Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 7 (2017) pp. 2069-2078

http://www.ripublication.com

A novel technique for Reversible Information Hiding

Geeta Sharma
Jagannath Institute of Management Studies
JIMS, Rohini, New Delhi, India.

Vinay Kumar
Professor, Vivekananda School of Information Technology
VIPS, Pitampura, New Delhi, India.

Abstract

The data in transit on the Internet is quite vulnerable to the various attacks. Mostly, an encrypted data is sent to avoid unauthorized interrupt. Digital watermarking allows information to be hidden and embedded in multimedia data like text, images, audio, video etc. Whereas reversible watermarking schemes are considered lossless as extracting the watermark does not affect the original image and it can be restored completely. We can classify the watermarking techniques according to various criteria like domain of the data, its visibility to the attackers, robustness and also according to the technique used for the retrieval at the receiver end. In this paper, we will consider approaches from different domain, like spatial or transform, for hiding information in some cover image. This paper proposes a novel reversible data hiding approach that can extract the hidden information from original image without any loss of the information. We will also discuss about lossy and lossless image format, where stressing on 8-bit and 24-bit, .bmp and .jpeg images. The hidden message will be converted in message digest that is secure hash code using MD5. This code will be stored in a digital image in such a way that it is not human perceptible.

Keywords: Watermarking, message digest, steganography, spatial domain, transform domain

I. INTRODUCTION

Steganography is the science of hiding the message in a carrier from human eye perception. It has many branches that include watermarking. A watermark, in general, is used for authorization and to avoid the counterfeit or fraud. A digital watermark is different as it is used for the protection of the copyright or license. It is used to hide the message in the multimedia data that can be text, audio, images etc. The changes in multimedia data are generally not visible. We can classify the watermarking techniques according to various criteria like domain of the data, its visibility to the attackers, robustness and also according to the technique used for the retrieval at the receiver end. The digital medium used for carrying the digest is known as cover. When the cover cannot be recovered at the receiver's end while extracting the digest, the technique is called Irreversible watermarking. In reversible digital watermarking, the cover can be accessed and used without any tampering to the data. This technique is also called lossless as we get the original image of cover without any loss of information

Watermarking is generally used to provide the proof of ownership of digital data. It is generally achieved by embedding some copyright information with digital data. Although it can be used for automatic monitoring of copy-write material on the World Wide Web. It helps in automatic audit of radio transmission that can be tracked easily during its broadcast. Watermarking also helps in embedding additional information for the advantage of public – data augmentation. Essentially, it has appeared as the important technology to solve above mentioned problems. [1] Its techniques can be divided into following domains:

A. Spatial Domain Methods: A spatial domain method is used to directly change the pixel values of the bits in the digital image for hiding some information. Least significant bit (LSB)-based steganography is one of the simplest techniques that hides a secret message in the LSBs of pixel values without introducing many perceptible distortions. Changes in the value of the LSB are imperceptible for human eyes. [1]

B. Transform Domain Technique: This technique is complex than spatial as various algorithms and transformations are used to hide digest. Transform domain embedding can be termed as a domain of embedding techniques for which a number of algorithms have been suggested [3]. The process of embedding data in the frequency domain of a signal is much stronger than embedding principles that operate in the time domain. Most of the strong steganographic systems today operate within the transform domain Transform domain techniques have an advantage over spatial domain techniques as they hide information in areas of the image that are less exposed to compression, cropping, and image processing.[1]

II. LITERATURE REVIEW

Digital watermarking techniques have been indicated so far as a possible solution when, in a specific application scenario (authentication, copyright protection, fingerprinting, etc.), there is the need to embed an informative message in a digital document in an imperceptible way [13]. Many researchers had proposed various techniques including a milestone form Tian[9]. It presents a high-capacity, high visual quality, and reversible data embedding method for grayscale digital images. This method calculates the difference of neighboring pixel values and then selects some of such differences to perform a difference expansion (DE). In such different values, a payload B made by the following parts will be embedded:(i) a JBIG compressed location map, (ii) the original LSB values, and (iii) the net authentication payload which contains an image hash. Tian[9] The method defines different kinds of pixel couples according to the characteristics of the corresponding h and behaves slightly different for each of them during embedding. Two are the main categories: changeable and expandable differences.

The previous method has been taken and extended by Alattar[10]. Instead of using difference expansion applied to pairs of pixels to embed one bit, in this case difference expansion is computed on spatial and cross-spectral triplets of pixels in order to increase hiding capacity; the algorithm embeds two bits in each triplet. Here a triplet is a term meant for a vector of size 1 X 3. This vector contains the value of colored image at pixel level. It can be classified as :

A. Spatial Triplet: First of all, an order is maintained to select the components of same color from the image. Then the three pixels (1X3) is selected from it.

B. Cross-spectral Triplet: In this triplet, value of 3 color components i.e. Red, Green, Blue is selected and stored in the triplet.

De Vleeschouwer et al. proposed in [11], a semi-fragile algorithm based on the identification of a robust feature of the luminance histogram for an image tile. As for the patchwork approach, the cover media is tiled in non-overlapping blocks of pixels that are associated to a bit of the embedded message. For a single block, the pixels are equally divided into two pseudorandom sets (i.e., zones A and B) and for each zone the luminance histogram is computed and mapped around a circular support. [11] A weight, proportional to the occurrence of each luminance value, is placed on the corresponding position of the circle and then a center of mass is calculated and localized respect to the center of the circle.

Since zones A and B are pseudo-randomly determined, it is highly probable that the localization of the corresponding centers of mass is very close to each other. This peculiarity can be exploited to embed a bit by simply rotating the center of mass of the A and B zones in opposite ways. A clockwise rotation of the A zone center of mass can be associated to the embedding of a bit “1,” while an anticlockwise rotation can be

associated to a bit "0." The B zone is rotated in the opposite direction accordingly to the technique previously presented. By using this approach, it is very easy to determine, during the watermark detection, if a "1" or "0" bit is embedded in a certain block and, eventually, remove the mark by counter rotating the histogram along the circular support.

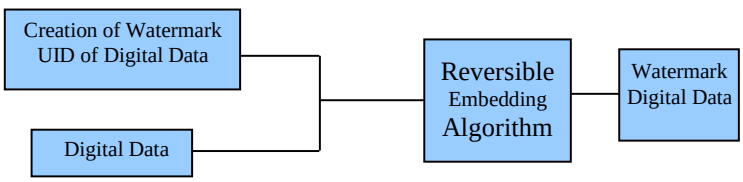
In a real image, some pathological cases can arise when the two centers of mass are not properly positioned and in general do not respect the mutual nearness. These cases are statistically negligible and do not affect significantly the available watermark payload [13].

In Ni et al. [12], an algorithm based on the De Vleeschouwer idea is proposed in order not to be fragile to JPEG compression. This method is based upon an analysis of the differences between couples of pixels belonging to an image tile.

An image tile is divided into pixel couples and a sum of differences of their luminance values (taken in an ad hoc manner) is computed. A statistical analysis shows that this computed value (named α) is very close to zero for most of the tiles. The main idea for bit embedding is that the difference value α is related to a reference value K (usually less than 5 according to numerous experiments) and a categorization of the α value respect to the K value is carried on[13]. The categorization is defined even by means of the parameter β that is usually $\beta > 2 \cdot K$. This approach is aimed to avoid falling into underflow/overflow errors that would significantly lower the stego image quality. In particular, four categories are identified [13].

III. METHODOLOGY PROPSED

The key theme of the proposed study is to present a solution for recovering the compressed image back to original form. In this work, we will discuss about reversible information hiding. This includes generation of hash code, that is, digest of the data to be hidden in the digital image. We will use MD5 for this purpose. The unique ID of digital data is created. This digest or hash code is then embedded with digital data using reversible embedding algorithm. Generally, the cover is not used after extracting the message. Here, we will work on reversible embedding algorithm that will allow us to use the cover even after the extraction of hidden message. The following figure can explain the concept:



Now, we will see the major issues in detail.

A. Digital Image Formats

Image file size is completely interrelated to the number of pixels in an image and the color depth, or bits per pixel, of the image. Compression is the method of storing an exact representation or an approximation of the original image in enormously smaller number of bytes. This compressed image can be expanded back to its original form. The image file compression algorithms can be classified into two broad categories:

a.) Lossless compression algorithms reduce file size while preserving a perfect copy of the original uncompressed image. Lossless compression generally, but not always, results in larger files than lossy compression. Lossless compression should be used to avoid accumulating stages of re-compression when editing images [4]. The BMP file format Lossless (no image data is lost on save) but there's also little to no compression at all, meaning saving as BMP results in very large file sizes.[4] Bitmap file formats can be saved in four different bit depths:1-bit (monochrome), 4-bit (16 colors),8-bit (256 colors), 24-bit (16 million colors) and 32-bit format. In this document, I only consider the most common 24-bit and 8-bit uncompressed bitmap formats.

1. 8-bit bitmap format (windows): The .bmp file format requires that the data along a scan line in the image be aligned on a 4-byte boundary. Thus, the width of an 8-bit image must be a multiple of 4. 8-bit images are a great format to use because of their small size. The drawback is that only 256 possible colors can be considered which can be a potential problem during encoding. The structure of an 8-bit Windows bitmap includes:

- BMP File Header: This data structure contains information about the .bmp file as a whole i.e. BMP identifier, size, location etc.
- BMP Info Header: This data structure contains information about the .bmp data i.e. size of BMPINFOHEADER structure, height and width of bitmap, compression flag etc.
- Palette: This is an array of RGBQUAD structures, each of which is a color. The RGBQUAD structure contains (rgbRed, rgbGreen, rgbBlue, rgbReserved).
- Image Data: This is a one-dimensional array of unsigned characters, where each value is a pixel stored in (b, g, r) format.

2. 24-bit bitmap format (windows): A 24-bit bitmap has no palette and writes a (b, g, r) for each pixel in the image. Thus, 3 bytes are stored for each pixel. The structure of a 24-bit Windows bitmap includes:

- BMP File Header: This data structure contains information about the .bmp file as a whole i.e. BMP identifier, size, location etc.
- BMP Info Header: This data structure contains information about the .bmp data i.e. size of BMPINFOHEADER structure, height and width of bitmap, compression flag etc.
- Image Data: This is a one-dimensional array of unsigned characters, where each value is a pixel stored in (b, g, r) format.

b.) *Lossy compression algorithms* preserve a representation of the original uncompressed image that may appear to be a perfect copy, but it is not a perfect copy. Often lossy compression is able to achieve smaller file sizes than lossless compression [4]. Generally a lossy compression algorithm can result in variable compression with variable image quality and file size. JPEG stands for "Joint Photographic Expert Group". JPEG images were designed to make detailed photographic images as small as possible by removing information that the human eye won't notice. As a result it's a Lossy format, and saving the same file over and over will result in more data being lost over time. JPEG does this by taking advantage of the fact that the human eye notices slight differences in brightness more than slight differences in color. The amount of compression achieved is therefore highly dependent on the image content; images with high noise levels or lots of detail will not be as easily compressed, whereas images with smooth skies and little texture will compress very well.

B. ZERO BYTE

Block cipher algorithms (like DES) require their input to be an exact multiple of the block size. If the plain text to be encrypted is not an exact multiple, you need to pad before encrypting by adding a padding string. When decrypting, the receiving party needs to know how to remove the padding in an unambiguous manner [7]. Byte padding can be applied to messages that can be encoded as an integral number of bytes. All the bytes that are required to be padded are padded with zero. The zero padding schemes have not been standardized for encryption. Zero padding may not be reversible if the original file ends with one or more zero bytes, making it impossible to distinguish between plaintext data bytes and padding bytes. It may be used when the length of the message can be derived out-of-band. It is often applied to binary encoded strings as the null character can usually be stripped off as whitespace. Zero padding is sometimes also referred to as "null padding" or "zero byte padding". Some implementations may add an additional block of zero bytes if the plaintext is already divisible by the block size.

The plain text is the ASCII code for "God is great for me". To encrypt, we break up the plaintext into blocks of 8 bytes (Note we are using 8 in this example)

|God_is_g|reat_for|_me?????|

|block 1- |block 2- |block 3-|

We need to pad the block with padding bytes to make it up to the required length, by padding it with 0x80 followed by zero bytes. That is:

INPUT BLOCK = _ m e _ _ _ _ _

(IN HEX) 66 6F 72 80 00 00 00 00

C. Creation of Digest

Hashes are a series of mathematical functions used to cover a base of encryption. Encryption is a method used to make human understandable plain text and transforms it into cipher which human's can not recognize. We will use any of these hashing algorithms*:*

a.)Message Digest Algorithms: Message Digest Algorithms are some of the strongest algorithms available, and the only way to attack the encryption is by randomly guessing and checking. This is known as a "Brute Force Attack". Since the computer can do this faster than us with predefined variables and character sets, depending on your processor speed, you can attempt anywhere from 100 passwords/sec to hundreds of thousands of passwords/sec. [5] MD5 still uses a 128 bit hash value, and is widely used today. MD5 is the most resistant to attacks - the most common attacks in which can be performed on MD5 today are Brute force attacks, mentioned early, and, Rainbow Table Cracking - a technique which revolves around a wide basis of pre-defined variables, is intensely faster than a brute force attack. Though, the problem with brute force and rainbow table attacks on MD5 strings are, it's just a game of chance - it depends on what the actual variable behind the MD5 hash actually is. MD5 hashes are depicted and recognized with 32 characters within the hash. Though MD5 is commonly used, it does not mean it is not vulnerable [5]. There have been reports of spoofing, since MD5 is commonly used upon SSL certificates; and by 2010, major security companies who care about cryptographic datum will be moving to SHA2. Reports of Collision Attacks have infact been seen on MD5, leaving its validity of secure questionable.

b.)Secure Hash Algorithms: Secure Hash algorithms are some of the most respected algorithms used today. Developed by the NSA, there are three branches into which SHA falls under. SHA0, SHA1, SHA2 - SHA0 being the least secure, SHA2 being the most secure [5]. SHA0 and SHA1 both output 160 bit values, with a block size of 512. As the message digest used fewer rounds, SHA uses 80 rounds of XOR and ROT inclusion. SHA0 has technically been cracked, as, collisions were found [5]. SHA-1 is an authentication algorithm that produces a 160-bit hash from a message of arbitrary length. It is generally regarded as more secure than Message Digest 5 (MD5) because of the larger hashes it produces.

Recent attacks presented for Secure Hash Algorithm -1 (SHA -1) and Message Digest 5 (MD5) have drawn attention of researchers and information security managers to look for a way to secure it. Visibility of information, whether encrypted or not, provides worst kind of vulnerability. Possibility of finding collisions by attackers can be made practically infeasible if statistics related to message digest is denied by hiding the digest. Complicating the process of digest creation may solve the problem temporarily [6].

CONCLUSION

Digital dataset needs to be protected against unauthorized replication to protect commercial and intellectual interest of the creator. An authentication mechanism needs to be developed so that digital dataset can be watermarked, can be retrieved when required and used for authorization. For security reason, it is desirable to hide the watermark. In such situation, information hidden can be used to hide unique identifier (message digest) of a digital dataset. The message digest can be obtained using various algorithms. In this process, the cover used for hiding the message is generally left unused. The reversible steganography is the process that emphasis on the use of cover

ACKNOWLEDGMENT

We are very grateful to all those who have been constantly encouraging and providing required resources for such scientific research work besides the regular work which we are doing in our respective departments.

AUTHOR[S] BRIEF INTRODUCTION

Dr. Vinay Kumar is working as a Professor in VIPS, Vivekanand school of information technology, Pitampura, Delhi. Before joining VIPS, he was working as a Scientist in National Informatics Center, MoCIT, Government of India. He has

authored a book on Discrete Mathematics. His area of interest is graph algorithm, steganography, discrete mathematics, data security and privacy.

Geeta Sharma is working as an Associate Professor in Jagannath Institute of Management Studies, Rohini, Delhi. Her area of interest is steganography, digital watermarking, data security and DBMS.

REFERENCES

[1] Geeta Sharma, Vinay Kumar " Reversible Information Hiding and its Application in Watermarking" in ICRDESM 2015. IJIACS, ISSN- 2347-8616,

[2] Neil F. Johnson, Zoran Duric, Sushil Jajodia "Information Hiding-Steganography and Watermarking-Attacks and Countermeasures" Ed. III, Kluwer Academic Publisher, 2000 ISBN 0-7923-7204-2

[3] "The GIS Spatial Data Model." ESRM 250, The University of Washington Spatial Technology, School of Forest Resources. 20th April. 2013 <https://courses.washington.edu/gis250/lessons/introduction_gis/spatial_data_model.html>

[4] "Image file formats." Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 1 February 2014. Web. 25 January 2014. <http://en.wikipedia.org/wiki/Image_file_formats>

[5] Curran Higgins. "Cryptography and Hashes". Yahoo Contributor Network. Share your voice on Yahoo websites. Jun 30, 2009. Web. 1 February 2014. <http://voices.yahoo.com/cryptography-hashes-3642121.html>

[6] Kumar V., Muttoo S.K. "Graph Theoretic Approach to Steganography to Secure Message Digest" Information Security Journal A Global Perspective 01/2010; 19:328-335

[7] "Using padding in encryption" DI Management Services Pty. Ltd.June , 2013 <http://www.di-mgt.com.au/cryptopad.html>

[8] "The GIS Spatial Data Model." ESRM 250, The University of Washington Spatial Technology, School of Forest Resources. 20th April. 2013 <https://courses.washington.edu/gis250/lessons/introduction_gis/spatial_data_model.html>

[9]J. Tian, "Reversible data embedding using a difference expansion," IEEE Transactions on Circuits and Systems for VideoTechnology, vol. 13, no. 8, pp. 890–896, 2003.

[10] A. M. Alattar, "Reversible watermark using difference expansion of triplets,"

in Proceedings of International Conference on Image Processing (ICIP ’03), vol. 1, pp. 501–504, September 2003.

[11] C. de Vleeschouwer, J. F. Delaigle, and B. Macq, “Circular interpretation of bijective transformations in lossless watermarking for media asset management,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 11, pp. 1423–1429, 2006.

[12] Z. Ni, Y. Q. Shi, N. Ansari,W. Su, Q. Sun, and X. Lin, “Robust lossless image data hiding designed for semi-fragile imageauthentication,” IEEE Transactions on Circuits and Systems forVideo Technology, vol. 18, no. 4, pp. 497–509, 2008.

[13] Roberto Caldelli,Francesco Filippini,& Rudy Becarelli, “ReversibleWatermarking Techniques:An Overview and a Classification” Hindawi Publishing Corporation. EURASIP Journal on Information Security Volume 2010, p.p. 19