Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 5, May 2013, pg.249 – 254

RESEARCH ARTICLE

SECURE COMMUNICATION OF SECRET DATA USING STEGNOGRAPHY

Sobia Habib[1], Atiya Parveen[2], Saoud Sarwar[3]

[1]M. Tech Scholar, Al-Falah School of Engg and Tech., Dhauj, Faridabad, India

[2]M. Tech Scholar, Al-Falah School of Engg and Tech., Dhauj, Faridabad, India

[3]HOD (Computer Science and Engg), Al-Falah School of Engg and Tech., Dhauj, Faridabad, India

[1] habib.sobia85@gmail.com; [2] atiya.parveen@gmail.com; [3] saoud.hod.cse@gmail.com

Abstract— The paper discusses and analyze about a method which provides a secure mechanism for the transfer of messages by the process of Steganography. The solution uses password protection to prevent unauthorized access of the files. The tool makes use of any file type which acts as a cover to the message/file. The message/file is hidden behind the cover using LSB flipping algorithm. The tool can accept all types of media files (including JPEG images). After the steganography is performed the user can also compare the original image with the image in which the data is hidden using the compare operation of the tool. The compare tool compares the two images using PSNR (peak to signal ratio) and thus shows the amount of distortion in the image. The retrieval of the message and the audio can also take place through this tool. The tool has a very appealing and an easy to use GUI thus making it an effective solution to hide messages/files.

Key Terms: - Stegnography; LSB flipping; PSNR; encryption; decryption

I. INTRODUCTION

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. The word which is steganography is of Greek origin and means "concealed writing" from the Greek words *steganos* meaning "covered or protected", and *graphei*(γραφή) meaning "writing". The first recorded use of the term was in 1499 by Johannes Trithemius in his *Steganographia*, a treatise on cryptography and steganography disguised as a book on magic. Generally, messages will appear to be something else: images, articles, shopping lists, or some other *covertext* and, classically, the hidden message may be in invisible ink between the visible lines of a private letter. It is high security technique for long data transmission.

The advantage of steganography over cryptography alone is that messages do not attract attention to themselves. Plainly visible encrypted messages—no matter how unbreakable—will arouse suspicion, and may in themselves be incriminating in countries where encryption is illegal. Therefore, whereas cryptography protects the contents of a message, steganography can be said to protect both messages and communicating parties.

Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. As a simple example, a sender might start with an innocuous image file and adjust the color of every 100th pixel to correspond to a letter in the alphabet, a change so subtle that someone not specifically looking for it is unlikely to notice it.

The project titled 'Image and Audio Steganograph' provides means for secure data transmission. Hereby, important files carrying confidential information can be stored in the server in an encrypted form. Access to these files is limited to certain authorized people only. Transmission also takes place in an encrypted form so that no intruder can get any useful information from the original file during transit. Along with Image steganography, File steganography can also be performed using this tool that accepts any file as cover and allows embedding of any file into it.

Encryption is the process of encoding a message in such a way as to hide its contents. Modern Cryptography includes several secure algorithms for encrypting and decrypting messages. They are all based on the use of secrets called ***keys.*** A cryptographic key is a parameter used in an encryption algorithm in such a way that the encryption cannot be reversed without the knowledge of the key.

The larger the cover message is (in data content terms — number of bits) relative to the hidden message, the easier it is to hide the latter. For this reason, digital pictures (which contain large amounts of data) are used to hide messages on the Internet and on other communication media. It is not clear how commonly this is actually done. For example: a 24-bit bitmap will have 8 bits representing each of the three color values (red, green, and blue) at each pixel. If we consider just the blue there will be 2^8 different values of blue. The difference between say 11111111 and 11111110 in the value for blue intensity is likely to be undetectable by the human eye. Therefore, the least significant bit can be used (more or less undetectably) for something else other than color information. If we do it with the green and the red as well we can get one letter of ASCII text for every three pixels.

Stated somewhat more formally, the objective for making steganographic encoding difficult to detect is to ensure that the changes to the carrier (the original signal) due to the inject the payload (the signal to covertly embed) are visually (and ideally, statistically) negligible; that is to say, the changes are indistinguishable from the noise floor of the carrier.

The Tool also provides a comparison operation which compares the original image and the image in which the message is hidden. This is done by PSNR.

Peak Signal-to-Noise Ratio, often abbreviated **PSNR**, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

PSNR is most commonly used to measure the quality of reconstruction of lossy compression codecs (e.g., for image compression). The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs, PSNR is an *approximation* to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality, in some cases the reverse may be true. One has to be extremely careful with the range of validity of this metric; it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content.

II. LITERATURE REVIEW

We have developed a project for steganography. The project works by embedding the text message in images and embedding audio file in another file. The main highlight of our project is that we are embedding text message in jpeg images. Because of the earlier compression in the jpeg images we are not able to embed the message in the jpeg file format as there is absence of redundant bit in the image. But we finally have achieved our goal by LSB flipping method.

We have added all other file format of images which are generally used as – bmp , tif , gif , tif , png . The same has been done for the Audio file format as we have added mp3, wav , ram , wma .

We have added an extra module for comparing the input image which does not contain any data and the output image which has the text embedded in it . This has been done via **'PEAK SIGNAL TO NOISE RATIO'.**

In our research so far we haven't found any tool which has all these modules integrated in one project.

2.1 JPEG IMAGE STEGANOGRAPHY

One of the research papers named "JOURNAL OF ADVANCES IN INFORMATION TECHNOLOGY, VOL. L NO.3. AUGUST 2010" stated the research propose a framework for hiding large volumes of data in images while incurring minimum perceptual degradation. The data is recovered successfully, without any errors, after operations such as decompression, additive noise, and image tampering. The proposed methods can be employed for applications that require high-volume embedding with robustness against certain non-malicious

attacks. The hiding methods we propose are guided by the growing literature on the information theory of data hiding

The key novelty of our approach is that our coding framework permits the use of local criteria to decide where to embed data. In order to robustly hide large volumes of data in images without causing significant perceptual degradation, hiding techniques must adapt to local characteristics within an image. The main ingredients of our embedding methodology are as follows.

(a) As is well accepted, data embedding is done in the transform domain, with a set of transform coefficients in the low and mid frequency bauds selected as possible candidates for embedding. (These are preserved better under compression attacks than high frequency coefficients)

(b) A novel feature of our method is that, from the candidate set of transform coefficients, the encoder employs local criteria to select which subset of coefficients it will actually embed data in. In example images, the use of local criteria for deciding where to embed is found to be crucial to maintaining image quality under high volume embedding.

(c) For each of the selected coefficients, the data to be embedded indexes the choice of a scalar quantizer for that coefficient. We motivate this by information theoretic analysis.

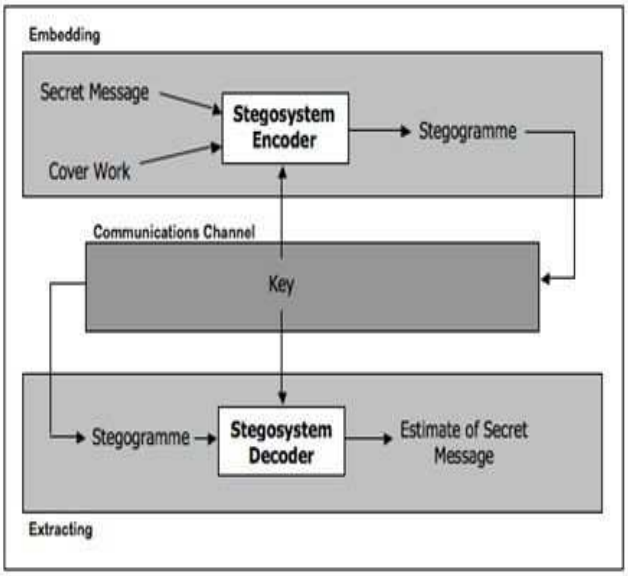
(d) The decoder does not have explicit knowledge of the locations where data is hidden, but employs the same criteria as the encoder to guess these locations. The distortion due to attacks may now lead to insertion errors (the decoder guessing that a coefficient has embedded data, when it actually does not) and deletion errors In principle, this can lead to desynchronization of the encoder and decoder.

(c) An elegant solution based on erasures and errors correcting codes is provided to the synchronization problem caused by the use of local criteria.

Specifically, we use a code on the hidden data that spans the entire set of candidate embedding coefficients, and that can correct both errors and erasures. The subset of these coefficients in which the encoder does not embed can be treated as erasures at the encoder. Insertions now become errors. and deletions become erasures (in addition to the erasures already guessed correctly by the decoder, using the same local criteria as the encoder).

While the primary purpose of the code is to solve the synchronization problem, it also provides robustness to errors due to attacks.

Coding Framework: The coding framework employs the idea of erasures at the encoder. The bit stream to be hidden is coded, using a low rate code, assuming that all host coefficients that meet the global criteria will actually be employed for hiding. A code symbol is erased at the encoder if the local perceptual criterion for the block or coefficient is not met. Since, we code over entire space of coefficients that lie in a designated low-frequency band.



Coding Framework

Code words can be constructed to achieve vel) good correction ability. A maximum distance separable (MDS) code such Rood Solomon (RS) code, doe" not incur any penalty for erasures at the encoder. Turbo-like codes, which operate vel)' close to capacity, incur only a minor overhead due to erasures at the encoder. Figure 3.4 shows how the sequence is decoded in the presence of attacks. As it is seen, insertions become errors, and deletions become additional erasures

It should be noted that a deletion, which causes an erasure, is about half as costly as an insertion, which causes an error.

We noted that use of image-adaptive criteria is necessary when hiding large volumes of data into images. A threshold is used to determine whether to embed in a block (ET scheme) or in a coefficient (SEC scheme). More advanced image-adaptive schemes would exploit the human visual system (HVS) models to determine where to embed information. Distortion due to attack may cause an insertion (decoder guessing that there is hidden data where there is no data) or a deletion (decoder guessing that there is no data where there was data hidden). There could also be decoding error, where the decoder makes a mistake in correctly decoding the bit embedded. While the decoding errors can be countered using simple error correction codes, insertions and deletions can potentially cause catastrophic loss of synchronization between encoder and decoder.

In the ET scheme, insertions and deletions are observed when the attack quality factor is mismatched with the design quality factor for JPEG attack. However, for the SEC scheme, there are no insertions or deletions for most of the images for JPEG attacks with quantization interval smaller than or equal to the design interval. This is because no hidden coefficient with magnitude $\leq$ t can be ambiguously decoded to t+1 due to JPEG quantization with an interval smaller than the design one. Both the ET and SEC schemes have insertions/deletions under other attacks.

The coding framework employs the idea of erasures at the encoder. The bit stream to be hidden is coded, using a low rate code, assuming that all host coefficients that meet the global criteria will actually be employed for hiding. A code symbol is erased at the encoder if the local perceptual criterion for the block or coefficient is not met. Since we code over entire space of coefficients that lie in a designated low-frequency band, long codewords can be constructed to achieve very good correction ability. A maximum distance separable (MDS) code [24], such as Reed Solomon (RS) code, does not incur any penalty for erasures at the encoder. Turbo-like codes, which operate very close to capacity, incur only a minor overhead due to erasures at the encoder. Figure 3.4 shows how the sequence is decoded in the presence of attacks. As it is seen, insertions become errors, and deletions become additional erasures. It should be noted that a deletion, which causes an erasure, is about half as costly as an insertion, which causes an error. Hence, it is desirable that the data-hiding scheme [4] be adjusted in such a manner that there are only a few insertions. Thus, using a good erasures and errors correcting code, one can deal with insertions/deletions without a significant decline in original embedding rate. Reed-Solomon codes have been used for ET scheme and Repeat Accumulate codes have been used for the SEC scheme.

2.2 AUDIO STEGANOGRAPHY

Moving forward to audio steganography a research paper by R SRIDEVI, DR. A DAMODARAM, DR. SVL.NARASIMHAM Assoc. Prof., Department of Computer Science and Engineering, JNTUCEH, Hyderabad Prof., Department of Computer Science and Engineering, JNTUCEH, Hyderabad Prof., School of Information Technology, JNTUH, Hyderabad

.PROPOSED SYSTEM: Enhanced Audio Steganography is a method of hiding the message in the audio file of any formats. EAS provides an easy way of implementation of mechanisms when compared with audio steganography. Apart from the encoding and decoding in Audio steganography, EAS contain extra layers of encryption and decryption. The four layers in EAS are:

1. Encoding 2) Decoding 3)Encryption 4)Decryption

Powerful encryption algorithm is used to encrypt the message before encoding for further security purpose. The following steps is used to

a. Encrypt the message

b. Adding all ASCII values of characters in the key given by user.

c. Converting the sum into bit pattern

d. Performing logical operation to the bit pattern.

Encoding:

The audio file contains set of bytes. For e.g. take an audio file which play for 10 secs. It has more than 60,000 bytes. Each byte is received and checked if the received byte is 254 or 255.If it is byte 255 or 254, encoding is done. So for one character to encode we need eight 254 or 255 bytes. One character is hidden in consecutive eight 254 or 255 bytes.In order to mark the end of message, the LSB bit of next eight consecutive 254 or 255 bytes which comes after all the messages have encoded are replaced by 1.Before encoding, message is encrypted using public key.

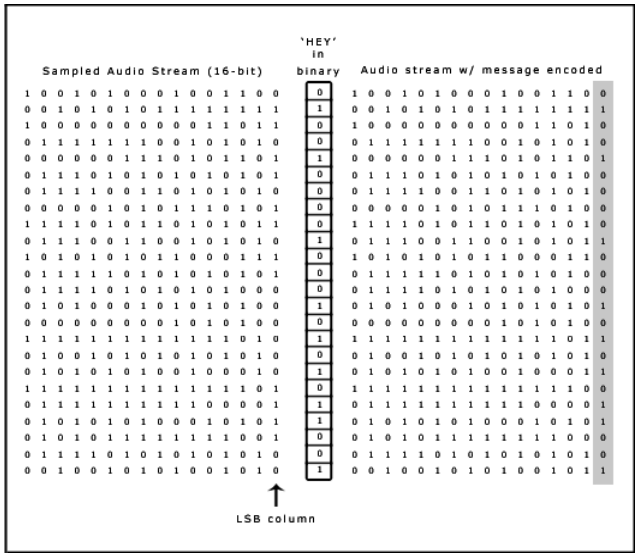
Decoding

The encoded file is decoded to get the message .The message is decoded first and then decrypted by the public key. The eight consecutive 254 or 255 bytes are taken and decrypted with the public key. This decrypted byte have value less than 128.So if the value is 255 after decrypted then it is said to be end of message.

Encryption:

The user allowed entering the public key/shared key in any combination of numbers, symbols and characters. The key contains set of characters. All characters are converted to ASCII value and add all the ASCII value to

get single number. And that single number is converted to bit pattern and by simple logical operation (XOR) you can get a single number less than 128.It is a new private key .It is added to the characters one by one in the message, before encoding .A



2.3 LSB FLIPPING

A research paper by Giacomo Cancelli, Gwenael Doerr, Ingemar J. Cox and Mauro Barni says that we have a cover image with M×N pixels and with pixel values from the set P. For example, for an 8-bit grayscale image, P = {0, …, 255}. The stego-detection method starts with dividing the image into disjoint groups of n adjacent pixels (x1, …, xn). For example, we can choose groups of n=4 consecutive pixels in a row. We define so called discrimination function f that assigns a real number f(x1, …, xn)∈R to each pixel group G = (x1, …, xn). The purpose of the discrimination function is to quantify the smoothness or "regularity" of the group of pixels G. The noisier the group of pixels G=(x1, …, xn) is, the larger the value of the discrimination function becomes. For example, we can choose the 'variation' of the group of pixels (x1, …, xn) as the discrimination function f: ∑−== +−111 2 1(, ,...,)nin i if x x x x x . (1)

Finally, we define an invertible operation F on P called "flipping". Flipping is a permutation of gray levels that entirely consists of two-cycles. Thus, F 2 = Identity or F(F(x)) = x for all x∈P. The permutation F1: 0 ↔ 1, 2 ↔ 3, …, 254 ↔ 255 corresponds to flipping (negating) the LSB of each gray level. We further define so called shifted LSB flipping F−1 as −1 ↔ 0, 1 ↔ 2, 3 ↔ 4, …, 253 ↔ 254, 255 ↔ 256, or F−1(x) = F1(x+1)−1 for all x. (2)

For completeness, we also define F0 as the identity permutation F(x)=x for all x∈P. We use the discrimination function f and the flipping operation F to define three types of pixel groups: R, S, and U Regular groups: G ∈ R ⇔ f(F(G)) > f(G)

Singular groups: G ∈ S ⇔ f(F(G)) < f(G)

Unusable groups: G ∈ U ⇔ f(F(G)) = f(G),

where F(G)= (F(x1), …, F(xn)). We may wish to apply different flipping to different pixels in the group G. The assignment of flipping to pixels can be captured with a mask M, which is a ntuple with values −1, 0, and 1. The flipped group FM(G) is defined as (FM(1)(x1), FM(2)(x2), …, FM(n)(xn)). Our stego-detection technique is based on analyzing how the number of regular and singular groups changes with the increased message length embedded in the LSB plane. This stego-detection technique is a result of pure serendipity stemming from our research on lossless data embedding (for more details, see the journal version of this paper.

2.4 PSNR (PEAK TO SIGNAL NOISE RATIO)

Rate-distortion analysis based optimization of image compression has been actively studied for many years, which is targeting at maximizing the coding performance under given bit rate constraint. Most of these works have been limited to gray level images. Very few works on the optimal bit allocation between luma and chroma components in color image compression were reported. It has been investigated the bit allocation amongst luma and chroma components in different color spaces in the context of subband/VQ compression of color images, their research was focused on the optimal perceptual weighting at different sub bands of either luma or chroma component. As for the issue of optimal sub-sampling in image coding, to our knowledge, the only analytical work was done by Alfred M.Bruchstein etc. [6]. In their work, an analytical model was first established to

explain the better transform coding of images with down-scaling at low bit rate. Subsequently, a simple algorithm was developed to derive the optimal down-scaling factor of level image at given bit rate. Apparently, their work was still limited to the compression of gray level images. Furthermore, their method only provides the optimal down-scaling factor, and could not accurately predict the actual rate-distortion performance.

III. TECHNIQUE USED

Steganographic Technique used is concealing data within encrypted data. The data to be concealed is first encrypted before being used to overwrite part of a much larger block of encrypted data. This technique works most effectively where the decrypted version of data being overwritten has no special meaning or use. Each pixel typically has three numbers associated with it, one each for red, green, and blue intensities, and these values often range from 0-255. Each number is stored as eight bits (zeros and ones), with a one worth 128 in the most significant bit (on the left), then 64, 32, 16, 8, 4, 2, and a one in the least significant bit (on the right) worth just 1.

IV. CONCLUSIONS

Existing solutions for steganography do not provide Jpeg steganography in a public sense. The tools are paid.

Steganography for all file types is provided. The tool can accept any type of file as cover.

Comparison can be performed between the two images using the compare tool.

The project after being tested was found to achieve its objectives in an effective manner. The system is found to be 100% error free and ready for implementation.

Video files are too big to use as a cover.

Changing the type of the output file produces unreadable format in the output file.

Password protection adds another layer of security to the tool.

Except Images, Comparison of any other file type can't be done.

V. FUTURE PROSPECTS

.Video Steganography can be added to the tool. Video files can be used as a cover to hide.

Comparison between audio and video files can be added.

The development of a system that will utilize the Steganographic Obliterator on incoming and email messages and attachments..

Feedback facility can be added so that user suggestions can be incorporated into the tool.

Windows or Web Services can be designed to further automate the operation.

REFERENCES

[1] Hilbert Schildt(2002) Java 2: The Complete Reference Fifth Edition
[2] William Stallings(2003) Cryptography and network security principle and practice.
[3] www.jjtc.com/steganography
[4] en.wikipedia.org/wiki/steganography
[5] doc.oracle.com
[6] "IMAGE-BASED JPEG STEGANOGRAPHY" Mat´uˇs J´okay — Tom´aˇs Moravˇc´ık[2010].
[7] Image Steganography by Paul Hammes.
[8] JPEG Compression Steganography by Meena Kumari

