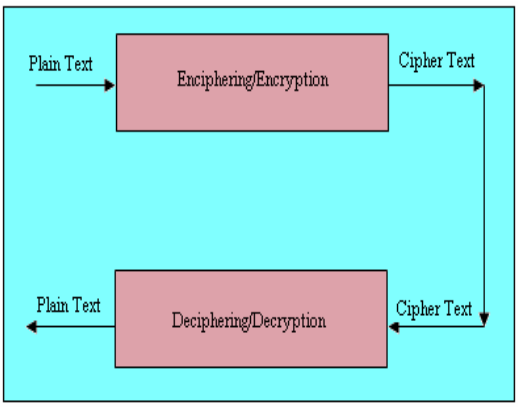

Fig. 3: Cryptography

With advanced computer software, authors of images, music and software can place a hidden "trademark" in their product, allowing them to keep a check on piracy. This is commonly known as watermarking. Hiding serial numbers or a set of characteristics that distinguishes an object from a similar object is known as fingerprinting. Together, these two are intended to fight piracy image and cryptography form of hiding data as shown in the Fig.4-6.

Fig.4: Cover image

Fig.5: Stego image

qANQR1DBwU4D/TlT68XXuiUQCADf
j2o4b4aFYBcWumA7hR1Wvz9rbv2BR6
WbEUsyZBIEFtjyqCd96qF38sp9IQiJIKlNa
Zfx2GLRWikPZwchUXxB+AA5+lqsG/ELB

Fig.6: Message

In studying Steganography in text, it is examined three main techniques line-shift coding, word shift coding, and feature coding. Each is designed to fight illegal distribution of text documents by stamping some recognizable feature into the text, by shifting the lines, shifting the word spacing, or altering characteristics of the letters themselves. It is found that some of these methods are quite strong, proving resistant to even ten levels of photocopying. An early researcher in Steganography and cryptography was Johannes Trithemius (1462-1526), a German monk, his first work on Steganography, steganographia**,** described systems of magic and prophecy, but also contained a complex system of cryptography. In random Least Significant Bit (LSB) insertion methods, a pseudo-random number generator is used to randomly distribute and hide the bits of a secret message into the LSBs of the pixels within a carrier image, called the cover image. A popular approach to achieve this is the random interval method. Both communication parties share a stego-key, usable as a seed for a random number generator. Random LSB insertions are intended to make it harder for an attacker to detect the embedded secret message with attacks such as the visual attacks and statistical attacks.

III. STEGANOGRAPHY IN IMAGES

When embedding data, Bender et al. reminds us that it is important to remember the following restrictions and features, the cover data should not be as imperceptible as possible [6]. The embedded data can directly encode into the media, rather than into a header or wrapper, to maintain data consistency across formats. The-embedded data should be as immune as possible to modifications from intelligent attacks or anticipated manipulations such as filtering and re sampling. Some distortion or degradation of the embedded data can expected when the cover data modified. To minimize this, error correcting codes should be used.

A. Characterizing Data Hiding Techniques

Steganographic techniques embed a message inside a cover; various features characterize the strengths and weakness of the methods. The relative importance of each feature depends on the application. The most important requirement is that a Steganographic algorithm has to be imperceptible. It is propose a set of criteria to define the imperceptibility of an algorithm. These requirements are as follows:

i. Invisibility: The invisibility of a Steganographic algorithm is the first and foremost requirement, since the strength of Steganography lies in its ability to be unnoticed by the human eyes. The moment one can see that an image tampered with, the algorithm compromised.

ii. Payload Capacity

Unlike watermarking, which needs to embed only a small amount of copyright information, Steganography aims at hidden communication and therefore requires sufficient embedding capacity.

B. Unsuspicious Files

This requirement includes all characteristics of a steganographic algorithm that may result in images that are not used normally and may cause suspicion. Abnormal file size, for example, is one property of an image that can result in further investigation of the image by a warden.

C. Hiding Capacity

Hiding capacity is the size of information that can be hidden relative to the size of the cover. A larger hiding capacity allows the use of a smaller cover for a message of fixed size, and thus decreases the bandwidth required to transmit the stego-image.

D. Perceptual Transparency

Preserving perceptual transparency in an embedded watermark for copyright protection is also of paramount importance because the integrity of the original work must be maintained. For applications where the perceptual transparency of embedded data is not critical, allowing more distortion in the stego-image can increase hiding capacity, robustness, or both.

IV. STEGANOGRAPHY TECHNIQUES

The most common approaches to information hiding in images are in time domain, Least significant bit (LSB) insertion, in frequency domain using Fast Fourier Transform, in spatial domain using Wavelets. Each of these can be applied to various images, with varying degrees of success. Each of them suffers to varying degrees from operations performed on images, such as resolution decrementing. Among these three techniques, here it is implemented using Least Significant Bit (LSB) Insertion method in time domain.

A. Least Significant Bit Insertion Method

Embedding data, which is to be hidden, into an image requires two files. The first is the innocent-looking image that will hold the hidden information, called the cover image. The second file is the message – the information to be hidden. A message may be plain text, cipher text, other images, or anything that can be embedded message make a stego-image. A stego-key (a type of password) may used to hide, and then later decode, the message. Most steganography software neither supports nor recommends using JPEG images, but recommends instead the use of lossless 24-bits images such as BMP. The next-best alternative to 24-bit images is 256 color or grayscale images. The most common of these found on the Internet are GIF files. In 8-bit color images such as GIF files, each pixel is represented as a single byte, and each pixel merely points to a color index table (a palette) with 256 possible colors. The pixel's value, then, is between 0 and 255. The software simply paints the indicated color on the screen at the selected pixel position.

B. Encoding the data using LSB Method

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image. The least significant bit of some or all of the bytes inside an image is changed to a bit of the secret messages. When using a 24-bit image, a bit of each of the red, green and blue color components can be used, since they each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 x600-pixel

image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data. Here it is implemented such that it can accept an image of any size. For example, a grid for 3 pixels of a 24-bit image can be follows,

(00101101 00011100 11011100)

(10100110 11000100 00001100)

(11010010 10101101 01100011)

When the number 200, whose binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows,

(00101101 00011101 11011100)

(10100110 11000101 00001100)

(11010010 10101100 01100011)

Although the number was embedded into the first 8 bytes of the grid, only the 8 highlighted bits needed to be changed according to the embedded message. The probability of matching of bit to be inserted and bit available in LSB of the pixel of the image 50% therefore, on average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. Since there are 256 possible intensities of each primary color, changing the LSB of a pixel results in small changes in the intensity of the colors. The human eye cannot perceive these changes thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the least as well as second to least significant bit and still not see the difference. In the above example, consecutive bytes of the image data-from the first byte to the end of the message are used to embed the information. This approach is very easy to detect. A slightly more secure system is for the sender and receiver to share a secret key that specifies only certain pixels to be changed. The emphasized bits are the only bits that actually changed. The main advantage of LSB insertion is that data can be hidden in the least and second to least bits and still the human eye would be unable to notice it. When using LSB techniques on 8-bit images, more care needs to be taken as 8-bit formats are not forgiving to data changes as 24-bits formats. Care needs to be taken in the selection of the cover image, so that changes to the data will not be visible in the stego-image. Such a change would be immediately noticeable on the displayed image, and is thus unacceptable. For this reason, data hiding experts recommended using gray-scale palettes, where the difference between shades is not pronounced. Alternatively, images consisting mostly of one color, such as the so-called Renoir palette, named because it comes from a 256 color version of Renoir's "Le Moulin de la Galette".

V. IMPLEMENTATION

A. LSB Encoding Algorithm

The flow chart of LSB encoding algorithm is shown in Fig. 7. First we will take the original image as shown in the Fig. 8 and the text file as shown in the Fig 6, which we have to embed into original image. Then we have to convert the text data into binary format. Binary conversion is done by taking the ASCII value of the character and converting those ASCII values into binary format and steam of bits are generated. Counter variable is taken which holds the total number of bits of message. Similarly, in cover image, bytes representing the pixels are taken in single array and byte steam is generated. Message bits are taken sequentially and then are placed in LSB bit of image byte. The index number of the image byte where replacement of LSB is to be done is controlled by polynomial equation, which is given in the key. Same procedure is followed till all the message bits are placed in image bytes. The number sequence generated by the polynomial is unique therefore identifying the image bytes where LSB encoding is done is very difficult without polynomial key. Image generated is called 'stego-image' as shown in the Fig. 9 is ready for transmission through the internet.

$$Covermedium + embeddedmessage + stegokey = stego\text{-}medium$$

B. LSB decoding Algorithm

Fig. 10 shows the flow chart of LSB decoding algorithm. First Stego-image is taken and single array of bytes is generated as performed at the time of encoding. Total numbers of bits of message and polynomial are taken from the key supplied. Counter is initially set to1 and is substituted in the polynomial which in turn gives the index number of the pixel byte where message bit is available in LSB. The procedure is repeated till final count of message bits is reached. After this step we have a bit steam of the message. Available bits are framed to form bytes such that each byte represents single ASCII character. Characters are stored in text file which represents the hidden message.

C. Set of polynomials:

$$x^2+x+1;\quad x^3+x+2;\quad 2x^2+x+1;\quad x^2+2x+1;\quad 3x^2+x+1;\quad 2x^3+x+2$$
$$x^2+4x+1;\quad x^3+2x+1;\quad 2x^2+3x+1;\quad 3x^2+2x+1;\quad 3x^2+x;\quad 2x^3+x+2$$

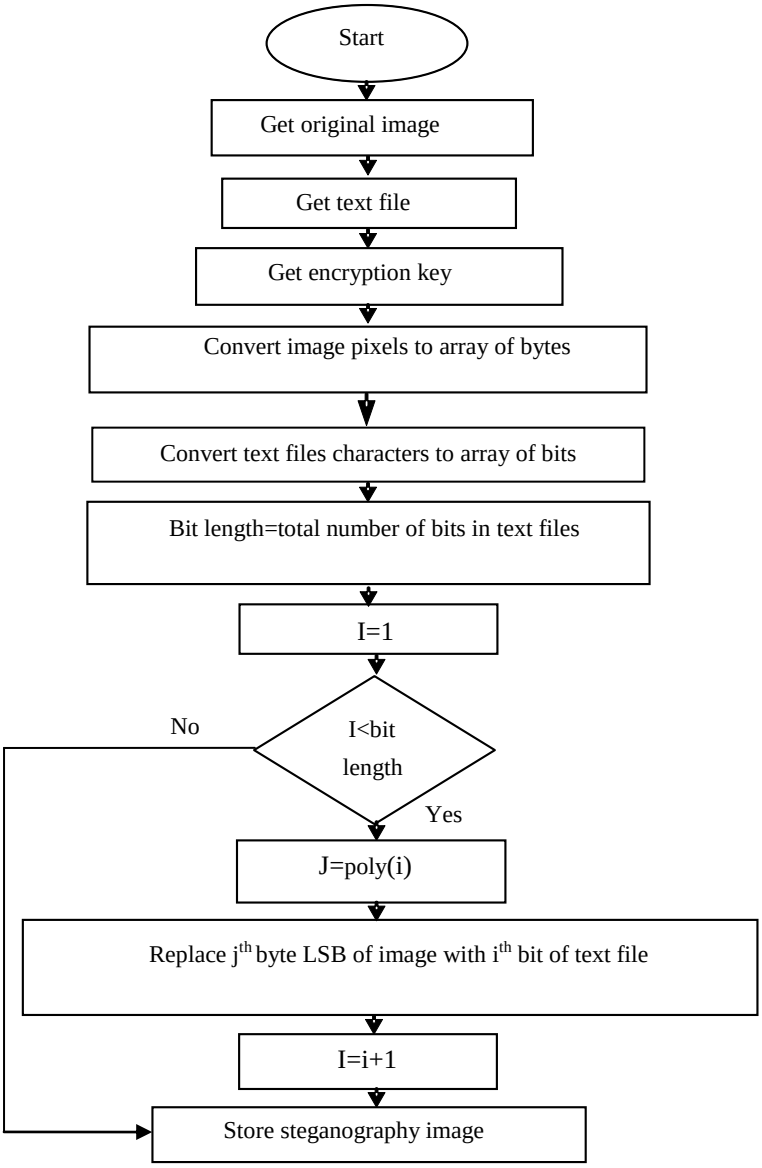

Fig.7: Flow chart of LSB Encoding algorithm

Fig. 8: Input image

Fig.9: Stego Image

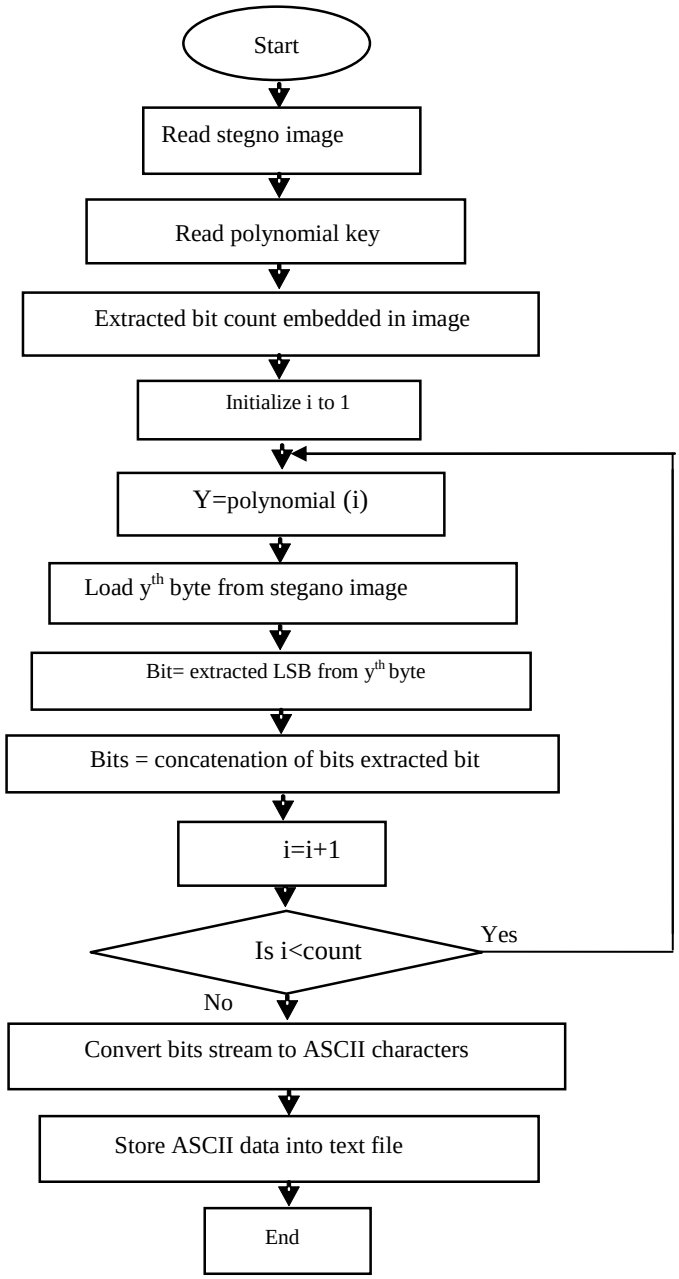

Fig. 10: Flow chart of LSB Decoding algorithm

VI. RESULTS

A. Encoding the Image

Fig. 11 shows the Graphical User Interface (GUI) designed using MATLAB to encode the image. Fig. 12 and Fig.13 shows the encoded image and stego image. The text message to be encoded in the input image is shown in Fig. 14.

Fig.11: GUI to Encode the Image

Fig.12: Input Image

Fig.13: Stego image

Fig.14: Text message

B. **Decoding the Stego Image**

Fig.15 shows the GUI designed using MATLAB to decode the stego image. The decoded stego image is shown in Fig. 16.

Fig.15: GUI to Decode the Image

Fig.16: Decoded Image

VII. CONCLUSIONS

The advantages of LSB embedding are its simplicity, many techniques use these methods. LSB embedding also allows high perceptual transparency. The data hiding capacity of this technique is high, more secure however, there are weaknesses when robustness, tamper resistance are considered. Scaling, rotation, cropping, addition of noise, or loss compression to the stego-image is very likely to destroy the message. Steganography has its place in security. It is not intended to replace cryptography but to supplement it. Hiding a message with Steganography methods reduces the chance of a message being detected. It goes well beyond simply embedded text in an image. Steganography does not only pertain to digital images but also other media. LSB insertion method in image Steganography works effectively for 24 BMP and PNG image formats. To hide the information in JPEG images we need to convert stego-image into BMP format and this BMP file is transmitted, but this BMP file should not be converted back into JPEG format. The strength of the stealth can be improved further by using multiple polynomials.

REFERENCES

[1]. Renato Innella, "Digital Rights Management (DRM) Architectures", D-Lib agazine, 2001, Vol. 7, No.6, pp. 1-10.
[2]. Frank Hartung et al, "Digital Rights Management and Watermarking of multimedia Content for M-Commerce Applications", IEEE Communications Magazine, pp.78-84, IEEE 2000.
[3]. B.Pfitzmann, "Information Hiding Terminology", Proc. of First Inter. Workshop on Information Hiding, Cambridge, UK, May30-June1, 1996, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.), pp.347-350.
[4]. David Kahn, "The History of Steganography", Proc.of First Int. Workshop on Information Hiding, Cambridge, UK, May30-June1 1996, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.), pp. 1-7.
[5]. E. Franz, et. al., "Computer Based Steganography", Proc. First Intl. Workshop on Information Hiding, Cambridge, UK, May 30 - June 1, 1996, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.).
[6]. W. Bender, D.Gruhl, N.Morimoto and A.Lu, "Techniques for data hiding", IBM Systems Journal, vol.25, 1996, pp.313-335.