Evaluation of Steganography Proficiencies & Taxonomies in Transformation Domain

K.Tulasi Krishna Kumar[1], G.Nagappa[2], P. Rameswara Anand[3]
Head: Training & Placements[1], Associate Professor[2], Senior Lecturer[3]
Department of Computer Science & Engineering

Abstract – In this paper we have explained in brief on steganography by explaining what it is, proficiencies and taxonomies in transforming a domain by providing a brief history with illustrations of some methods for implementing steganography. Though the forms are many, in this paper we focused on the use of images in steganography. In section II we have explained on various taxonomies and the enactments. In section III to understand the steganography algorithms we concentrated on the effigies on JPEG file format. Section IV & V concentrates on prophecies and attacks. In section VI we have shown experimental outcomes with a real time experiment on effigy in a cover effigy.

Index terms – cryptography, steganography, robustness, marked documents, suspicious files and mosaic attack

1. INTRODUCTION

Steganography was gleaned from the Greek words ‘stegos’, that means a roof or covered and ‘graphia’, which means writing is the art and science of hiding the fact that communication is taking place. To give a more formal definition, the Merriam Webster Dictionary defines steganography as: “The Art or practice of concealing a file, effigy or a message.” As we all know, Osama Bin Laden and Al-Qaeda and many other ISI organization used steganography to send messages through websites and news groups. However, until now no substantial evidence supporting this claim has been found, so either al-Qaeda has used or created real good steganographic algorithms or the claim is probably false. Many different motives exist to detect the use of steganography, so proficiencies to do so continue to be developed while the hiding algorithms become more advanced.

2. TAXONOMY & ENACTMENTS

2.1: Taxonomy

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.

2.1.1: Fragile

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 proficiencies tend to be easier to implement than robust methods.

2.1.2: Robust

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. There are two main types of robust marking: Fingerprinting and Water marking.

2.2: Enactments

There are ways to hide information in an effigy, audio and even text files. Almost all digital file formats can be used for steganography, but the formats that are more suitable are those with a high degree of redundancy. Redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object’s use and display.

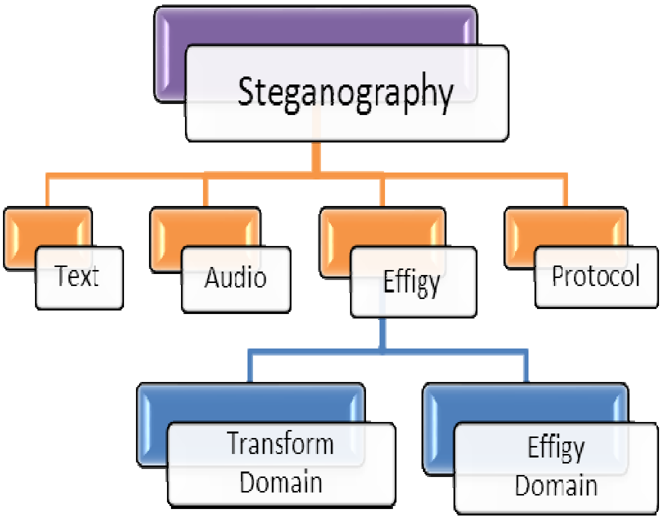


Figure 1: Enactments of steganography

The redundant bits of an object are those bits that can be altered without the alteration being detected easily. The below figure shows the four main categories of file formats that can be used for steganography. Hiding information in text is historically the most important method of steganography. An obvious method was to hide a secret message in every nth letter of every word of a text message. It is only since the beginning of the Internet and all the different digital file formats that is has decreased in importance. Text steganography using digital files is not used very often since text files have a very small amount of redundant data. Given the proliferation of digital effigies, especially on the Internet, and given the large amount of redundant bits present in the digital representation of an effigy, effigies are the most popular cover objects for steganography. This paper will focus on hiding information in effigies in the next sections. To hide information in audio files similar proficiencies are used as for effigy files. One different proficiency unique to audio steganography is masking, which exploits the properties of the human ear to hide information unnoticeably. A faint, but audible, sound becomes inaudible in the presence of another louder audible sound. This property creates a channel in which to hide information.Although nearly equal to effigies in steganography potential, the larger size of meaningful audio files makes them less popular to use than effigies. The term protocol steganography refers to the proficiency of embedding information within messages and network control protocols used in network transmission. In the layers of the OSI network model there exist covert channels where steganography can be used. An example of where information can be hidden is in the header of a TCP/IP packet in some fields that are either optional or are never used. Hiding information into a media requires following elements

- The cover media(x) that will hold the hidden data
- The secret message (y), may be plain text, cipher text or any type of data
- The stego function (Ke) and its inverse (Ke^{-1})
- An optional stego-key (z) or password may be used to hide and unhide the message.

The stego function operates over cover media and the message (to be hidden) along with a stego - key (optionally) to produce a stego media (S). Steganography and Cryptography are great partners in spite of functional difference. It is common practice to use cryptography with steganography. The schematic of steganography operation is shown below.

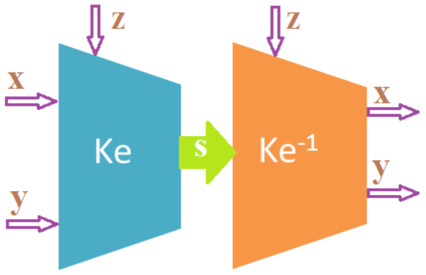


Figure 2: The Steganographic operation

2.2.1: Text Proficiencies

Hiding information is to conceal it in what seems to be inconspicuous text. It is more difficult when it comes to electronic versions of text. Copies are identical and it is impossible to tell if it is an original or a copied version. To embed information inside a document we can simply alter some of its characteristics. These can be either the text formatting or characteristics of the characters. The key to this problem is that we alter the document in a way that it is simply not visible to the human eye yet it is possible to decode it by computer. Figure shows the general principle in embedding hidden information inside a document.

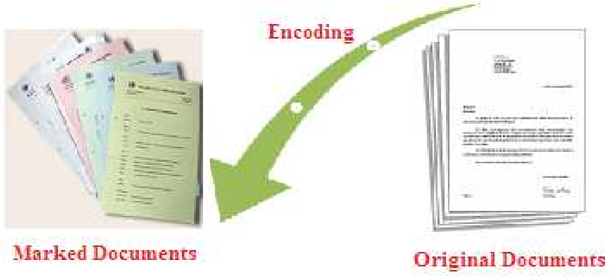


Figure 3: Document embedding process

Again, there is an encoder and to decode it, there will be a decoder. The codebook is a set of rules that tells the encoder which parts of the document it needs to change. It is also worth pointing out that the marked documents can be either identical or different. By different, we mean that the same watermark is marked on the document but different characteristics of each of the documents are changed. Some of the text steganography proficiencies are discussed below

2.2.2: Line Shift Coding Protocol

In line shift coding, we simply shift various lines inside the document up or down by a small fraction (such as 1/300th of an inch) according to the codebook. The shifted lines are undetectable by humans because it is only a small fraction but is detectable when the computer measures the distances between each of the lines. Differential encoding proficiencies are normally used in this protocol, meaning if you shift a line the adjacent lines are not moved. These lines will become a control so that the computer can measure the distances between them. By finding out whether a line has been shifted up or down we can represent a single bit, 0 or 1, and if we put the whole document together, we can embed a number of bits and therefore have the ability to hide large information.

2.2.3: Feature Coding Protocol

In feature coding, there is a slight difference with the above protocol, and this is that the document is passed through a parser where it examines the document and it automatically builds a codebook specific to that document. It will pick out all the features that it thinks it can use to hide information and each of these will be marked into the document. This can use a number of different characteristics such as the height of certain characters, the dots above i and j and the horizontal line length of letters such as f and t. Line shifting and word shifting proficiencies can also be used to increase the amount of data that can be hidden.

2.2.4: White Space Manipulation

One way of hiding data in text is to use white space. If done correctly, white space can be manipulated so that bits can be stored. This is done by adding a certain amount of white space to the end of lines. The amount of white space corresponds to a certain bit value. Due to the fact that in practically all text editors, extra white space at the end of lines is skipped over, it won't be noticed by the casual viewer. In a large piece of text, this can outcome in enough room to hide a few lines of text or some secret codes.

2.2.5 Effigy proficiencies

Effigies are the most popular cover objects used for steganography. In the domain of digital effigies many different effigy file formats exist, most of them for specific applications. For these different effigy file formats, different steganographic algorithms exist.

3. TRANSFORM DOMAIN

To understand the steganography algorithms that can be used when embedding data in the transform domain, one must first explain the type of file format connected with this domain. The JPEG file format is the most popular effigy file format on the Internet, because of the small size of the effigies.

3.1: JPEG syncopate

To compress an effigy into JPEG format, the RGB tinge representation is first converted to a YUV representation. In this representation the Y component corresponds to the luminance (or brightness) and the U and V components stand for chrominance (or tinge). According to research the human eye is more sensitive to changes in the brightness (luminance) of a pixel than to changes in its tinge. This fact is exploited by the JPEG syncopate by down sampling the tinge data to reduce the size of the file. The tinge components (U and V) are halved in horizontal and vertical directions, thus decreasing the file size by a factor of 2. The next step is the actual transformation of the effigy. For JPEG, the Discrete Cosine Transform (DCT) is used, but similar transforms are for example the Discrete Fourier Transform (DFT).

These mathematical transforms convert the pixels in such a way as to give the effect of "spreading" the location of the pixel values over part of the effigy. The DCT transforms a signal from an effigy representation into a frequency representation, by grouping the pixels into 8 × 8 pixel blocks and transforming the pixel blocks into 64 DCT coefficients each. A modification of a single DCT coefficient will affect all 64 effigy pixels in that block. The next step is the quantization phase of syncopate. Here another biological property of the human eye is exploited: The human eye is fairly good at spotting small differences in brightness over a relatively large area, but not so good as to distinguish between different strengths in high frequency brightness. This means that the strength of higher frequencies can be diminished, without changing the appearance of the effigy. JPEG does this by dividing all the values in a block by a quantization coefficient. The outcomes are rounded to integer values and the coefficients are encoded using Huffman coding to further reduce the size.

3.2: JPEG steganography

Originally it was thought that steganography would not be possible to use with JPEG effigies, since they use forfeiture syncopate which outcomes in parts of the effigy data being altered. One of the major characteristics of steganography is the fact that information is hidden in the redundant bits of an object and since redundant bit left out when using JPEG it was feared that the hidden message would be destroyed. Even if one could somehow keep the message intact it would be difficult to embed the message without the changes being noticeable because of the harsh syncopate applied. However, properties of the syncopate algorithm have been exploited in order to develop a steganographic algorithm for JPEGs. One of these properties of JPEG is exploited to make the changes to the effigy invisible to the human eye. During the DCT transformation phase of the syncopate algorithm, rounding errors occur in the coefficient data that are not noticeable. Although this property is what classifies the algorithm as being forfeiture, this property can also be used to hide messages. It is neither feasible nor possible to embed information in an effigy that uses forfeiture syncopate, since the syncopate would destroy all information in the process.

Thus it is important to recognize that the JPEG syncopate algorithm is actually divided into forfeiture and lossless stages. The DCT and the quantization phase form part of the forfeiture stage, while the Huffman encoding used to further compress the data is lossless. Steganography can take place between these two stages. Using the same principles of LEAST SIGNIFICANT BIT insertion the message can be embedded into the least significant bits of the coefficients before applying the Huffman encoding. By embedding the information at this stage, in the transform domain, it is extremely difficult to detect, since it is not in the visual domain.

4. EVALUATION OF DIFFERENT PROFICIENCIES

All the above mentioned algorithms for effigy steganography have different strong and weak points and it is important to ensure that one uses the most suitable algorithm for an application. All steganographic algorithms have to comply with a few basic requirements. The most important requirement is that a steganographic algorithm has to be imperceptible. We propose a set of criteria to further define the imperceptibility of an algorithm. These requirements are as follows:

4.1 : 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 eye. The moment that one can see that an effigy has been tampered with, the algorithm is compromised

4.2 : 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.

4.3 : Robustness against statistical attacks

Statistical steganalysis is the practice of detecting hidden information through applying statistical tests on effigy data. Many steganographic algorithms leave a 'signature' when embedding information that can be easily detected through statistical analysis. To be able to pass by a warden without being detected, a steganographic algorithm must not leave such a mark in the effigy as be statistically significant.

4.4 : Robustness against effigy manipulation

In the communication of a stego effigy by trusted systems, the effigy may undergo changes by an active warden in an attempt to remove hidden information. Effigy manipulation, such as cropping or rotating, can be performed on the effigy before it reaches its destination. Depending on the manner in which the message is embedded, these manipulations may destroy the hidden message. It is preferable for steganographic algorithms to be robust against either malicious or unintentional changes to the effigy.

4.5 : Independent of file format

With many different effigy file formats used on the Internet, it might seem suspicious that only one type of file format is continuously communicated between two parties. The most powerful steganographic algorithms thus possess the ability to embed information in any type of file. This also solves the problem of not always being able to find a suitable effigy at the right moment, in the right format to use as a cover effigy.

4.6 : Unsuspicious files

This requirement includes all characteristics of a steganographic algorithm that may outcome in effigies that are not used normally and may cause suspicion. Abnormal file size, for example, is one property of an effigy that can outcome in further investigation of the effigy by a warden. The comparison between least significant bit (LEAST SIGNIFICANT BIT) insertion in BMP and in GIF files, JPEG syncopate steganography, the patchwork approach and spread spectrum proficiencies as discussed below according to the above requirements.

4.1.A: Invisibility

Least significant bit in BMP	:	*High*
Least significant bit in GIF	:	*Medium*
JPEG syncopate	:	*High*
Patch work	:	*High*
Spread spectrum	:	*High*

4.2.A: Payload capacity

Least significant bit in BMP	:	*High*
Least significant bit in GIF	:	*Medium*
JPEG syncopate	:	*Medium*
Patch work	:	*Low*
Spread spectrum	:	*Medium*

4.3.A: Robustness against statistical attacks

Least significant bit in BMP	:	*Low*
Least significant bit in GIF	:	*Low*
JPEG syncopate	:	*Medium*
Patch work	:	*High*
Spread spectrum	:	*High*

4.4. A: Robustness against effigy manipulation

Least significant bit in BMP	:	*Low*
Least significant bit in GIF	:	*Low*
JPEG syncopate	:	*Medium*
Patch work	:	*High*
Spread spectrum	:	*Medium*

4.5.A: Independent of file format

Least significant bit in BMP	:	*Low*
Least significant bit in GIF	:	*Low*
JPEG syncopate	:	*Medium*
Patch work	:	*High*
Spread spectrum	:	*Medium*

4.6.A: Unsuspicious files

Least significant bit in BMP	:	*Low*
Least significant bit in GIF	:	*Low*
JPEG syncopate	:	*Medium*
Patch work	:	*High*
Least significant bit in BMP	:	*Low*
Least significant bit in GIF	:	*Low*
Spread spectrum	:	*Medium*

The levels at which the algorithms satisfy the requirements are defined as high, medium and low. A high level means that the algorithm completely satisfies the requirement, while a low level indicates that the algorithm has a weakness in this requirement. A medium level indicates that the requirement depends on outside influences, for example the cover effigy used. Least significant bit in GIF effigies has the potential of hiding a large message, but only when the most suitable cover effigy has been chosen. Unfortunately in the algorithms that are evaluated here, there is not one algorithm that satisfies all of the requirements. Thus a trade-off will exist in most cases, depending on which requirements are more important for the specific application.

5. ATTACKS

Information hiding proficiencies still suffer from several limitations leaving them open to attack and robustness criteria vary between different proficiencies. Attacks can be broadly categorized although some attacks will fit into multiple categories but the attackers always finds new ways in finding the bugs and makes some other methods to perform the attack. Broadly speaking they can be classified into five types.

5.1: Basic Attacks

Basic attacks take advantage of limitations in the design of the embedding proficiencies. Simple spread spectrum proficiencies, for example, are able to survive amplitude distortion and noise addition but are vulnerable to timing errors. Synchronisation of the chip signal is required inorder for the proficiency to work so adjusting the synchronisation can cause the embedded data to be lost. It is possible to alter the length of a piece of audio without changing the pitch and this can also be an effective attack on audio files.

5.2: Robustness Attacks

Robustness attacks attempt to diminish or remove the presence of a watermark. Although most proficiencies can survive a variety of transformations, syncopate, noise addition, etc they do not cope so easily with combinations of them or with random geometric distortions. If a series of minor distortions are applied the watermark can be lost while the effigy remains largely unchanged. What changes have been made will likely be acceptable to pirates who do not usually require high quality copies. Since robustness attacks involve the use of common manipulations, they need not always be malicious but could just be the outcomeof normal usage by licensed users. Protecting against these attacks can be done by anticipating which transformations pirates are likely to use. Embedding multiple copies of the mark using inverse transformations can increase the resistance to these attacks. The below Figure shows the outcomes of Stir Mark applied to effigy (a) in effigy (c). The distortion here is almost unnoticeable and is easier to see when the same distortions are applied to grid (c) to give (d).

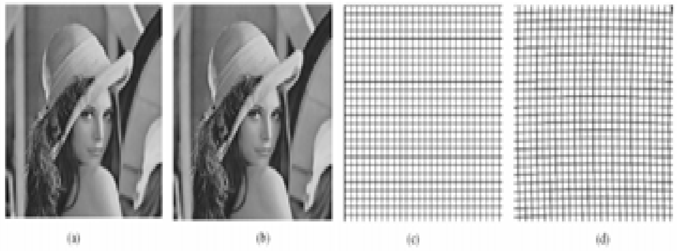


Figure 3: Outcomes of StirMark

The echo hiding proficiency encodes zeros and ones by adding echo signals distinguished by different values for their delay and amplitude to an audio signal. Decoding can be done by detecting the initial delay using the auto-correlation of the cepstrum of the encoded signal but this proficiency can also be used as an attack. If the echo can be detected then it can be removed by inverting the formula used to add it. The difficult part is detecting the echo without any knowledge of the original or the echo parameters. This problem is known as ‘blind echo cancellation’. Finding the echo can be done using a proficiency called cepstrum analysis. Other attacks will attempt to identify the watermark and then remove it. This proficiency is particularly applicable if the marking process leaves clues that help the attacker gain information about the mark.

5.3: Presentation Attacks

Presentation attacks modify the content of the file in order to prevent the detection of the watermark. The mosaic attack takes advantage of size requirements for embedding a watermark. In order for the marked file to be the same size as the original the file must have some minimumsize to accommodate the mark. By splitting the marked file into small sections the mark detection can be confused. Many web browsers will draw effigies together with no visible split enabling the full effigy to be effectively restored while hiding the mark. If the minimum size for embedding the mark is small enough the mosaic attack is not practical. This attack can defeat web crawlers which download pictures from the Internet and check them for the presence of a client’s watermark. In this example an effigy had a simple watermark embedded in it using Digimarc included in Jasc Paint Shop Pro. The effigy was then separated into 16 tiles, each of which was then checked for the presence of the watermark. Tiles are shown separated here for clarity and those surrounded by the red border no longer contain the watermark. However this does show how small the tiles need to be in order to lose all watermark information as 6 tiles still contain the watermark at this size. If the tiles are made small enough, the watermark could be lost.

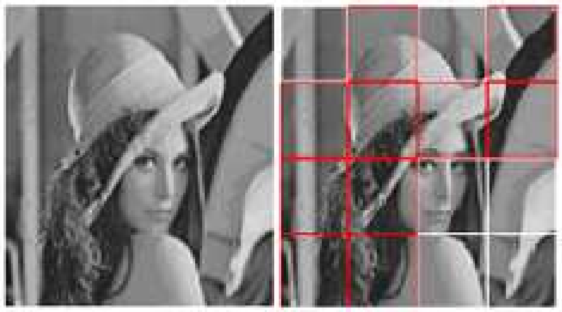


Figure 4: The mosaic attack

5.4: Interpretation Attacks

Interpretation attacks involve finding a situation in which the assertion of ownership is prevented. Robustness is usually used to refer to the ability of the mark to survive transformations and not resistance to an algorithmic attack. Therefore the definition of robustness may not be sufficient. One interpretation attack takes advantage of mark detection being unable to tell which mark came first if multiple marks are found. If the owner publishes a document, d + w (where d isthe original and w is the watermark) a pirate can add a second watermark w’ and claim that the document is his and that the original was d + w - w’. Though it is clear that at least one party has a counterfeit copy, it is not clear which one. This would seem to suggest the need to use other proficiencies to identify the original owner of a file.

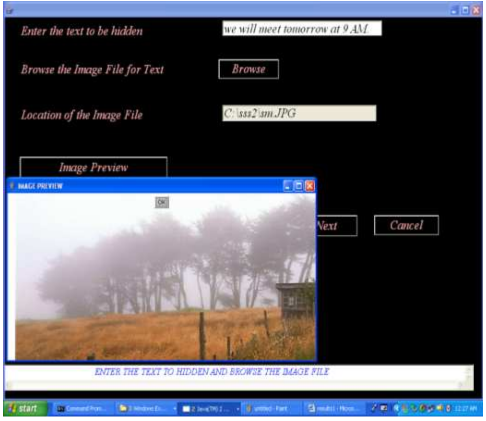
5.5: Enactment Attacks

As with other areas in computer security the enactment of a marking system can provide more opportunities for attack than the marking proficiency itself. If the mark detection software is vulnerable it may be possible for attackers to deceive it. Digimarc, one of the most widely used picture marking schemes was attacked using a weakness in the enactment. Users register an ID and password with the marking service. A debugger was used to break into the software which checks these passwords and disable the checking. The attacker can change the ID and this will change the mark of already marked effigies. The debugger also allowed bypassing of checks to see if a mark already existed and therefore allowed marks to be overwritten. There is a general attack on mark readers which explores an effigy on the boundary between no mark having been found and one being detected. An acceptable copy of the effigy can be iteratively generated which does not include the mark. Clearly the software used to implement steganographic proficiencies needs to be secure and ideas from other areas of computer security can be used to ensure this.

6. EXPERIMENTAL OUTCOME

6.1: Text in effigy

Fig.1 represents how the text is hidden inside the effigy. Figure 5 represents the transmitted effigy and figure 6 represents the decryption at the receiver. The private text to be hidden is: “we will meet tomorrow at 9 AM”. The cover effigy

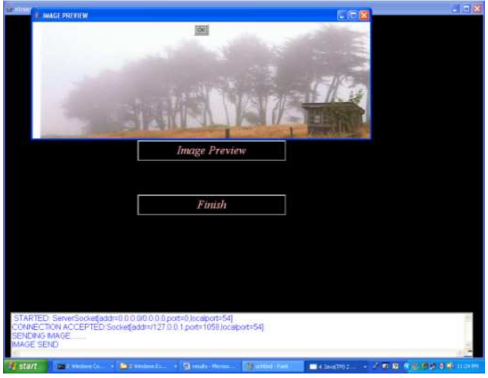


Enter the text to be hidden: we will meet tomorrow at 9 AM

Location of the Image File: C:\sss2\sm.JPG

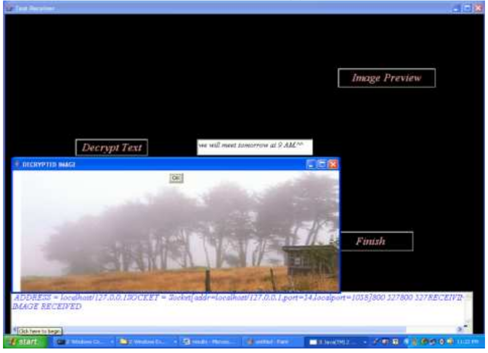
ENTER THE TEXT TO HIDDEN AND BROWSE THE IMAGE FILE

Figure 5: hiding the text in the effigy



```
STARTED: ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=54]
CONNECTION ACCEPTED:Socket[addr=/127.0.0.1,port=1058,localport=54]
SENDING IMAGE........
IMAGE SEND
```

Figure 6: Transmitted Effigy (Text is hidden)



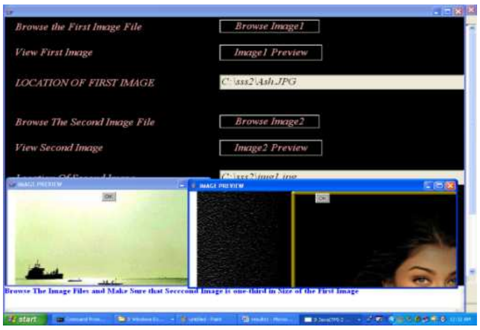
we will meet tomorrow at 9 AM^^

```
ADDRESS = localhost/127.0.0.1SOCKET = Socket[addr=localhost/127.0.0.1,port=54,localport=1058]800 327800 327RECEIVE
IMAGE RECEIVED
```

Figure 7: Decryption at the receiver

6.2: Effigy in effigy:

Fig.4 represents how the private effigy is hidden inside the cover effigy. Above figure 6 represents the transmitted total effigy & figure 7 represents the decryption at the receiver.

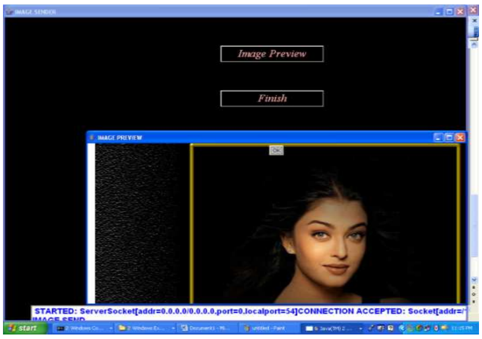


LOCATION OF FIRST IMAGE: C:\sss2\Ash.JPG

Location Of Second Image: C:\sss2\img1.jpg

Browse The Image Files and Make Sure that Seccond Image is one-third in Size of the First Image

Figure 8: hiding the effigy inside the cover effigy



```
STARTED: ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=54]CONNECTION ACCEPTED: Socket[addr=/
IMAGE SEND
```

Figure 9: Transmitted total Effigy

7. CESSATION

Cryptanalysis is a dynamic tool with a long history and the capability to adapt to new levels of technology. It has its own place in computer data security. By the amount of free and commercial tools available today, one can deduce that the use of Cryptanalysis is growing. Steganography is just another tool for someone to use to hide data, and We believe it will be used more often in the future, whether for covert communication or personal data concealment. Security professionals will surely need to be aware of its existence as its use becomes more prevalent. Hiding a message with Cryptanalysis methods reduces the chance of a message being detected. In and of itself, steganography is not a good solution to secrecy, but neither is simple substitution and short block permutation for encryption. But if these methods are combined, you have much stronger encryption routines. Like any tool, Cryptanalysis is neither inherently good nor evil, it is the manner in which it is used which will determine whether it is a benefit or detriment to our society.

REFERENCE

[1] Ethical Hacking 2k13 – A National Workshop on Ethical Hacking by Sai Satish in Yogananda Institute of Technology & Science, Tirupathi.

[2] A paper on 'Cryptanalysis of the multilinear map over the integers' on springer 2015 by JH Cheon, K Han, C Lee, H Ryu, D

[3] A book on ‘Steganography & Steganalysis’ - J.R.Krenn.

[4] A paper on 'Cryptanalysis of iterated Even-Mansour schemes with two keys-2014' by I Dinur, O Dunkelman, N Keller, A Shamir - Advances in Cryptology.

[5] A Detailed Look at Steganagraphic Proficiencies and their Use in an Open-System Environment in http://www. sans.org/rr/white papers/covert/677.php

[6] An article on 'Methods involving maps, imagery, video and steganography' by TF Rodriguez, TJ Brundage, SM Shovoly

[7] A paper on 'Steganography forensics method for detecting least significant bit replacement attack' by X Wang, C Wei, X Han

[8] Book on ‘An introduction to steganography’, steganalysis – by authors micohlosli, clint white side, chrisshultz.

[9] A book on 'Trends in steganography' by the authors E Zielińska, W Mazurczyk, K Szczypiorski

[10] Journal of Cryptology, 2015 in Springer on 'Cryptanalysis of SHA-0 and reduced SHA-1' by E Biham, R Chen, A Joux

[11] You tube video on 'Challenging the doctrines of JPEG steganography' by V Holub, J Fridrich

[12] Petitocolos,FabienA.P-“Information hiding:Proficiencies for Steganography and Digital watermarking”, 2000.

[13] An international on 'Comparison of different techniques for Steganography in images' in ijaiem.org by MFM Shelke, MAA Dongre, MPD Soni

[14] A reference ‘http://en.wikipedia.org/wiki/ Steganography’

[15] Role of Material Science in Engineering & Medical – An International Workshop by IWMEM, Krishna Teja Technical Campus.

[16] Computer Forensics, Cybercrime and Steganography Resources http://www.forensics.nl/steganography

[17] Chandra mouli.R ,kharrazi.M and Menon.N “Effigy steganography and Steganalysis : Concepts and practise” Proceedings of international workshop on digital water marking October 2003.

AUTHORS PROFILE

Mr. K. Tulasi Krishna Kumar: is serving as Head: Training & Placements received his Masters in CSE from VTA & also a fellow member in IACSIT, CSTA, IRED, IAENG. He is an affiliate in Pro-E, C.N.C certified by CITD - Government of India. He published various international journals in data mining, software project management & information security.

Mr. G. Nagappa: is an Associate Professor at YITS (India). He perceived his M. Tech in Software Engineering in 2008 and ever since he is in teaching line taking classes to U.G & P.G students. He research focuses on data mining, Design Patterns & published various international journals.

Mr. P.Rameswara Anand: is currently, working as a Senior Lecturer in Jigjiga University – Ethiopia. He did his MCA from S.V. University in 1993 & M.Tech in 2012 ever since he is in the teaching line taking classes to B. Tech and MCA students of various institutions. In the year 2012, he got qualified in APSET Conducted by Osmania - Hyderabad.