Journal of Computer Science 5 (1): 33-38, 2009
ISSN 1549-3636

Image Steganography by Mapping Pixels to Letters

Mohammed A.F. Al-Husainy
Department of Computer Science,
Faculty of Sciences and IT, Al-Zaytoonah University of Jordan

Abstract: Problem statement: Steganography hides the very existence of a message so that if successful it generally attracts no suspicion at all. Using steganography, information can be hidden in carriers such as images, audio files, text files, videos and data transmissions. In this study, we proposed a new framework of an image steganography system to hide a digital text of a secret message. **Approach:** The main idea for this is to use enough number of bits from each pixel in an image (7-bits in this study) to map them to 26 alphabetic English characters ('a'…'z') with some special characters that are mostly using in writing a secret message. The main goal of this method, like any steganography techniques must do, is to hide a text of a secret message in the pixels of the image in such a manner that the human visual system is not able to distinguish between the original and the stego-image, but it can be easily performed by a specialized reader machine. **Results:** This method was implemented practically on different (long and short) messages and images. The carrier images that are used in the experiments of this research have no discernible change in it. **Conclusion:** The recorded experimental results showed that this proposed method can be used effectively in the field of steganography.

Key words: Information hiding, watermarking, copyright, cryptography

INTRODUCTION

The information communicated comes in numerous forms and is used in many applications. In a large number of these applications, it is desired that the communication to be done in secrete. Such secrete communication ranges from the obvious cases of bank transfers, corporate communications and credit card purchases, on down to a large percentage of everyday email. Steganography is the ancient art of embedding a secret message into a seemingly harmless message. Most of the newer applications use steganography like a watermark, to protect a copy right on information. The forms of steganography vary, but unsurprisingly, innocuous spam messages are turning up more often containing embedded text.

Steganography or Stego as it is often referred to in the IT community, literally means, "Covered writing" which is derived from the Greek language. Steganography is defined in[1] as follows, "Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to Cryptography, where the enemy is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of Steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second message present".

There has been a rapid growth of interest in this subject over the last ten years and for two main reasons. Firstly, the publishing and broadcasting industries have become interested in techniques for hiding encrypted copyright marks and serial numbers in digital films, audio recordings, books and multimedia products; an appreciation of new market opportunities created by digital distribution is coupled with a fear that digital works could be too easy to copy. Secondly, moves by various governments to restrict the availability of encryption services have motivated people to study methods by which private messages can be embedded in seemingly innocuous cover messages. The ease with which this can be done may be an argument against imposing restrictions[2].

Information may be covered by coding as in cryptography or by hiding as in watermarking (steganpography). Many techniques can be used for hiding the digital data, from an application point of view. Many common digital hiding techniques employ graphical images or audio files as the carrier medium.

There are many ways in which messages can be hidden in digital media. Digital forensics examiners are very familiar with data that remains in file slack or unallocated space as the remnants of previous files and, of course, one can write programs that can access slack and unallocated space directly. Small amounts of data can also be hidden in the unused portion of file headers[3].

Information can also be hidden on a hard drive in a secret partition. A hidden partition will not be seen under normal circumstances although disk configuration and other tools might allow complete access to the hidden partition[4]. This theory has been implemented in a steganographic ext2fs file system for Linux. A hidden file system is particularly interesting because it protects the user from being inextricably tied to certain information on their hard drive. This form of plausible deniability allows a user to claim not to be in possession of certain information or to claim that certain events never occurred. Under this system, users can hide the number of files on the drive, guarantee the secrecy of the files' contents and not disrupt non-hidden files by the removal of the stego file driver[5-7].

Another digital carrier can be the network protocols themselves. Covert TCP by Craig Rowland, for example, forms covert communications channels using the Identification field in Internet Protocol (IP) packets or the Sequence Number field in Transmission Control Protocol (TCP) segments[3,8].

Image and audio files remain the easiest and most common carrier media on the Internet today because of the plethora of potential carrier files already in existence, the ability to create an infinite number of new carrier files and the easy access to stego software that will operate on these carriers. For that reason, we will return our focus back to image and audio files.

The most common stego method in audio and image files employ some type of Least Significant Bit (LSB) substitution (or overwriting). The "least significant bit" term comes from the numeric significance of the bits in a byte. The high-order, or most significant, bit is the one with the highest arithmetic value ($2^7 = 128$) while the low-order, or least significant, bit is the one with the lowest arithmetic value ($2^0 = 1$)[9].

Newer, more complex, steganography methods continue to emerge. Spread spectrum steganography methods are analogous to spread spectrum radio transmissions (developed in World War II and commonly used in data communications systems today) where the "energy" of the signal is spread across a wide frequency spectrum rather than focused on a single frequency, in an effort to make detection and jamming of the signal harder. Spread spectrum stego has the same function; avoid detection. These methods take advantage of the fact that little distortions to image and sound files are least detectable in the high energy portions of the carrier; i.e., high intensity in sound files or bright colors in image files. Even when viewed side-by-side, it is easier to fool human senses when small changes are made to loud sounds and/or bright colors[10].

Steganography can be used in a large amount of data formats in the digital world of today. Most of steganography research uses cover media as pictures[11], video clips[12] and sounds[13]. The most popular data formats are .bmp, .doc, .jpeg, .mp3, .txt and .wav.

Capacity, security and robustness[14], are the three main aspects affecting steganography and its usefulness. Capacity refers to the amount of data bits that can be hidden in the cover medium. Security relates to the ability of an eavesdropper to figure the hidden information easily. Robustness is concerned about the resist possibility of modifying or destroying the unseen data.

Steganography is different than cryptography and watermarking although they all have overlapping usages in the information hiding processes[15]. Steganography security hides the knowledge that there is information in the cover medium, where cryptography revels this knowledge but encodes the data as cipher-text and disputes decoding it without permission; i.e., cryptography concentrate the challenge on the decoding process while steganography adds the search of detecting if there is hidden information or not. Watermarking is different from steganography in its main goal. Watermarking aim is to protect the cover medium from any modification with no real emphasis on secrecy. It can be observed as steganography that is concentrating on high robustness and very low or almost no security.

Steganography may have different applications. For example, it can be used by medical doctors to combine explanatory information within X-ray images. It can be useful in communications for codes self-error correcting. It can embed corrective audio or image data in case corruption occurs due to poor connection or transmission. Steganography may be practical to form a secure channel for private communication, however, it does not cover the fact that the communication happened or the data is hidden. This makes steganography as a special technique of encryption or cryptography[16].

Steganography can also be utilized for posting secret communications on the Web to avoid transmission or to hide data on the network in case of a violation. It can be useful for copyright protection, which is, in reality, digital watermarking[15]. Copyright protection is to protect the cover medium from claiming its credit be others, with no real emphasis on secrecy.

MATERIALS AND METHODS

An English message text is written by using the alphabetic characters of the English language (which are 26 letters ('a'…'z')). Some other special characters are useful to use in writing messages which are giving the reader a good understanding of the message. Some of these characters that are adopted in this study: ('space character', '.', ',', '(' , ')' , '"'). Therefore, the total numbers of characters that are used to write a message become 32-characters. This means that we need at least 5-bits to represent these 23-characters in any digital system.

Now, a gray scale image is using 256 gray scales for each pixel in it. This means that we need (1-byte ≡ 8-bits) per pixel to produce ($2^{(8\text{-bits})}$ ≡ 256) gray scales.

The main operation of the algorithm in this proposed research is to map each 4-cases from (2^7 = 128) of the 7 Most Significant Bits (MSBs) in a pixel to one of the 32-cases of the (above mentioned) characters in the message. The algorithm's goal for using 4-cases instead of one case is to increase the probability of finding the matched pixels in the image that are mapped to a character in the message. Table 1 shows this mapping, we must note here the following points about Table 1:

- Each number in the column of each case is represented as a 7-bits Binary number
- Each number in the column of each case is calculated from the equation

  (Seq + (32 * (CaseNumber – 1)))

- The 1-LSB in each pixel is using by the algorithm to form a special pattern of bits string, as we will explain that later

Now, to explain how the algorithm in this proposed image stegonography system does. We consider the following:

M (Length) = Refers to the message M that we want to hid it in the stego-image, where Length represent the number of characters in the message M (which are indexed from 0…(Length-1))

I(Size) = Refers to the image I that is used as a stego-image to hid the message M, where Size is the number of pixels in the image I. (we represent the stego-image I as a one dimensional array of pixels (from the pixel at the upper-left corner to the pixel at the down-right corner (and read pixels raw by raw)). The pixels of the image I are indexed from 0…(Size-1).

T = Refers to the above mapping Table 1

k = Refers to the index of the current character in the message M

CURRENT = Refers to the index of the current pixel in I that is matched to M(k) character in the message M

NEXT = Refers to the index of the next pixel in I to be tested for matching

Table 1: Mapping table T

		8-bits (byte) of pixel				
		7-MSBs				
Seq	Message characters	Case 1 (Seq+(32*0))	Case 2 (Seq+(32*1))	Case 3 (Seq+(32*2))	Case 4 (Seq+(32*3))	1-LSB
0	'a'	0000000	0100000	1000000	1100000	-
1	'b'	0000001	0100001	1000001	1100001	-
2	'c'	0000010	0100010	1000010	1100010	-
	:	:	:	:	:	:
	:	:	:	:	:	:
25	'z'	0011001	0111001	1011001	1111001	-
26	space character	0011010	0111010	1011010	1111010	-
27	'.'	0011011	0111011	1011011	1111011	-
28	','	0011100	0111100	1011100	1111100	-
29	'('	0011101	0111101	1011101	1111101	-
30	')'	0011110	0111110	1011110	1111110	-
31	'"'	0011111	0111111	1011111	1111111	-

The general steps that the algorithm must do can be summarized as follow:

Step 1: When the algorithm start:

- Set the Length of the message M in the first two pixels (of indices 0 and 1) of the image I
- Set CURRENT = 2
- Set the 7-MSBs of the pixel in I(CURRENT) to the 7-bits (of one of 4-cases in the table T) that is represent the first character in M(0)
- Set the 1-LSB of the pixels: ($I_{1\text{-LSB}}$ (CURRENT) = 1, $I_{1\text{-LSB}}$ (CURRENT+1) = 1 , $I_{1\text{-LSB}}$ (CURRENT+2) = 0)
- And set NEXT = CURRENT+4

Step 2: For each characters (k: 1…(Length-1)) in M. The algorithm search (from the pixel I(NEXT) to the pixel I(Size-1)) for finding a pixel that (the 7-MSBs of it) is matching (to one of 4-cases in the table T) of the character M(k). For each pixel that is scanned by the algorithm, one of two cases might be happen.

Case 1: If a match pixel found. The algorithm does the following:

- Set CURRENT = NEXT
- Set ($I_{1\text{-LSB}}$ (CURRENT-2) = 0, $I_{1\text{-LSB}}$ (CURRENT-1) = 1, $I_{1\text{-LSB}}$ (CURRENT) = 1, $I_{1\text{-LSB}}$ (CURRENT+1) = 1 , $I_{1\text{-LSB}}$ (CURRENT+2) = 0)
- And set NEXT = CURRENT+4

Case 2: If a match pixel not found. The algorithm checks the following condition:
If ($I_{1\text{-LSB}}$ (NEXT-2) = 0) and
($I_{1\text{-LSB}}$ (NEXT-1) = 1) and
($I_{1\text{-LSB}}$ (NEXT) = 1) and
($I_{1\text{-LSB}}$ (NEXT+1) = 1) and
($I_{1\text{-LSB}}$ (NEXT+2) = 0) Then
Set ($I_{1\text{-LSB}}$ (NEXT) = 0)
And set NEXT = NEXT+1

We must refer that in Step 2, the algorithm is using the 1-LSB of each pixel in I to form a special pattern in the bits string of the 1-LSB of the successive pixels in I. This special pattern will be used later for extracting the hided message from the stego-image I. To clarify this, consider the following bits string of the 1-LSB of some successive pixels in I:

……010101101010101111100010100001110100001101010111011101011……

From the above bits strings, we can note that the pattern 01<u>1</u>10 is used, by the algorithm, for indicating to the place of finding the pixel in I that is matched to one of characters in the message M. Such that, the under line bit <u>1</u> in this pattern represents the 1-LSB of the matched pixel in I. The algorithm, keep using this pattern in each matched pixels in I (as in case 1 of step 2) and prevent any appearance of this pattern in any another place of unmatched pixel in I (as in case 2 of step 2).

RESULTS

The proposed algorithm of the image steganography system is tested by taking different messages of different length and hiding them in some images of different sizes. The results that are got from these experiments are recorded and can be summarized in the Table 2. Also, Table 3 shows the stego-images that are used in the above experiments, all images in the Table 1 are resized to 50% of the original images size for editing reasons.

Table 2: Experiments table

Experiment #	Image (Size)	Message length	PSNR
1	Lena (16384)	100	51.9512
2	Girls (12288)	100	53.5311
3	Lion (16384)	100	57.0207
4	Lena (65536)	880	63.7236
5	Girls (49152)	880	56.2095
6	Lion (65536)	880	62.2214
7	Lena (262144)	2000	62.5024
8	Girls (196608)	2000	69.1726
9	Lion (262144)	2000	62.4635

Table 3: Stego-images of the experiments in Table 2

 Lena (16384)	 Lena (65536)	 Lena (262144)
 Girls (12288)	 Girls (49152)	 Girls (196608)
 Lion (16384)	 Lion (65536)	 Lion (262144)

DISCUSSION

We noted that the system was success to satisfy many goals that we can conclude them in the following points:

- First, the recorded PSNR from different experiments shows that the system successes to hide a message in the stego-image without appear notable changes in the stego-image. The system takes the advantage of human visual system which cannot recognize little changes in some pixels of the image. This is the main goal of any steganography system
- Second, the using of the mapping table T in the algorithm of the system is producing to the system the effect of the first of the two main operations in any cryptography system, which is the substitution operation. This is done by the algorithm through mapping a character from the message to more than one value of 7-MSBs of the pixel. Also the mapping table T helps the system to use the second operation of any cryptography system, which is the transposition operation. This can be done by rotating the sequences Seq of characters in the mapping table T to produce many others substitution values for each characters in the message. This can be done at each time that the algorithm takes a next character from the message. Therefore, it is truly that the system works as a cryptography system in addition to its research as a steganography system even if this is done as in simple way

CONCLUSION

We designed an image steganography system by using the bits of pixel in the stego-image to hide the characters of the message. After testing the system and studied the recorded results from the experiments. We recommend using this proposed system in hiding secure information in any digital system, because this system collect the properties of both steganography and cryptography sciences.

ACKNOWLEDGMENT

The author acknowledges the support received from Al-Taiey Z. M., Amman, Jordan.

REFERENCES

1. Johnson, N.F., 2000. Steganography. http://www.jjtc.com/stegdoc/index2.html
2. Wolfgang, R.B. and E.J. Delp, 1996. Watermark for digital images. Proceeding of the IEEE International Conference on Image Processing, Sep. 16-19, IEEE Computer Society, Washington DC., USA., pp: 219-222. DOI: 10.1109/ICIP.1996.560423
3. Curran, K. and K. Bailey, 2003. An evaluation of image based steganography methods. Int. J. Digital Evid., 2: 1-40. http://www.utica.edu/academic/institutes/ecii/publications/articles/A0AD276C-EACF-6F38-E32EFA1ADF1E36CC.pdf
4. Johnson, N.F., Z. Duric and S. Jajodia, 200, Information Hiding: Steganography and Watermarking-Attacks and Countermeasures. 1st Edn., Kluwer Academic Publishers, USA., ISBN: 0-79237-204-2.
5. Anderson, R., R. Needham and A. Shamir, 1998. The steganographic file system. Lecture Notes Comput. Sci., 1525: 73-82. http://www.springerlink.com/content/jmw3k974qybrqd20/
6. Artz, D., 2001, Digital steganography: Hiding data within data. IEEE Internet Comput., 5: 75-80. http://dx.doi.org/10.1109/4236.935180
7. McDonald, A.D. and M.G. Kuhn, 2000. StegFS: A steganographic file system for linux. Lecture Notes Comput. Sci., 1768: 463-477. http://www.springerlink.com/content/f022570830606504/
8. Rowland, C.H., 1996. Covert channels in the TCP/IP protocol suite. http://outreach.lib.uic.edu/www/issues/issue2_5/rowland/index.html
9. Fridrich, J. and R. Du, 2000. Secure steganographic methods for palette images. Lecture Notes Comput. Sci., 1768: 47-60. http://www.springerlink.com/content/0m681386v4065707/
10. Wayner, P., 2002. Disappearing Cryptography-Information Hiding: Steganography and Watermarking. 2nd. Edn., Morgan Kaufmann, San Francisco, USA., ISBN: 10: 1558607692, pp: 413.
11. Chandramouli, R. and N. Memon, 2001. Analysis of LSB based image steganography techniques. Proceedings of the International Conference on Image Processing, Oct. 7-10, IEEE Computer Society, Washington DC., USA., pp: 1019-1022. DOI: 10.1109/ICIP.2001.958299
12. Doërr, G. and J.L. Dugelay, 2003. A guide tour of video watermarking. Signal Processing: Image Commun., 18: 263-282. DOI: 10.1016/S0923-5965(02)00144-3

13. Mark Noto, 2001. MP3Stego: Hiding text in MP3 files. http://www.sans.org/reading_room/whitepapers/stenganography/mp3stego_hiding_text_in_mp3_files_550
14. Chen, B. and G.W. Wornell, 2001. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Trans. Inform. Theor., 47: 1423-1443. DOI: 10.1109/18.923725
15. Provos, N. and P. Honeyman, 2003. Hide and seek: An introduction to steganography. IEEE Secur. Privac., 1: 32-44. DOI: 10.1109/MSECP.2003.1203220
16. Judge, J.C., 2001. Steganography: Past, present, future. http://www.sans.org/reading_room/whitepapers/stenganography/steganography_past_present_future_552