Analysis of Different Steganographic Algorithms for Secured Data Hiding

Dr.M.Umamaheswari[1] **Prof.S.Sivasubramanian**[2] **S.Pandiarajan**[3]

Department of Computer Science and Engineering,
Bharath University, Chennai-73, Tamil Nadu, India.

Abstract: The goal of steganography is to avoid drawing suspicion to the transmission of a hidden message. If suspicion is raised, steganography may fail. The success of steganography depends on the secrecy of the action. If steganography is detected, the system will fail but data security depends on the robustness of the applied algorithm. In this paper, we compress the secret message and encrypt it by the receiver's public key along with the stego key and embed both messages in a carrier using an embedding algorithm. The stego - image is the result we get by running the algorithm you select on the message (file to hide) and cover (image). It can be saved into BMP or PNG format. The reason that it can only be saved in these formats is because they are lossless - there is no information lost as part of the file formatting. The various applications of steganography include secure military communications, multimedia watermarking and fingerprinting applications for authentication purposed to curb the problem of digital piracy.

Key words*:* Steganography, Cryptography, Data Hiding, Steganographic algorithms.

1. INTRODUCTION

The word "*Steganography*" is of Greek origin and means *"covered or hidden writing"*. The main aim in steganography is to hide the very existence of the message in the cover medium. Steganography and cryptography are counter parts in digital security the obvious advantage of steganography over cryptography is that messages do not attract attention to themselves, to messengers, or to recipients. Also, the last decade has seen an exponential growth in the use of multimedia data over the Internet. These include Digital Images, Audio and Video files. This rise of digital content on the internet has further accelerated the research effort devoted to steganography.

The initial aim of this study was to investigate steganography and how it is implemented. Based on this work a number of common methods of steganography could then be implemented and evaluated. The strengths and weaknesses of the chosen methods can then be analysed. To provide a common frame of reference all of the steganography methods implemented and analysed used GIF images.

To make a steganographic communication even more secure the message can be compressed and encrypted before being hidden in the carrier. Cryptography and steganography can be used together[3]. If compressed the message will take up far less space in the carrier and will minimise the information to be sent. The random looking message which would result from encryption and compression would also be easier to hide than a message with a high degree of regularity. Therefore encryption and compression are recommended in conjunction with steganography

Steganography refers to the science of "invisible" communication. Unlike cryptography, where the goal is to secure communications from an eavesdropper, steganographic techniques strive to hide the very presence of the message itself from an observer. The general idea of hiding some information in digital content has a wider class of applications that go beyond steganography, Fig. 1.

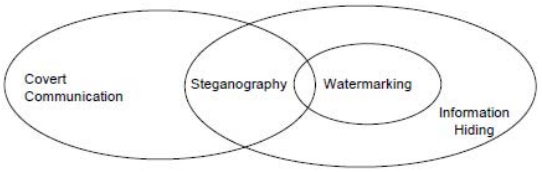

Fig. 1 Relationship of steganography to related fields

In order to defeat the steganalytic attacks,algorithms have been proposed which try to restore the statistics which get distorted during the embedding procedure and which may be used for steganalysis.

Manuscript received August 5, 2010
Manuscript revised August 20, 2010

The techniques involved in such applications are collectively referred to as *information hiding*. For example, an image printed on a document could be annotated by metadata that could lead a user to its high resolution version.

In general, metadata provides additional information about an image. Although metadata can also be stored in the file header of a digital image, this approach has many limitations. Usually, when a file is transformed to another format (e.g., from TIFF to JPEG or to BMP), the metadata is lost. Similarly, cropping or any other form of image manipulation destroys the metadata[5]. Finally, metadata can only be attached to an image as long as the image exists in the digital form and is lost once the image is printed. Information hiding allows the metadata to travel with the image regardless of the file format and image state (digital or analog).A special case of information hiding is *digital watermarking*.

Digital watermarking is the process of embedding information into digital multimedia content such that the information (the watermark) can later be extracted or detected for a variety of purposes including copy prevention and control. Digital watermarking has become an active and important area of research, and development and commercialization of watermarking techniques is being deemed essential to help address some of the challenges faced by the rapid proliferation of digital content.

The key difference between information hiding and watermarking is the absence of an active adversary. In watermarking applications like copyright protection and authentication, there is an active adversary that would attempt to remove, invalidate or forge watermarks. In information hiding there is no such active adversary as there is no value associated with the act of removing the information hidden in the content. Nevertheless, information hiding techniques need to be robust against accidental distortions.

Unlike information hiding and digital watermarking, the main goal of steganography is to communicate securely in a completely undetectable manner.

Steganography provides a means of secret communication which cannot be removed without significantly altering the data in which it is embedded. The embedded data will be confidential unless an attacker can find a way to detect it.

	Confidentiality	Integrity	Unremovability
Encryption	Yes	No	Yes
Digital Signatures	No	Yes	No
Steganography	Yes/No	Yes/No	Yes

Table.1 Comparison of secret communication techniques

The modern formulation of steganography is often given in terms of the *prisoner's problem* [11] where Alice and Bob are two inmates who wish to communicate in order to hatch an escape plan. However, all communication between them is examined by the warden, Wendy, who will put them in solitary confinement at the slightest suspicion of covert communication. Specifically, in the general model for steganography, illustrated in Fig. 2, we have Alice wishing to send a secret message *m* to Bob. In order to do so, she "embeds" *m* into a *cover-object c*, and obtains a *stego-object s*. The stego-object *s* is then sent through the public channel. Thus we have the following definitions:

Cover-object: refers to the object used as the carrier to embed messages into. Many different objects have been employed to embed messages into for example images, audio, and video as well as file structures, and html pages to name a few.

Stego-object: refers to the object which is carrying a hidden message. So given a cover object, and a messages the goal of the steganographer is to produce a stego object which would carry the message.

In a *pure steganography* framework, the technique for embedding the message is unknown to Wendy and shared as a secret between Alice and Bob. However, it is generally considered that the algorithm in use is not secret but only the key used by the algorithm is kept as a secret between the two parties, this assumption is also known as Kerchoff's principle in the field of cryptography. The secret key, for example, can be a password used to seed a pseudo-random number generator to select pixel locations in an image cover-object for embedding the secret message (possibly encrypted)[11]. Wendy has no knowledge about the secret key that Alice and Bob share, although she is aware of the algorithm that they could be employing for embedding messages.

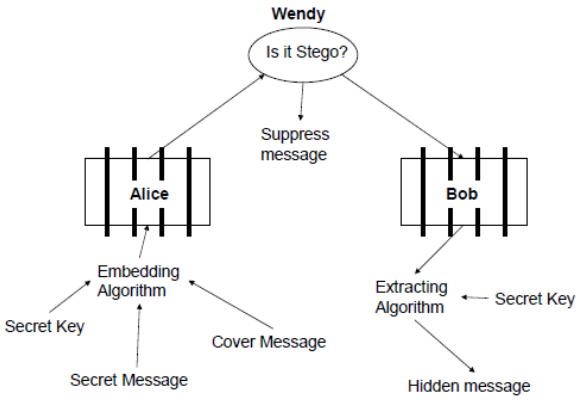

Fig. 2 General model for steganography

2. TYPES OF STEGANOGRAPHY

Steganography can be split into two types, these are Fragile and Robust. The following section describes the definition of these two different types of steganography.

Fragile steganography involves embedding information into a file which is destroyed if the file is modified. This method is unsuitable for recording the copyright holder of the file since it can be so easily removed, but is useful in situations where it is important to prove that the file has not been tampered with, such as using a file as evidence in a court of law, since any tampering would have removed the watermark. Fragile steganography techniques tend to be easier to implement than robust methods.

Robust marking aims to embed information into a file which cannot easily be destroyed. Although no mark is truly indestructible, a system can be considered robust if the amount of changes required to remove the mark would render the file useless. Therefore the mark should be hidden in a part of the file where its removal would be easily perceived.

There are two main types of robust marking. Fingerprinting involves hiding a unique identifier for the customer who originally acquired the file and therefore is allowed to use it. Should the file be found in the possession of somebody else, the copyright owner can use the fingerprint to identify which customer violated the license agreement by distributing a copy of the file.

Unlike fingerprints, watermarks identify the copyright owner of the file, not the customer. Whereas fingerprints are used to identify people who violate the license agreement watermarks help with prosecuting those who have an illegal copy. Ideally fingerprinting should be used but for mass production of CDs, DVDs, etc it is not feasible to give each disk a separate fingerprint.

Watermarks are typically hidden to prevent their detection and removal, they are said to be imperceptible watermarks. However this need not always be the case. Visible watermarks can be used and often take the form of a visual pattern overlaid on an image. The use of visible watermarks is similar to the use of watermarks in non-digital formats (such as the watermark on British money).

2.1 ALGORITHMS USED IN STEGANOGRAPHY

There are four algorithms currently implemented, each use least significant bit steganography and some filter the image first.

2.1.1 BLINDHIDE

This is the simplest way to hide information in an image. It *blind*ly *hide*s because it just starts at the top left corner of the image and works it's way across the image (then down - in scan lines) pixel by pixel. As it goes along it changes the least significant bits of the pixel colours to match the message. To decode the process the least significant bits starting at the top left are read off. This is not very secure - it's really easy to read off the least significant bits. It also isn't very smart - if the message doesn't completely fill up the possible space then just the top part of the image is degraded but the bottom is left unchanged - making it easy to tell what's been changed.

Algorithm Pixel Swap

- *Randomly select 2 pixels x1 and x2 from the cover image using a pseudo–random sequence.*
- *If the two pixels lie within a specified distance α (α=2 or 3 generally), they are suitable for embedding, otherwise generate another set of pixels.*
- *Pick up the message bit. If the message bit is zero (or one), check if x1 > x2 otherwise swap x1 and x2. Do the reverse operation for the message bit one (zero)*
- *For decoding, select the pixels using the same pseudo-random sequence. Check if the 2 pixels are within the pre-specified range α. If x1>x2, the message bit is zero (one) otherwise the message bit is one (zero).*

This scheme preserves the first order statistic (histogram) inherently without applying separate restoration process. This scheme also does not add any visual distortion to the image since the threshold used for swapping of pixels is kept considerably small ($\alpha <= 5$) which only affects the least significant bit planes of an image. To measure the distortion introduced by the embedding in the cover image, the Peak Signal to Noise Ratio (PSNR) after embedding was observed for one hundred images.

2.1.2 HIDE SEEK

This algorithm randomly distributes the message across the image. It is named after "*Hide* and *Seek*" - a Windows 95 steganography tool that uses a similar technique. It uses a password to generate a random seed, then uses this seed to pick the first position to hide in. It continues to randomly generate positions until it has finished hiding the message. It's a little bit smarter about how it hides because you have to try every combination of pixels in every order to try and "crack" the algorithm - unless you have the password. It's still not the best method because it is not looking at the pixels it is hiding in - it might be more useful to figure out areas of the image where it is better to hide in.

2.1.3 FILTER FIRST

This algorithm filters the image using one of the inbuilt

filters and then hides in the highest *filter* values *first*. It is essentially a fancier version of BlindHide as it doesn't require a password to retrieve the message. Because we are changing the pixels we need to be careful about filtering the picture because we don't want to use information for filtering that might change. If we do, then it may be difficult (if not impossible) to retrieve the message again. So this algorithm filters the most significant bits, and leaves the least significant bits to be changed. It is less noticeable on an image because using the filter ensures we are hiding in the parts of the image that are the least noticeable.

2.1.4 BATTLE STEG

The best of all. This algorithm performs "*Battle*ship *Steg*anography". It first filters the image then uses the highest filter values as "ships". The algorithm then randomly "shoots" at the image (like in HideSeek) and when it finds a "ship" it clusters it's shots around that hit in the hope of "sinking" the "ship". After a while it moves away to look for other ships. The effect this has is that the message is randomly hidden, but often hidden in the "best" parts to hide in thanks to the ships. It moves away to look for other ships so that we don't degrade an area of an image too greatly. It is secure because you need a password to retrieve the message. It is fairly effective because it is hiding (if you set the values right) the majority of the information in the best areas.

2.1.5 DYNAMIC BATTLESTEG AND FILTERFIRST

These two algorithms do the same as BattleSteg and FilterFirst, except they use dynamic programming to make the hiding process faster and less memory intensive. They are NOT compatible with the original algorithms because the order of pixels kept in the dynamic array is not exactly the same.

2.2 IMAGE TECHNIQUES

2.2.1 LEAST SIGNIFICANT BIT

LSB – Least Significant Bit Hiding (Image Hiding).
This method is probably the easiest way of hiding information in an image and yet it is surprisingly effective. It works by using the least significant bits of each pixel in one image to hide the most significant bits of another[8]. So in a JPEG image for example, the following steps would need to be taken

1. First load up both the host image and the image you need to hide.
2. Next chose the number of bits you wish to hide the secret image in. The more bits used in the host image, the more it deteriorates. Increasing the number of bits used though obviously has a beneficial reaction on the secret image increasing its clarity.
3. Now you have to create a new image by combining the pixels from both images. If you decide for example, to use 4 bits to hide the secret image, there will be four bits left for the host image. (PGM - one byte per pixel, JPEG - one byte each for red, green, blue and one byte for alpha channel in some image types)

Host Pixel: 1011<u>0001</u>
Secret Pixel: 0011<u>1111</u>
New Image Pixel: **10110011**

4. To get the original image back you just need to know how many bits were used to store the secret image. You then scan through the host image, pick out the least significant bits according the number used and then use them to create a new image with one change - the bits extracted now become the most significant bits.

Host Pixel: 1011<u>0011</u>
Bits used: 4
New Image: **00110000**

Hiding depends on the settings you choose - but as an example if we hide in the 2 least significant bits then, we can hide:

```
MaxBytes = (image.height() * image.width() * 3 * 2) /8
```

i.e. the number of pixels, times the number of colours (3), times the number of bits to hide in, all divided by 8 to get the number of bytes. It helps to hide a bit less than this because the algorithms may take a while to find places that haven't had anything hidden in it when we are close to the threshold.

2.3 TYPES OF FILTERS

There are 2 different filters - the Laplace filter and the Sobel filter. Traditionally these filters are used for detecting edges in pictures. As a side effect they happen to pick the "best areas" to change. This is because an edge has a really light pixel next to a darker one. If we make the lighter pixel darker and the darker pixel lighter we aren't going to notice as much as if we make two pixels the same

color different. The Sobel filter is better at detecting edges, but the Laplace filter is better at picking up noise.
The embedding rate bar shown on the encoding and simulation panels shows the percentage of space available for steganography that will be written to. Anything less than 10% is a good rate. You can decrease the embedding rate by using a smaller message, a larger image or by changing the number of bits that will be written to (in the algorithm options). For steganalysis, the embedding rate is an approximation of the percent of space available that has been written to. However, the steganalysis techniques only use the very least significant bit in each color for their calculation.

3. STEGANALYSIS

For hiding information, an equal number of clever techniques have been designed to detect the hidden information [3]. These techniques are collectively known as 'steganalysis'. As introduced earlier, the Laplace formula is one such steganalytic method. Attacks on steganography can involve detection and/or destruction of the embedded message. A stego-only attack is when only the stego-image is available to be analysed . A known cover attack is when the original cover image is also available. It involves comparing the original cover image with the stego-image. As explained above hiding information results in alterations to the properties of a carrier which may result in some sort of degradation to the carrier. Original images and stego-images can be analysed by looking at colour composition, luminance and pixel relationships and unusual characteristics can be detected. If a hidden message is revealed at some later date the attacker could analyse the stego-image for future attacks. This is called known message attack. The chosen stego attack is used when the steganography algorithm and the image are known. A chosen message attack is when the stegoanalyst generates stego-images using a given steganography algorithm using a known message. The purpose is to examine the patterns produced in the stego-images that may point to the use of certain steganography algorithms. Most steganographic algorithms embed messages by replacing carefully selected pixels bits with message bits. Any changes to the data associated with the image through embedding will change the properties of the image in some way. This process may create patterns or unusual exaggerated noise
Two other popular techniques are RS Analysis and Sample Pairs Analysis.RS Analysis makes small modifications to the least significant bit plane in an image then uses these modifications and a discrimination function to classify groups of pixels. The counts of the groups based on the modifications allow the calculation of an estimated embedding rate. Images that do not contain steganography often have a natural embedding rate of up to 3%, whereas images containing hidden information usually have estimated embedding rates which accurately reflects the amount of hidden information.
RS Analysis is a special case of Sample Pairs Analysis, which also uses least significant bit modifications to help calculate an estimated embedding rate. Sample Pairs Analysis utilises finite state machines to classify groups of pixels modified by a given pattern. Both steganalysis techniques are very accurate at predicting the embedding rate on stego-images using least
significant bit embedding. Since the two proposed techniques, FilterFirst and BattleSteg, both use least significant bit embedding, we can use RS Analysis and Sample Pairs Analysis to compare them against more traditional techniques such as BlindHide and HideSeek.

4.0 NEED FOR DATA HIDING

- Covert communication using images (secret message is hidden in a carrier image)
- Ownership of digital images, authentication, copyright
- Data integrity, fraud detection, self-correcting images
- Traitor-tracing (fingerprinting video-tapes)
- Adding captions to images, additional information, such as subtitles, to video, embedding subtitles or audio tracks to video (video-in-video)
- Intelligent browsers, automatic copyright information, viewing a movie in a given rated version
- Copy control (secondary protection for DVD)

5. PLAUSIBLE DENIABILITY

Plausible deniability is defined as: "encryption scheme is deniable if the sender can generate plausible keys and random choices that will satisfy the authority and at the same time keep the past communication private."
In this paper, we propose a novel plausible deniability scheme in steganography by using a diversionary message and encrypt it with a DES-based algorithm. Then, we compress the secret message and encrypt it by the receiver's public key along with the stego key and embed both messages in a carrier using an embedding algorithm. It will be demonstrated how this method can support plausible deniability and is robust against steganalysis.

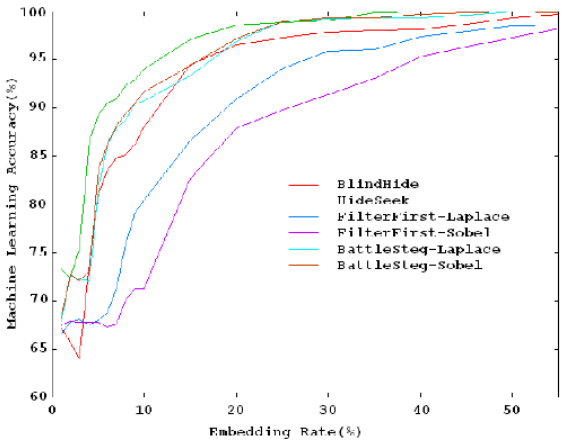

Fig.3 Graph of Embedding Rate versus Machine Learning Accuracy

For implementation of the proposed method the following five steps are considered:

Step 1: Construct cipher text as cover medium. Let m be the plain text, E encryption algorithm, K decryption key and m_c cipher text, then:

$$E_K(m) = m_c$$

Step 2: Embed secret message in the cipher text. Let SE be the steganography algorithm, K_s the stego key, M the secret message, SG the stego object, then:

$$SE_{Ks}(M, m_c) = SG$$

Step 3: Uncover secret message from stego object. Let SD be the algorithm for recovering the secret message using the same stego key, then:

$$SD_{Ks}(SG) = M$$

Step 4: Deny secret communication – Reveal encryption key K to uncover the cover medium plain text to deny information hiding. Thus:

$$D_K(SG) = \hat{m}$$

Step 5: Verification – Encryption of the resulting text in Step 4 must give the stego object. That is:

$$E_K(\hat{m}) = \hat{m}_c$$

The condition for plausible deniability is:

$$\hat{m}_c = SG$$

6. CONCLUSION

Success in steganographic secrecy results from selecting the proper mechanisms. However, a stego medium which seems innocent enough may, upon further investigation, actually broadcast the existence of embedded information. Development in the area of covert communications and steganography will continue. Research in building more robust methods that can survive image manipulation and attacks continues to grow. The more information is placed in the public's reach on the Internet, the more owners of such information need to protect themselves from theft and false representation. Systems to recover seemingly destroyed information and steganalysis techniques will be useful to law enforcement authorities in computer forensics and digital traffic analysis. The future enhancements can be done for location based hiding, more number of filters can be added and can use same stego image with different filters.

ACKNOWLEDGEMENTS

The authors would like to thank the Director, Secretary, Correspondent, Principal, HOD of Bharath University, Chennai for their motivation and constant encouragement. The authors would like to thank the Faculty Members of Department of Computer Science and Engineering for critical review of this manuscript and for his valuable input and fruitful discussions. Also, he takes privilege in extending gratitude to his family members and friends who rendered their support throughout this research work.

REFERENCES

[1] S. Katzenbeisser and F..A. P. Petitcolas, Eds., Information HidingTechniques for Steganography and Digital Watermarking. Norwell, MA: Artech House, 2000.

[2] I. J. Cox, M. L. Miller, and J. A. Bloom, Digital Watermarking. San Mateo, CA: Morgan Kaufmann, 2002.

[3] M. Wu and B. Liu, Multimedia Data Hiding. New York: Springer-Verlag, 2003.

[4] C.S. Lu, Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property. Hershey, PA: Idea Group Publishing, 2004.

[5] B. Furht and D. Kirovski, Multimedia Security Handbook, Part III and IV. Boca Raton, FL: CRC, 2005.

[6] N. F. Johnson and S. Jajodia, "Steganalysis: The investigation of hidden information," in IEEE Information Technology Conference, New York, Sep. 1998.

[7] N. F. Johnson and S. Jajodia, "Steganalysis of images created using current steganography software," Lecture Notes in Computer Science,vol. 1525, 1998.

[8] Digital-Invisible-Ink Data Hiding Based on Spread-Spectrum and Quantization Techniques,Chun-Hsiang Huang, Shang-Chih Chuang, and Ja-Ling Wu, Fellow, IEEE,IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 10, NO. 4, JUNE 2008.

[9] J. Fridrich, M. Goljan and T. Holotyak "New Blind Steganalysis and its Implications", with, in Proc. SPIE Electronic Imaging, Photonics West, January 2006
[10] P. Moulin and Y. Wang, "New results on steganography," Proc. of CISS, 2004.

[11] G. Simmons, "The prisoners problem and the subliminal channel" *CRYPTO*,pp. 51-67, 1983.

Dr.M.Umamaheswari, M.C.A., M.Phil. M.Tech., Ph.d. works as the professor & Head of department of CSE of Bharath University, Chennai, Tamil Nadu. She has more than 12 years of teaching and research experience and her areas of specialization are Computer Networks, Networks Security and Data Mining.

Mr.S.Sivasubramanian, M.Sc., M.Tech. pursuing Ph.d. works as an Asst.professor from Department of CSE of Bharath University, Chennai, Tamil Nadu. He has more than 8 years of teaching and research experience and his areas of specialization are mobile computing, Database Management System, Computer Networks, Networks Security and Data Mining.

S.Pandiarajan received his Bachelor of Science in Chemistry from Madurai Kamaraj University, Diploma In Computer Software from APTECH Institute, Master of Science in Computer Science and Information Technology from Madurai Kamaraj University and pursuing Master of Technology in Computer Science and engineering from Bharath University Chennai.