Development of Multi-Level System of Steganography

Tunde Joseph Ogundele
Department of Computer Science
Federal University of Technology Akure, Nigeria
and
Adebayo Olusola Adetunmbi
Department of Computer Science
Federal University of Technology Akure, Nigeria

ABSTRACT

Internet world is characterized by many users among which are crackers and thieves. Hence, the need for a secured system to safely exchange confidential information among users across the web is required. Of such tool is steganography that simply hides the user information under other kind of information such as image so that no one suspects that a sensitive data is being transferred. This paper presents a steganography scheme with an improved capacity and enhanced security by compressing the information before embedding it under an image. This is done by encoding the message before embedding it in the blue object of the cover image components (pixels). To prove this scheme, several testing are performed and results are compared.

Keyword: Steganography; Cryptography; Encoding; Image; Encryption; Decryption; Compression

1. INTRODUCTION

The Internet has revolutionized the modern world and the numerous Internet based applications that get introduced these days add to the high levels of comfort and connectivity in every aspects of human life have increased. It is estimated that, over 2.095 billion people worldwide use Internet for various purposes [1]. This ranges from accessing information for educational needs to financial transactions and procurement of goods and services. It is therefore imperative to have robust security measurements to safeguard the privacy and security of the underlying data [9].

Cryptography techniques [10][9] have been widely used to encrypt the plaintext data, with the ciphertext not really making much sense when interpreted as it is. This can naturally raise the curiosity level of a malicious hacker or intruder to conduct cryptanalysis attacks on the ciphertext. The needs arise for more prudent secured way to send the secret information, either in plaintext or ciphertext, by cleverly embedding it as part of a cover media such as an image. This idea forms the basis for Steganography. Steganography is the art and science [7] of communicating in such a way that the presence of the message is not detected. In this sense, messages or information sent will not attract suspicion to themselves, to messengers or to recipients [13]14].

In this paper, we propose a model that improves the security and hidden capacity of existing steganographic design and come up with more secured design that will make life difficult for steganalyst to break even if at all they detect it. The system is multi-level in which the message needs to be encrypted, then compressed in case of high size and then, embed the message inside the cover image which is JPEG image. The rest of the paper is organized as follows: section 2 discusses the concept of digital images while related works is reviewed in section 3. The proposed model is discussed in section 4.

2. CONCEPT OF DIGITAL IMAGES

Pictures are the most common and convenient means of conveying or transmitting information [2]. A picture is worth a thousand words. Pictures concisely convey information

about positions, sizes and inter-relationships between objects. They portray spatial information that we can recognize as objects. Human beings are good at deriving information from such images, because of our innate visual and mental abilities. About 75% of the information received by human is in pictorial form [2].

An image is a collection of numbers that constitute different light intensities in different areas of the image [3]. This numeric representation forms a grid and the individual points are referred to as pixels. Most of the images consist of a rectangular map of the image pixel known as the bits where each pixel is located and its color. These pixels are displayed horizontally row by row. The number of bits in a color scheme, called the bit depth, refers to the number of bits used for each pixel [3]. The smallest bit depth in current color schemes is 8, meaning that there are 8 bits used to describe the color of each pixel. Monochrome and grayscale images use 8 bits for each pixel and are able to display 256 different colors or shades of gray. Digital color images are 24-bit images which use 3 byte per pixel to represent a color and are represented with RGB color model [12].

3. RELATED WORKS

There are Lots of techniques available that implement steganography on a variety of different electronic medium, such as F5 algorithm, LSB Coding, Palettes Modification [3]. We are using a technique that depend on the digital images as they often have a large amount of redundant data and they can take advantage of the limited power of the human visual system (HVS) [11]. Some of the researches on hiding information in an image that have been carried out and used different approach are summarized below:

[4] developed a steganography model that enhanced the LSB technique by randomly dispersing the bits of the message in the image and thus making it harder for unauthorized people to extract the original message. In his approach, the data is converted into the bytes that is, each character in the message is converted into its ASCII equivalent. Then the message bit is embedded in the digital image, since each pixel of the image typically has three numbers associated with it, one for red, green and blue intensities, and these value often range from 0 – 255. Then each bit of the message is embedded into LSB position of each pixel position.

[3] implemented a concept of steganography by using least significant bits technique which works by changing a few pixel colour value and the model was developed using MATLAB. He only considered grayscale image in his approach and messages are hidden in the least significant bits of the 8-bit binary strings representing the color numbers. Each character in a message is converted into bits and each bit of the message is mapped to a single pixel of image. The remaining bytes from the image pixel are used to encode the length of the message.

[5] designed a steganography model where image acts as a shared key between the sender and receiver and not transmitted over the channel. In their work, every character in the text (message) is converted into its integer value and that integer value is mapped to the single pixel value of the image. Index array are the only information required to recover the message back from the image.

[6] designed a steganography model using hybrid design (combination of spatial and transform domain) which produced a better peak signal-to-noise ratio (PSNR) compared to the existing transform domain techniques with improved security. In their work, the message data is embedded into the cover image by segmentation, they used DCT/DWT to generate the stego image for secret information to be transported to the destination over communication channel confidentially.

4. PROPOSED APPROACH

Steganography literally means covered writing derived from Greek word and it is the art of concealing information in ways that prevent the detection of hidden messages [11]. For a communication to occur between a sender and a receiver, sender supply message M (which can either be in form of plaintext or file), a key K and cover image C; combines them to generate a stego image Z which he sends to the receiver.

$$Z = f(M, K, C) \quad Eq.\ (1)$$

Our approach to hiding the message is quite different since we focus so much on the security and embedding capacity of our steganography system. The schematic representation of the system is shown in figure 1 which denotes the sender phase and figure 2 which represent the receiver phase.

At the sender phase, the system gets the message from the user, encrypts it and compresses it to reduce the size of the message file and passes it on for concealment into an image. At the stage for the information concealment, the system uses the secret key to encode the blue object of the image component and then embed the message inside the remaining image component; at the receiver end, the system collect the secret key from the stego image object, compare it with the secret key supplied by the receiver. If it matches, it decode the stego image with the secret bit, collect the embedded message, decompress it, decrypt it using the secret key provided and return the message file.

All data (Original Plaintext) is encoded. This means that the data is originally a combination of elements e, from some alphabet A. This combination of elements is a message M. This message from the alphabet A, is encoded into the binary alphabet B. The string of bits, binary digits (0's and 1's), is the encoded data. The encoded message B (in plaintext form P) will be encrypted to generate ciphertext C using key K.

$$E(K, P) = C, and\ D(K, C) = P \quad Eq.\ (2)$$

$where$ E represents encryption and D represents decryption. Data encryption standard (DES) is the cryptography algorithm used our work. DES is a block cipher that operates on a single chunk of data at a time, encrypting 64 bits of plaintext to produce 64 bits of ciphertext [15].

Data Compression shrinks down a file so that it takes up less space. This is desirable for data storage and data communication. The Lempel-Ziv algorithm constructs its dictionary on the fly, only going through the data once. Suppose we have broken our string up into $c(n)$ phrases, where n is the length of the string. Each phrase is broken up into a reference to a previous phrase and a letter of our alphabet. The previous phrase can be represented by at most $\log_2 c(n)$ bits, since there are $c(n)$ phrases, and the letter can be represented by at most $\log_2 \propto$ bits, where $\propto$ is the size of the alphabet. We have thus used at most eq. (3) bits total in our encoding.

$$c(n)(\log_2 c(n) + \log_2 \propto) \quad Eq.\ (3)$$

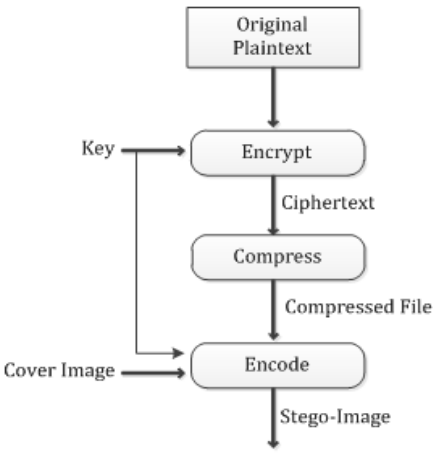

Figure 1: Stego sender phase

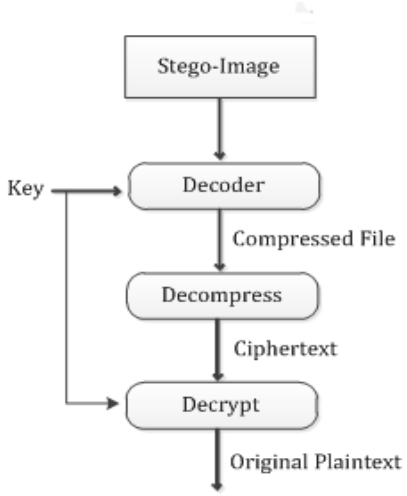

Figure 2: Stego receiver phase

Information Concealment in an Image

A digital image is a rectangular grid of pixels, or "picture elements". In a computer, images are represented as arrays of values (bytes) in a computer's memory. These values represent the intensities of the three colors R (Red), G (Green) and B (Blue), where a value for each of three colors describes a pixel. Each pixel is combination of three components(R, G, and B) [7].

For the information concealment in an image session, we need to determine the capacity of the JPG image file as shown in eq. (4)

$$\text{Capacity} = \text{DCT} - \text{DCT}_1 - \text{DCT}_0 \quad Eq.\ (4)$$

DCT is the number of DCT coefficients, DCT_0 is the number of zero Discrete Cosine Transform (DCT) coefficients and DCT_1 is the number of DCT coefficient with value 1.

In the process of a message insertion we use a key–dependent permutation (depends on a user password) that mixes all the DCT coefficients. Secret bits are inserted in the order given by the permutation. Decoder (at the receiver end) is able to repeat the same permutation only when the correct key is used. To reduce the number of modified DCT coefficients we use $(2^k - 1, 2^k - k - 1)$ Hamming codes, for $K \geq 1$. Using these codes the system inserts k-bits of secret information (denoted by $\overline{s}$) by $2^k - 1$ LSBs of DCT coefficients (denoted by $\overline{x}$). Selection of the modified DCT coefficient (denoted by c) is performed by the following relations

$$h(x) = H_c * \overline{x}^T \quad Eq.\ (5)$$

$$i = \overline{s} \otimes h(\overline{x}) \quad Eq.(6)$$

$where\ H_c$ is the control matrix of a Hamming code $(2^k - 1, 2^k - k - 1)$ and $\otimes$ is the binary bit operation xor.

Coding in the above process does not change any bit if

$$H_c * \overline{x}^T = \overline{s}^T \quad Eq.(7)$$

$when\ H_c * \overline{x}^T \neq \overline{s}^T$ the value of the LSB of the DCT coefficient in this group is changed. This increases the efficiency of inserting the secret information, because it does not modify $2^k - 2\ or\ 2^k - 1$ DCT coefficients for each k-bit block of the secret information

The system inserts a secret information using a Hamming $(2^k - 1$ or $2^k - k - 1)$ code. The k-bit block of the secret information is encoded by $2^k - 1$ LSBs of the DCT coefficients. Using the Hamming codes ensures that at most one of the LSB of the DCT coefficients is modified in each block.

The process of a message insertion uses a key–dependent permutation (depends on a user password) that mixes all the DCT coefficients of the cover image. In this sense, the secret key given by the sender is used to encode the cover image before the message embedding process takes place. The secret key is converted to its bits equivalence and the system generates a constant which will serve as a terminator after all the secret bit has been used in encoding the image, and before we start insert our message bits. The constant bits C (usually 8-bits) generated will be based on the system time, that is, day (dd) and month (mm) as shown in eq. (8).

$$C = (dd) \oplus (mm) \quad Eq.\ (8)$$

we arranged our secret bits in group of 8-bits and we encoded each group with our constant bits, after which now replace blue object of the image component with each group and continue until all our secret bits group has replaced used up, then we now replace the next blue object of the image component with the constant bit to serve as our terminator

In the same sense, we convert our message to its bits equivalent and arrange them in group of 8-bits. Then we continue from the next blue object of the image component and start replacing each of them with the message group

The algorithm below summarizes the algorithm that describe our approach

1. Encrypt using DES and compress message file
2. Convert into binary, arrange in group of eight (a byte) and store the message file
3. Convert into binary, arrange in group of eight and store the secret key
4. Select and convert our constant into binary
5. Encode each group in 3 with our constant byte and store
6. Extract and store all the image's pixel
7. Select the first pixel from 6, replace its first component with each group in 5; Continue this step until all the groups in 5 have been hidden
8. Replace the first component of the next pixel with constant to serve as terminator
9. Replace the first component of the next pixel and replace it with first group in 2; Continue until all the message group in 2 is hidden
10. Convert the resulting image pixel's bits into image

5. EXPERIMENTAL SETUP AND RESULT

The stego images remain unchanged after carrying out testing on our approach and the amount of information that our approach can hide is very high. So, this is a big advantage of using this technique for steganography. Table 1 describes the sizes of the text file and the stego images when tested with cover images such as Hydrangea.jpg, Jellyfish.jpg and Lena.jpg (the initial size of the images are 28.43, 19.15, 69.14 respectively) as shown in figure 3 while table 2 shows their respective sizes when the data is compressed before hiding. Besides, the approach used in hiding information in our cover image makes it very secured.

Figure 3: Images showing Hydrangea, Jellyfish and Lena

Table 1: Table showing plaintext message sizes and their respective stego-image sizes

Plaintext Size (KB)	Hydrangea (KB)	Jellyfish (KB)	Lena (KB)
6.20	29.29	19.75	39.76
12.4	29.29	19.75	39.76
20.60	29.2	19.79	39.75
28.7	29.30	19.77	39.40
41.30	29.30	19.81	39.75
57.40	29.31	19.80	39.80
70.4	29.30	19.80	39.80
82.7	29.37	19.84	39.76
93.8	29.36	19.87	39.76
113	29.45	20.01	39.77
134	29.57	20.19	39.72
153	29.91	20.47	39.64
165	30.14	20.78	39.69
186	30.24	20.98	39.81
207	31.80	23.35	39.92
240	34.00	26.33	40.14
295	40.21	33.83	42.27

Table 2: Table showing plaintext sizes, compressed message sizes and their respective stego-image sizes. Where PS represent plaintext size, CFS is compressed file size, H is Hydrangea, J is Jellyfish and L is Lena stego-image.

PS (KB)	CFS	H (KB)	J (KB)	L (KB)
6.20	499 b	29.27	19.72	39.77
12.4	512 b	29.27	19.72	39.77
20.60	578 b	29.27	19.72	39.75
28.7	655 b	29.27	19.72	39.77
41.30	735 b	29.28	19.72	39.75
57.40	882 b	29.28	19.72	39.85
70.4	973 b	29.28	19.73	39.76
82.7	1.04 KB	29.28	19.73	39.76
93.8	1.11 KB	29.28	19.73	39.76
113	1.24 KB	29.28	19.73	39.77
134	1.36 KB	29.29	19.73	39.77
153	1.52 KB	29.29	19.73	39.75
165	1.60 KB	29.29	19.71	39.76
186	1.76 KB	29.28	19.73	39.76
207	1.91 KB	29.28	19.75	39.95
240	2.16 KB	29.28	19.75	39.76
295	2.45 KB	29.28	19.75	39.75

6. CONCLUSION

As the result has shown, we can find out that the outcome of the paper is to create a multi-level security model of steganography that can effectively hide a message inside image file and also that the amount of information our approach can hide is higher when the information is compressed than when the information is uncompressed. As there are many applications of image steganography that allow for two parties to communicate secretly and covertly, but most of the current methods are not as necessarily secure due to the size of the data they hide and the lack of concern for the content of the message cover. Taking the cover into accounts, it is likely to increase the security of the message by encoding and hiding it in a less obvious location, which we have done. So in conclusion, as more emphasis is placed on the area of security, privacy protection, we believe that this work will do exactly that by improving the security and the hiding capacity of the existing technologies. This paper has successfully investigated whether taking the image as the cover into account increases the security of the message by creating multi-level model.

7. REFERENCES

[1] Internet – Wkipedia, www.wikipedia.com/internetworld, 2011

[2] Minakshi Kumar (2011), Digital Image Processing: Photogrammetry and Remote Sensing Division, Indian Institute of Remote Sensing, Dehra Dun

[3] Saurabh S. and Gaurav A (2010), Use of image to secure text message with the help of LSB replacement: ISSN 0976-4259, Invertis Institute of Engineering and Technology, Bareilly India.

[4] Muhalim M, Subariah I, Mazleena S and Mohd R (2003), Information Hiding Using Steganography: Department of Computer System & Communication, Faculty of Computer Science and Information system, University Teknologi Malaysia, 2003.

[5] Rupinder K, Mandeep K, Rahul M (2011), A New Efficient Approach towards Steganography: ISSN 0975-9646, International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 2 (2), 2011, 673-676

[6] Shiva-Kumar K, Raja K, and Sabyasachi P (2011), Hybrid Domain in LSB Steganography: International Journal of Computer Applications (0975 – 8887) Volume 19– No.7, April 2011

[7] Amanpreet K, Renu D, Geeta S (2009), A new Image Steganography Based on First Component Alteration Technique: International of Computer Science and Information Security. Vol 6, No 3, 2009

[8] Wu D. and Tsai W (2003), A Steganographic Method for Images by Pixel-value differencing. Pattern Recognition Letters

[9] Natarajan M and Lopamudra N (2010), Steganalysis Algorithms for Detecting the Hidden Information in Image, Audio and Video Cover Media: International Journal of Network Security & Its Application (IJNSA), Vol.2, No.1, January 2010.

[10] D. Stinson, Cryptography: Theory and Practice, 2nd Edition, Chapman and Hall, CRC, February 2002.

[11] Bret Dunbar, A detailed look at Steganographic Techniques and their use in Open-Systems Environment, SANS Institute. January 2002

[12] Neil J and Sushil J, Exploring Steganography: Seeing the Unseen: 0018-9162/98 IEEE 1998

[13] Debrup Banerjee, Assymetric Key Steganography: 2011 International Conference on Information and Electronics Engineering, IPCSIT vol 6, 2011.

[14] Steganography – Wikipedia, http://en.wikipedia.org/wiki/steganography

[15] Donal O, Micheal P and Hitesh T (2001), Electronic Payment Systems for E-Commerce: ISBN 1-58053-463-5, Artech House Inc.

Received: November 2012. Accepted: February 2013.