International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

Assessing Image Steganography Techniques

Riddhi Chavda[Å*], Amit Doshi[Å] and Khushali Deulkar[Å]

[Å]Computer Engineering, D.J.Sanghvi College of Engineering,Mumbai University, Mumbai, India.

Accepted 10 Nov 2014, Available online 01 Dec 2014, **Vol.4, No.6 (Dec 2014)**

Abstract

Steganography is the art of hiding information innocuous looking objects like images, files etc. This paper explains Steganography, various techniques of Steganography and compares them. The focus of this paper is image steganography. This paper evaluates the different algorithms for digital image steganography both in the spatial and transform domain like F5, Pixel Indicator Technique, OPAP, LSB substitution etc. It compares those algorithms in terms of speed, accuracy and security and suggests some possible future research directions.

Keywords: *Steganography, LSB substitution, F5, OPAP, Pixel Indicator Technique.*

1. Introduction

In the recent times, the need for digital communication has increased dramatically and as a result the Internet has essentially become the most effective and fast media for digital communication. However, information over the internet has become vulnerable to eavesdropping, hacking etc. and thereby a need for secret communication has emerged. As a result a new domain dealing involving hiding of information has evolved. The word 'Steganography' is composed of Greek words 'Steganos' and 'Graphein', which mean 'Concealed Writing'. The advantage of Steganography over Cryptography, is that the cover objects do not attract unwanted attention. The goal of Steganography is to conceal the existence of the information in the message. Use of multimedia data over the Internet has grown exponentially in the last decade. This has further fast-tracked the research effort devoted to steganography (Ratnakirti Roy, Suvamoy Changder, Anirban Sarkar, Narayan C Debnath, 2013).

1.1 Types

1.1.1 Image

- In image steganography, the cover medium is an image, that is, any hidden file (text, image, etc.) is embedded in an image.

Fig 1.2 The carrier image **Fig 1.3** The extracted image

1.1.2 Text

- Here, the cover file is a text file. Thus, you insert your secret message by using techniques like modifying text format, variable spaces, different alignments etc.

1.1.3 Audio

- The secret message is embedded in an audio file. Technique used in audio steganography is disguising, which takes advantage of the ability of the human ear to conceal information inconspicuously. Soft audible sound can go undetected in the presence of another loud audible sound. It is not preferable because of large size of audio files.

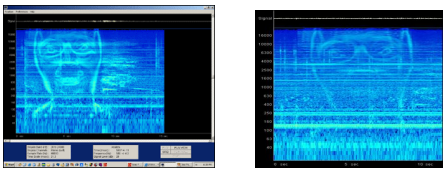

Fig 1.2 Example of Audio Steganography

1.1.4 Network

- It includes all information hiding techniques used to exchange steganograms in networks. Communication protocols' control elements are used by network steganography. It is harder to detect and eliminate. In network steganography properties of a single network protocol are modified according to the needs. Modifications are applied to the Protocol Data Unit (PDU), to the time relations between the exchanged PDUs. It can be even applied to both.

1.2 Methodology

Ancient Steganographic Techniques:

*Corresponding author: **Riddhi Chavda**

- In ancient Greece, people used to write messages on piece of wood, then cover it with wax. Upon this wax, an innocent message would be then written.
- Concealed messages on messenger's body.
- Messages were written on the backs of the couriers using invisible ink, during World War 2.
- Secret inks were used to write messages on blank parts of the paper.

Table 1.1 5x5 Tap Code Used By Armed Forces Prisoners in VietNam (James C. Judge, 2001)

	1	*2*	*3*	*4*	*5*
1	A . .	B . ..	C, K . …	D . ….	E . …..
2	F .. .	G	H .. …	I .. ….	J .. …..
3	L … .	M … ..	N … …	O … ….	P … …..
4	Q … .	R …. ..	S …. …	T …. ….	U …. …..
5	V ….. .	W ….. ..	X ….. …	Y ….. ….	Z ….. …..

- Messages were knitted using Morse Code on to a cloth, which was knitted to another piece of cloth, worn by couriers.
- Messages were written on a small area on envelopes, and then concealed underneath postage stamps.

2. Review of Literature

The algorithms for image steganography are primarily classified into two major parts based on whether the pixels of the image are modified directly or some mathematical transform is applied on the images before embedding. The former techniques are spatial domain techniques while the latter are transform domain techniques (Ratnakirti Roy, Suvamoy Changder, Anirban Sarkar, Narayan C Debnath, 2013).

Image definition (R.Amirtharajan, Sandeep Kumar Behera, Motamarri Abhilash Swarup, Mohamed Ashfaaq K and John Bosco Balaguru Rayappan, 2010):

An image is a grid of numbers that represent light intensities in different parts of the image. Individual points of the grid are called as pixels. Most images consist of a matrix of the image's pixels, which tells us where each pixel is located, and what colour it has.

Bit depth denotes the number of bits used for each pixel. 8 is the smallest bit depth. It means 8 bits describe intensity of each colour. Monochrome and greyscale images use 8 bit depth and have 256 shades of intensities. Colour images use RGB model, which uses 24 bits, 8 bits for each colour. Thus 1 pixel can have 256 shades of R, G and B adding up to more than 16-million combinations.

2.1 Spatial domain techniques

2.1.1 Least significant bit substitution

Least Significant Bit (LSB) substitution is the method in which the least significant bit pixels of the cover image are manipulated. It is a straight forward approach for embedding information into the image. The LSB substitution depends upon the number of bits in an image. In an 8 bit image, the least significant bit (bit no 8) of each byte of the image is replaced with the bit of secret message. In 24 bit image, the values of each component like RGB are altered. LSB is effective with BMP images since BMP images are not compressed. For concealing a message inside BMP images, large images are required. LSB substitution can also be applied to GIF images. However, in GIF image, whenever the LSB is altered, entire colour palette can change. A solution to this problem is using only the gray scale GIF images. Since, gray scale has only 256 shades, the changes will be hard to detect For JPEG images, direct substitution is not possible as it uses lossy compression.

It's the simplest and surprisingly effective way of concealing information in an image. In LSB method, the LSB of image is replaced by MSB of information.

Fig 2.1 Example of LSB substitution

Steps:

i. Load the cover image and image to be hidden.

ii. Choose the bits you wish to hide the secret image in. More information bits used in host image reduces quality of host image.

iii. Combine pixels from both images and create a new image. For example, use 4 bits from secret image, 4 bits from host image.

Host Pixel: 10111001

Secret Pixel: 10111111

New Image Pixel: 10111011

iv. To get back the original image we need to know how many bits were used to store the secret image. Scanning the host image, extract the LSBs according to the number used and create a new image with the bits extracted now becoming the MSBs.

Host Pixel: 10111011

Bits used: 4

New Image: 10110000

Operation for embedding of LSB substitution algorithm is

$$Y_i = 2\frac{X_i}{2} + m_i \quad (1)$$

where mi, Xi, Yi are the i-th message bit, value of selected pixel before embedding and value of modified pixel after embedding respectively.

2.1.2 Optimal Pixel Adjustment Procedure (OPAP):

The OPAP (R.Amirtharajan, Sandeep Kumar Behera, Motamarri Abhilash Swarup, Mohamed Ashfaaq K and John Bosco Balaguru Rayappan, 2010) alters the embedded bits to improve the visibility of the stego-image. Pixel difference (δi), the difference between original pixel (pi) and the pixel (pi´) of the stego-image is calculated.

Adjustment is made on the basis of the pixel difference. It helps to minimize the variation between the original and embedded stego pixel.

Procedure

Step1: LSBs are substituted within data to be hidden.
Step2: The pixels are arranged in a manner of placing the hidden bits before the pixel of each cover image to minimize the errors.
Step3: Let 'n' LSBs be substituted in each pixel.
Step4: Let d=decimal value of pixel after substitution.
d1 = decimal value of last n bits of the pixel.
d2 = decimal value of n bits hidden in that pixel.
Step5: If (d1~d2)<=(2^n)/2
then no adjustment is made in that pixel.
Else
Step6: If(d1<d2)
d = d – 2^n.
If(d1>d2)
d = d + 2^n.
'd' is first converted into binary and then written back to the pixel.
The encoder algorithm is as given below:
1: for i = 1, ..., len(msg) do
2: p = LSB(pixel of the image)
3: if p != message bit then
4: pixel of the image = message bit
5: end if
6: end for
It works by obtaining the LSB value of first pixel of image. This is achieved by calculating the modulus 2 of the pixel value. It will return zero if the number is even, and one if the number is odd, which is the LSB value. This value is compared with the message bit. If they are equal, then we leave it as it is, else then we replace the pixel value with the message bit. This process continues till there are no values left to be encoded.
The decoder algorithm is:
1: for i = 1,..., length(image string) do
2: message string = LSB (pixel string of the image)
3: end for
The decoding phase is even simpler. In the encoding phase, we replaced the LSBs of the pixel values in c in sequence. So we already know the order in which data should be retrieved. So we just need to calculate the modulus 2 of all the pixel values in the image, and we can find our hidden image. After converting this to ASCII, the message will be readable up to the point that the message was encoded. Afterwards, it appears gibberish.

2.1.3 Pixel Indicator Technique (PIT):

PIT (Adnan Abdul-Aziz Gutub, 2010) is a modification of the LSB method of embedding and it primarily aims at enhancing the security of the present LSB scheme. It works on 24 bit RGB images. In this technique, two LSB of one colour channel are marked to indicate the presence of data in the other two channels. The size of secret information is used as a key for choosing the selection channel. PIT produces very low visual distortion when the rate at which data is embedded is less than 3 bits and has low vulnerability to histogram and visual attacks at this rate.

PIT uses the pixel value as an indicator to see whether a data bit is embedded or not in that pixel. In grayscale, pixel value ranges from 0-255. Binary bits in locations specified will determine bits to be embedded. PIT technique works well for colour images.

PIT with same pixel used as indicator and channel:
A single pixel of cover image is used as indicator and the secret data bits are embedded in that pixel itself. The method adapted here is using the 3rd, 2nd bit of the pixel as indicator and 1st, 0th bit as channel for hiding information. The maximum bits that can be embedded per pixel is 2 & minimum is 0 bpp (bits per pixel).
Let Pi = b7 b6 b5 b4 b3 b2 b1 b0
Here, b3, b2 are indicator pixels & b1, b0 are channels.
The PIT algorithm for embedding and retrieval are as shown in the following flowcharts.

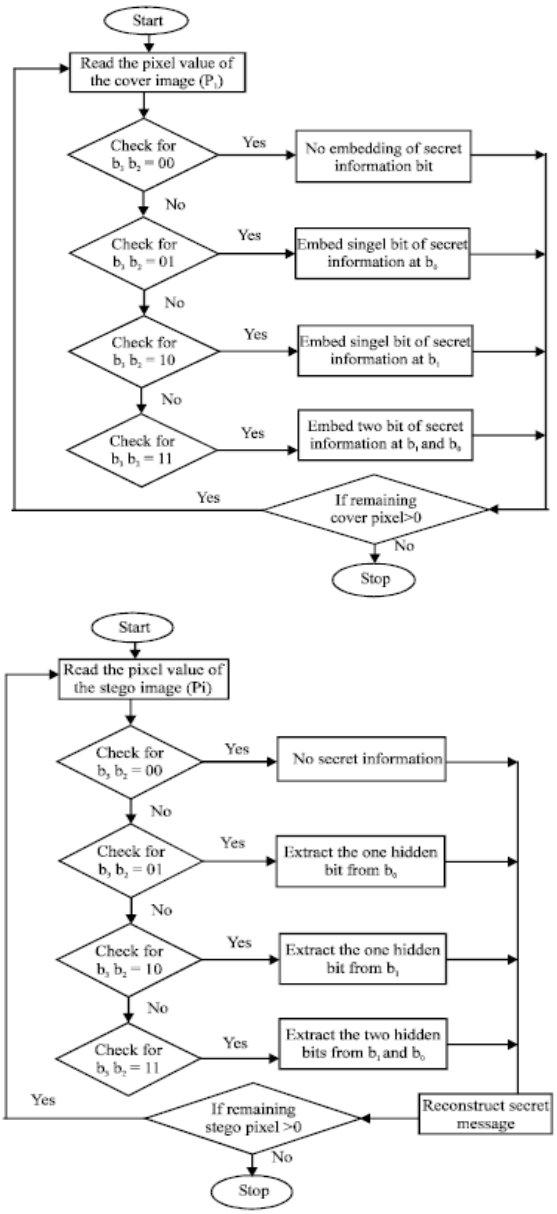

Fig 2.4: Flow chart for (a) Embedding (b) Retrieval (PIT considering same pixel as indicator and channel)

2.1.4 Pixel Value Differencing (PVD)

In PVD scheme, number of insertion bits in PVD depends on whether the pixel is an edge or a smooth area. The difference between pixels in smooth areas is much less as compared to that between the edge pixels. Human vision is sensitive to subtle changes in the smooth areas as compared to edges. So embedding in edge pixels causes less visual distortion. PVD does not cause much visual distortion and it is not directly susceptible to the histogram attack. It is vulnerable to histogram analysis of the differences of the pixel pairs and $\chi 2$-attack.

2.1.5 Selected LSB (SLSB) algorithm

SLSB embeds into single colour components of the pixels. It does not embed into the LSBs only, it also chooses the colour plane and its modifiable bits to produce minimum distortion. A sample pair analysis filter is applied before embedding to ensure that only the best candidate pixels are selected for embedding. Its' embedding rate is more than 1 bit per pixels. However, this might lead to alteration of the degree of randomness of pixels, thus making it vulnerable to statistical attacks when high degree of embedding is involved.

2.2 Transform Domain Techniques

2.2.1 JSTEG algorithm

In JSteg algorithm, the LSBs' of the Discrete Courier Transform Coefficients of JPEG images are replaced. Before the embedding process begins, the image is converted to the DCT domain in 8x8 blocks such that the values of ci switch from pixel values to DCT coefficients. For values to be presented as whole numbers, each 8x8 block is quantised according to a Quantisation Table Q. The result is where the embedding algorithm operates. The hiding mechanism skips all coefficients with values of 0 or 1. JSteg is resistant to visual attacks and has high capacity with a compression ratio of 12%. JSteg is restricted for visual attacks and is less immune for statistical attacks.

Table 2.1 Example of 8x8 SubBlock of DCT Coefficients

257	6	3	0	0	0	-6	7
8	0	1	-5	2	4	-4	3
-5	-1	-1	1	-1	-1	2	-2
2	2	2	2	-1	-2	0	0
-1	-2	0	-2	2	1	-1	0
2	-1	-2	0	-2	1	2	1
-2	0	3	2	2	-3	-1	-1
2	0	-2	-2	-1	2	0	1

2.2.2 F5 algorithm

F5 withstands statistical and visual attacks. It also offers a large embedding capacity. F5 has an improved embedding efficiency because it uses matrix encoding. F5 uses per mutative straddling to evenly distribute the changes over entire steganogram.

F5 algorithm avoids the security problem when the data is embedded into the JPEG images (Andreas Westfeld, 2001). It embeds the data into chosen Discrete Courier Transform coefficients which are chosen randomly. Changes made to the length of the message are reduced by using matrix embedding. As it doesn't replace any bits, it can go undetected in chi-square attacks. It can withstand statistical and visual attacks. Its embedding capacity is greater than 13%. This algorithm supports GIF, BMP, TIFF, JPEG formats.

3. Evaluation of parameters

Setting up specific evaluation parameters helps in developing newer algorithms & improving performance of existing algorithms.

- *Level of Security*

Security depends upon how susceptible the system is to steganalysis. A steganographic technique is undetectable or secure if no statistical tests can point out the difference between the cover & stego-image.

- *Capacity*

It is the amount of information that can be effectively hidden inside a cover image. Embedding rate, in relative measurement called data embedding rate, is given in bits per pixel/bpp, or bits per non-zero DCT coefficients/ bpnc, etc.

- *Imperceptibility or fidelity*

Stego-images are expected to be inconspicuous. Under the same level of security and capacity, higher the fidelity of the stego-image, better the imperceptibility.

- *Domain of Embedding*

It is important in determining the overall performance of steganographic algorithms. Spatial domain techniques offer higher capacity but are vulnerable to statistical steganalysis. Transform domain techniques are more resistant to statistical steganalysis.

- *Type of Images Supported*

Images are available in many formats like GIF, BMP, JPEG. It is important to understand which types of images are suitable for which steganographic algorithms. It is more difficult to conceal information inside images having lossy compression as compared to images having lossless compression.

- *Time Complexity*

It is important for determining the practical application of the algorithm for embedding into large images and also their execution systems having lesser resources like mobile phones.

Thus the possible improvements that might be adopted for future systems are:

- Increasing embedding efficiency
- Decreasing distortion after embedding
- Choosing alternate colour spaces

Table 4.1: Parameter Based Comparison (Ratnakirti Roy, Suvamoy Changder, Anirban Sarkar, Narayan C Debnath, 2013)

Domain	Algorithm	Security Level	Embedding Capacity	Fidelity	Image support	Complexity
Spatial	Direct LSB	Low	1-3 bpp	High	Lossless	Low
	PIT	Medium	>1 bpp	High*	Lossless	Low
	OPAP	Medium	1 bpp	High	Lossless(GS)	Medium
	PVD	Medium	>1 bpp	High$	Lossless(GS)	Medium
	SLSB	Medium	1-3 bpp	High*	Lossless	Medium
Transform	Jsteg	Medium	<1 bpnc	High	Lossy/Lossless	Medium
	F5	Very high	0.8 bpnc	High	Lossy/Lossless	High

**-Till capacity <3 bpp*; $-*Till capacity < 4bpp;* GS-*Grayscale*; bpsc- *Bits per singular value coefficient*; *#-Till compression ≤ 50%*

3.1 Proposed parameters

- *Invisibility*

The steganographic algorithm should be well hidden. An undetectable algorithm increases strength of the technique.

- *Capacity of payload*

Steganography involves hiding information bits inside cover objects like images. Information hiding capacity of the image and the algorithm should he high.

- *Resistance to statistical attacks*

Steganalysis can easily detect signatures left by the algorithm, thereby compromising security. The algorithm should be robust.

- *Robustness against image manipulation*

Cropping, editing the images can result in loss of information. The algorithm should be designed in such a way that it can withstand changes in the images.

- *Support for all file types*

The steganographic algorithm should be applicable to any type of file. Using a single type of file can arouse suspicion.

- *Inconspicuous files*

The images used should be inconspicuous. Large files may alert the attacker, of possible hidden information. If possible, information should be segmented, and sent through multiple images.

Table 4.3 Comparison of Image Steganography Algorithm

	LSB in BMP	LSB in GIF	JPEG	Spread Spectrum
Invisibility	High	Medium	High	High
Capacity of payload	High	Medium	Medium	Medium
Resistance statistical attacks	Low	Low	Medium	High
Robustness against image manipulation	Low	Low	Medium	Medium
Support for all file formats	Low	Low	Low	High
Inconspicuous files	Low	Low	High	High

Conclusions

We have thus evaluated the various steganographic algorithms on basis of their method of implementation as well as other parameters. This comparison reveals that the security level, robustness of the transform domain techniques are higher than spatial domain techniques. However, spatial techniques offer more capacity for embedding, have low time complexity and work well on systems with low resources. In order to decide which steganographic algorithm to use, we need to consider type of application and if we are willing to trade some features for better security.

References

Ratnakirti Roy, Suvamoy Changder, Anirban Sarkar, Narayan C Debnath, 2013, *Evaluating Image steganography Techniques* IEEE Journal 978-1-4673-2088-7/13/2013.

James C. Judge, 2001, Steganography- Past, Present, Future SANS Institute.

R.Amirtharajan, Sandeep Kumar Behera, Motamarri Abhilash Swarup, Mohamed Ashfaaq K and John Bosco Balaguru Rayappan, 2010, *Colour Guided Colour Image Steganography* Universal Journal of Computer Science and Engineering Technology 1.

Adnan Abdul-Aziz Gutub, 2010, *Pixel Indicator Technique for RGB Image Steganography* Journal of Emerging Technologies in Web Intelligence, Vol. 2, No. 1.

Kanzariya Nitin K. Nimavat Ashish V., 2013, *Comparison of Various Images Steganography Techniques* International Journal of Computer Science and Management Research Vol 2 Issue 1.

Andreas Westfeld, 2001, *F5-A Steganographic Algorithm High Capacity Despite Better Steganalysis* pp. 289–302.

R. Amritharajan, R. Akila, P. Deepikachowdavarapu, 2010, *A Comparative Analysis of Image Steganography*, International Journal of Computer Applications, Vol. 2, No.3, pp. 41-47.