A New Image Steganography Based on Decimal-Digits Representation

Article *in* Computer and Information Science · October 2011

DOI: 10.5539/cis.v4n6p38 · Source: DBLP

Mohammed Abbas Fadhil Al-Husainy
Middle East University

A New Image Steganography Based on Decimal-Digits Representation

Mohammed Abbas Fadhil Al-Husainy
Department of multimedia systems, faculty of science and information technology
Al-Zaytoonah University of Jordan, Amman-Jordan
P.O. Box 130, Amman 11733, Jordan
Tel: 962-79-6846-110 Fax: 962-6-4291-432
E-mail: dralhusainy@yahoo.com; alhusainy@alzaytoonah.edu.jo

Received: September 19, 2011 Accepted: October 8, 2011 Published: November 1, 2011
doi:10.5539/cis.v4n6p38 URL: http://dx.doi.org/10.5539/cis.v4n6p38

Abstract

Steganography is the art and science of hiding important information by embedding message within other file. In this paper, a new technique to hide text message in image by using what is called image steganography. By representing the ASCII code decimal value of each character, in the secret message, as a set of separated single decimal-digit, also represent each decimal pixel value in the stego-image as a set of separated single decimal-digit. The technique creates a matching list between the decimal-digits of the characters in the secret message with the decimal-digits of the pixels in the stego-image. The technique compresses the created matching list to be as small as possible to embed it in the unused file space at the end of the stego-image file. The results show that our technique provides more security against visual attack because it does not make any changes in the pixel of the stego-image.

Keywords: Security, Steganography, Embedding, Compression, Encoding

1. Introduction

Steganography can be defined as the technique used to embed data or other secret information inside some other object commonly referred to as cover, by changing its properties. The purpose of steganography is to set up a secret communication path between two parties such that any person in the middle cannot detect its existence; the attacker should not gain any information about the embedded data by simply looking at cover file or stego file. Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from Greek, literally means “covered writing”. It includes a vast array of secret communications methods that conceal the message’s very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum (A. Cheddad, J. Condell, K. Curran, & P. Kevitt, 2010; N.F. Johnson, & J. Suhil, 2006).

The basic model of steganography uses a cover object (any object that can be used to hold secret information inside), the secret message (the secret information that is to be sent to some remote place secretly), a stego key that is used to encode the secret message to make its detection difficult and a steganography algorithm/technique (the procedure to hide secret message inside cover object). The outcome of the process is the stego object which is the object that has the secret message hidden inside. This stego object is sent to the receiver where receiver will get the secret data out from the stego image by applying decoding algorithm/technique (A. Cheddad, J. Condell, K. Curran, & P. Kevitt, 2010).

Recently, steganography is implemented by using digital media. Secret message is embedded inside digital cover media like text, images, audio, video or protocols depending upon the requirement and choice of the sender. Compared with the other types of steganography, the image steganography is most widely used. The reason behind the popularity of image steganography is the large amount of redundant information present in the images that can be easily altered to hide secret messages inside them, and because it can take advantage of the limited power of the human visual system (HVS). With the continued growth of strong graphics power in computer and the research being put into image based steganography, this field will continue to grow at a very rapid pace (A. Cheddad, J. Condell, K. Curran, & P. Kevitt, 2010; Adnan Gutub, Ayed Al-Qahtani, & Abdulaziz Tabakh, 2009; A. Kaur, R. Dhir, & G. Sikka, 2009; D. Bhattacharyya, A. Roy, P. Roy, & T. Kim, 2009).

Steganography has a wide range of applications. The major application of steganography is for secret data communication. Covert channels in TCP/IP involve masking identification information in the TCP/IP headers to hide the true identity of one or more systems. Cryptography is also used for the same purpose but steganography is more widely used technique as it hides the existence of secret data. Another application of steganography is feature tagging. Captions, annotations, time stamps, and other descriptive elements can be embedded inside an image, such as the names of individuals in a photo or locations in a map (A. Cheddad, J. Condell, K. Curran, & P. Kevitt, 2010; A. Kaur, R. Dhir, & G. Sikka, 2009; M.T. Parvez , & A. Gutub, 2008; N.F. Johnson, & J. Suhil, 2006).

Steganography can be also used to combine explanatory information with an image (like doctor's notes accompanying an X-ray). Steganography is used by some modern printers, including HP and Xerox brand color laser printers. Tiny yellow dots are added to each page. The dots are barely visible and contain encoded printer serial numbers, as well as date and time stamps. The application list of image steganography is very long (A. Cheddad, J. Condell, K. Curran, & P. Kevitt, 2010; M.T. Parvez, & A. Gutub, 2008).

In this paper, image steganography is used to hide information by performing a proposed kind of encryption. The Steganography technique is the perfect supplement for encryption that allows a user to hide large amounts of information within an image. Thus, it is often used in conjunction with cryptography so that the information is doubly protected; first it is encrypted and then hidden so that an adversary has to first find the hidden information before decryption take place (EE. Kisik Chang, J. Changho, & L. Sangjin, 2004; G. C. Kessler, 2001; H. Kathryn, 2005; H. Zhang, & H. Tang, 2007). The problem with cryptography is that the encrypted message is obvious. This means that anyone who observes an encrypted message in transit can reasonably assume that the sender of the message does not want it to be read by casual observers. This makes it possible to deduce the valuable information. Thus, if the sensitive information will be transmitted over unsecured channel such as the internet, steganography technique can be used to provide an additional protection on a secret message (Ran-Zan Wang, & Yeh-Shun Chen, 2006).

A good technique of image steganography aims at three aspects. First one is capacity (the maximum data that can be stored inside cover image). Second one is the imperceptibility (the visual quality of stego image after data hiding) and the last is robustness (security against attacks) (H. Zhang, & H. Tang, 2007).

2. Related Works

When hiding information inside images usually Least Significant Bit (LSB) method is used. In the LSB method the 8th bit of every byte of the carrier file is substituted by one bit of every bit of the secret information (Mohammad Ali BaniYounes, & AmanJantan, 2008). The LSB method usually does not increase the file size, but depending on the size of the information that is to be hidden inside the file, the file can become noticeably distorted.

Ross J. Anderson and Fabien A.P. Petitcolas argued that every steganographic approach will have its limitations; they proposed an information theoretic approach using Shannon's theory for perfect secrecy (Ross J. Anderson, & Fabian A.P. Petitcolas, 1998). In the methods that are proposed by H. Motameni and his colleague's one can embed at the dark corners of an image (H. Motameni, M.Norouzi, M.Jahandar, & A. Hatami, 2007). One can also embed the secret information in frequency domain by using Discrete Wavelet Transform method (Po Yuch Chen, & Hung Ju Lin, 2006). In this method the embedding should be done at high frequency coefficients. P. Mohan Kumar and D. Roopa suggested that one can apply block matching procedure to search the highest similarity block for each block of the secret image and embed in LSBs of the cover image (P.Mohan Kumar, & D.Roopa, 2007). Mohammed A.F. Al Husainy employed different strategy in image steganography art by mapping the pixels of image to English letters and special characters (Mohammed A.F Al Husainy, 2009). Lisa M Marvel and CharlesG Boncelet proposed to hide at the inherent noise places (Lisa M. Marvel, & Charles G. Boncelet, 1999). Ran-Zan Wang and Yeh-shun Chen also did the two way block matching for image in image steganography (Ran-Zan Wang, & Yeh-Shun Chen, 2006). But this approach is suspicious to the hackers. Xinpeng Zhang and his colleagues proposed an approach called "multibit assignment steganography for palette images", in which each gregarious color that possesses close neighboring color in the palette is exploited to represent several secret bits (Xinpeng Zhang, Shuozhong Wang, & Zhenyu Zhou, 2008). In reference (Gandharba Swain, & S.K.lenka, 2010) authors have discussed a double substitution algorithm for encrypting at sender and decrypting at receiver and the embedding process was at 7th and 8th bit positions alternatively. In (Mei-Yi Wu, Yu-Kun Ho, & Jia-Hong Lee, 2004) an image steganography with palette based images is suggested. The method is based on a palette modification scheme, which can iteratively embed one message bit into each pixel in a palette based image. In each iteration, both the cost of removing an entry color in a palette and the benefit of generating a new one to replace it are calculated. If the maximal benefit exceeds the minimal cost, an entry color is replaced. It is found that the fundamental statistics of natural images are altered by the hidden non-natural information (Alvaro Martin, Guillermo Sapiro, & GadielSeroussi, 2005). But if we do not touch the bytes those carry the image features and embed in the other bytes then the problem can be solved. As LSB embedding is very common, many steganalysis tools are available for it (SorinaDumitrescu, & Xiaolin, 2005). So LSB embedding is no more secured now-a-days. So, new embedding techniques are to be welcomed to the steganographic world. Due to the large number of steganographic tools available over the internet, a particular threat exists when criminals use steganography to conceal their activities within digital images in cyber space. Reference (Hideki Noda, MichiharuNimi, & Eiji Kawaguchi, 2006) presents two JPEG steganographic methods using Quantization Index Modulation (QIM) in the Discrete Cosine Transform (DCT) domain. The two methods approximately preserve the histogram of quantized DCT coefficients, aiming at secure steganography against histogram-based attacks.

3. Proposed Technique

An image steganography technique is presented in this section. This technique uses a bitmap images to hide a secret English message by employing a new way to encoding the secret message and hide it in the stego-image. Before going deep in the details of this technique, simple definitions that are adopt, in this technique, for the secret message and stego-image will be given below:

3.1 Secret Message

A secret message in this proposed image steganography technique is an English message which contains 26 alphabetic letters/characters (‘a’…’z’), with some special characters that are useful in writing any message to give the reader a good understanding of the message. The candidate special characters are ('space character', '.', ',', '(', ')') which form 31 characters all. As it’s known, each character has a decimal value that is representing a value of its ASCII code. Table 1 shows the desired segments, in this work, from the ASCII code table of all characters.

3.2 Stego-image

For the purpose of testing, a type of stego image that is candidate to be used in this work is a bitmap images (.bmp). In general, each file of type (.bmp) is consisting of a header part which is containing much information like (width and height of the image, number palette, number of bits for each pixel) followed by the data of the bitmap image pixels. While the pixels of each image are representing as a two dimensional array, the proposed technique looks to the pixels of the image as a one dimensional array list of bytes, which have values between (0…255), by reading the bytes of the two dimensional image row by row and stores them as a one dimensional array list.

Obviously, the operating system of any computer system stores files in the digital storage as a set of unified size of blocks (i.e., Kilobytes, Megabytes, and Gigabytes). This means that, for example, when we have a file of size (3920 bytes). This file will be stored in the digital storage as 4 Kilobytes (where: 1 Kilobyte=1024 bytes), such an operating system strategy remains 176 bytes unused at the end of this file. These unused bytes at the end of the (.bmp) image file will be used by the proposed steganography technique to store the encoding information of the secret message within the stego-image.

3.3 The Embedding Process Framework

1. Preparation Operation

First of all, the technique change the ASCII code value of the alphabetic characters ‘a’…’z’ in Table1 to be represented in two decimal digits only by subtracting 97 from each ASCII code value. The new sequence of the ASCII code values of the alphabetic character ‘a’…’z’ become (0…25). The ASCII code of the remaining special characters in Table1 is already represented in two decimal digits. The modified Table 1will be shown as Table 2.

After that, the technique creates a list ***M***, such that each element in this list is representing one decimal digit of the ASCII code of the characters in the secret message. The length of the list ***M*** (i.e., number of elements) is equal **double** of the length of the secret message (i.e., number of characters in it). For example:

***Secret Message*:** steganographyart.

***M*:** 1 8 1 9 0 4 0 6 0 1 3 1 4 0 6 1 7 0 0 1 5 0 7 2 4 3 2 0 0 1 7 19 4 6

In the same manner, the technique creates a list ***D*** from the one dimensional array list of bytes of the stego-image, such that each element in this list is representing one decimal digit of each byte in the image (where each byte in this work is representing as a three decimal digits, (i.e., the byte value 22 is representing as 022)). The length of the list ***D*** (i.e., number of elements) is equal **triple** of the length of the one dimensional array list of bytes of the stego-image. For example:

***One dimensional array list of bytes of the image*:** 16816893441709

***D*:** 168168 0 93044170009……….

2. Build Lists of Frequencies

   The next operation of the proposed technique is building a two list of frequencies for the decimal digits (sorted in descending order depend on the frequencies of the decimal digits '0'…'9'),one list for the frequencies of the elements in list ***M*** and other for the frequencies of the elements in the list ***D***. These two lists will be used later, in the next section (3), to match the first high frequency decimal digit in the list ***M*** with the first high frequency decimal digit in the list ***D***, and the second high frequency decimal digit in the list ***M*** with the second high frequency decimal digit in the list ***D***, etc. For example: if the number **1**has most occurrences in the ***M*** list, then its right to map this value to such element that also has the most occurrences in the ***D*** list. This operation will help the technique to find as much as possible match between the two lists ***L*** and ***D***.

3. Create The Matching List ***L***

   The main operation of the proposed technique is creating the match list ***L*** by finding the match between each element in the list ***M*** with the elements in the list ***D***, where each element in the list ***L*** represents either 0 or 1, (where 0: false/not match and1: true/match).To clarify this operation, consider the list M and D from the above two examples:

   ***M*:** **1 8 1 9 0 4 0** 6 0 1 3 1 4 0 6 1 7 0 0 1 5 0 7 2 4 3 2 0 0 1 7 1 9 4 6

   ***D*:** **1** 6 **8 1** 6 8 0 **9** 3 **0 4** 4 1 7 **0** 0 0 9 ……….

   ***L*: *1 0 1 1 0 0 0 1 0 1 1 0 0 0 1* ………**

   After create the matching list ***L***, re-represent the list ***L*** as a new list ***B***, such that each element in ***B*** represents the number of continuous values of 0s or 1s in the list ***L***. For example, for the above created list ***L***, the list ***B*** will be as follow:

   ***B: 1 1 2 3 1 1 2 3 1* ………**

   From the above B list the successive element values represent number of ones followed by number of zeros followed by number of ones, and so on. By saving what the first element in the list B represents (i.e., number of zeros/number of ones), we can know what the rest elements represent. This information will store through the embedding operation to be used next in the extracting process.

4. Compress the B List

   After the list ***B*** created, our proposed technique try to minimize the size (in byte) that is required to store this list in the unused bytes at the end of the (.bmp) stego-image file. One of the most well-known compression methods is Huffman coding method; this is a candidate compression method that is used in this proposed technique for this purpose.

5. Embedding Operation

   When the proposed technique complete the above four steps, all the necessary information about the secret message will be hide/embed in the unused bytes at the end of the (.bmp) stego-image file. This information includes: the compressed ***B*** list, the list of frequencies that is created from the list ***M*** in step (2) and the Huffman coding table.

3.4 The Extracting Process Framework

When the receiver get the file of the stego-image, it is easy to read all the information about the embedded secret message from the unused bytes space at the end of the (.bmp) stego-image file. Then decompress the ***B*** list by using the Huffman coding table and then convert it to the list ***L***. After that, the receiver extracts the list ***M*** from the two lists ***D*** and ***L*** by finding the match positions in the list ***D***. At the end of the extracting process, use the list of frequencies to reconstruct the original secret message from the list ***M***.

4. Experimental Results and Discussion

To have a look (in numbers) about the performance of the proposed steganography technique, some experiments are done by using different stego-images of different sizes (see Table 3) to hide randomly selected secret messages of different length. The required programs to implement the proposed technique are written by using C++ programming language and executing them on a computer system of 2.53GHz processor with 4.0 GB memory and Microsoft Windows 7 operating system. The details results of these experiments are recorded in Table 4.

From the above operations that are done in the technique and the results in Table 4, we can record the following points about the new proposed image steganography technique:

- The length of the secret message (i.e., number of characters) which can be embedded in the stego-image is depending mainly on the number of unused bytes at the end of the stego-image file. In case that the length of the secret message is too long to hide in the unused byte at the end of the stego-image file, the technique inform the owner of the message and suggest him/her to split the message into more than one parts and then hide these parts in different stego-images or using the same stego-image many times to hide the parts of the message.
- The proposed technique applies some types of encoding on the embedded information in the stego-image; this will increase the security of the secret message that is covered by the stego-image. The encoding is applied twice: First, when the proposed technique modifies the ASCII code value of the alphabetic characters in Table 2. We can note here that the technique can get, from time to time, a new modified ASCII code table for the characters of the secret message. This can be done by using any ASCII code values that are representing in maximum two decimal-digits (i.e., 0…99).Second, when the proposed technique uses the Huffman coding method to compress the list ***B***. This will add additional difficulties in front of the attackers. We can note here that the Huffman coding method will produce different codes for the embedded information when we using different stego-image to hide the same secret message.
- Because the proposed technique does not make any changes in the pixels values of the stego-image, and tries to hide all the secret information of the message in the unused space at the end of the stego-image file. Therefore, by using Human Visual System (HVS), the attackers don't believe that the stego-image contain any secret information in it. This is a good point in this steganography technique and there is no need to calculate the Peak Signal to Nose Ratio (PSNR) because there is no distortion appears in the pixels values of the stego-image. While the stego-image in all the steganography technique that is using the Least Significant Bit (LSB) to hide the secret message suffer from having some distortion that is appearing in the pixels values.
- By comparing the time that is required in the embedding and extracting operation of the proposed technique with the most well-known simple steganography technique (i.e., such a technique that is using Least Significant Bit (LSB) of the pixel value to hide the secret message) in Table 5. It is really clear that the proposed technique need a time near to the time that the simple LSB technique need in both embedding and extracting operation.

5. Conclusion and Future Direction

In this paper, a new technique to hide English text message inside images was presented. The main objective of this technique is to add more security to the message by applying some types of coding to the characters in the secret message and saving the pixels values of the stego-image no changed by hiding all the secret information about the message in the unused bytes at the end of the stego-image file to exclude the doubt about existing any secret message in the stego-image. The experimental results show that this technique is performing well in hiding secret messages in images and it was success to overcome some of problems that are appeared in other steganography techniques.

The future works include: increasing the capability of this technique by finding the best matched block of pixels in the stego-image with the characters of the secret message, trying to provide the technique immunity about any rotation and resizing operations that might be done on the stego-image, applying this technique on another types of stego-files like (text, audio, video).

References

A. Cheddad, J. Condell, K. Curran, & P. Kevitt. (2010). Digital image steganography-survey and analysis of current methods. *Signal Processing*, 90, 727-752. http://dx.doi.org/10.1016/j.sigpro.2009.08.010

A. Kaur, R. Dhir, & G. Sikka. (2009). A new image steganography based on first component alteration technique. *International Journal of Computer Science and Information Security (IJCSIS)*, 6, 53-56.

Adnan Gutub, Ayed Al-Qahtani, & AbdulazizTabakh. (2009). Triple-A: secure RGB image steganography based on randomization. AICCSA, IEEE/ACS International Conference on Computer Systems and Applications, Rabat, Morocco, 400-403. http://dx.doi.org/10.1109/AICCSA.2009.5069356

Alvaro Martin, Guillermo Sapiro, & GadielSeroussi. (2005). Is Steganography Natural. *IEEE Transactions on Image Processing*, 14(12), 2040-2050. http://dx.doi.org/10.1109/TIP.2005.859370

D. Bhattacharyya, A. Roy, P. Roy, & T. Kim. (2009). Receiver compatible data hiding in color image. *International Journal of Advanced Science and Technology*, 6, 15-24.

EE.Kisik Chang, J. Changho, & L. Sangjin. (2004). High Quality Perceptual Steganographic Techniques. *Springer*, 2939, 518-531.

G. C. Kessler. (2001). Steganography: Hiding Data Within Data. An edited version of this paper with the title "Hiding Data in Data". *Windows & .NET Magazine*. [Online] Available: http://www.garykessler.net/library/steganography.html (October 4, 2011)

GandharbaSwain, & S.K.lenka. (2010). Steganography-Using a Double Substitution Cipher. *International Journal of Wireless Communications and Networking*, 2(1), 35-39. ISSN: 0975-7163.

H. Kathryn. (2005). *A Java Steganography Tool*. http://diit.sourceforge.net/files/Proposal.pdf

H. Motameni, M.Norouzi, M.Jahandar, & A. Hatami. (2007). Labeling method in Steganography. Proceedings of world academy of science, engineering and technology, 24, 349-354. ISSN 1307-6884.

H. Zhang, & H. Tang. (2007). A novel image steganography algorithm against statistical analysis. *Proceeding of the IEEE*, 19, 3884-3888.

Hideki Noda, MichiharuNimi, & Eiji Kawaguchi. (2006). High-performance JPEG steganography using Quantization index modulation in DCT domain. *Pattern Recognition Letters*, 27, 455-461. http://dx.doi.org/10.1016/j.patrec.2005.09.008

Lisa M. Marvel, & Charles G. Boncelet. (1999). Spread Spectrum Image Steganography. *IEEE Transactions on Image Processing*, 8(8), 1075-1083.http://dx.doi.org/10.1109/83.777088

M.T. Parvez, & A. Gutub. (2008). RGB intensity based variable-bits image steganography. *APSCC 2008 –Proceedings of 3rd IEEE Asia-Pacific Services Computing Conference,* Yilan, Taiwan.

Mei-Yi Wu, Yu-Kun Ho, & Jia-Hong Lee. (2004). An iterative method of palette-based image steganography. *Pattern Recognition Letters,* 25, 301-309.http://dx.doi.org/10.1016/j.patrec.2003.10.013

Mohammad Ali BaniYounes, & AmanJantan. (2008). A New Steganography Approach for Image Encryption Exchange by using the LSB insertion. *IJCSNS International Journal of Computer Science and Network Security*, 8(6), 247-254.

Mohammed A.F Al Husainy. (2009). Image Steganography by mapping Pixels to letters. *Journal of Computer Science*, 5(1), 33-38. ISSN 1549-3636.

N.F. Johnson, & J. Suhil. (2006). Exploring Steganography: Seeing the Unseen. *Computing Practices*.

P.Mohan Kumar, & D.Roopa. (2007). An Image Steganography Framework with Improved Tamper Proofing. *Asian Journal of Information Technology*, 6(10), 1023-1029. ISSN: 1682-3915.

Po Yuch Chen, & Hung Ju Lin. (2006). A DWT Based Approach for Image Steganography. *International journal of Applied Science and Engineering*, 4(3), 275-290.

Ran-Zan Wang, & Yeh-Shun Chen. (2006). High Payload Image Steganography Using Two-Way Block Matching. *IEEE Signal Processing letters*, 13(3), 161-164.http://dx.doi.org/10.1109/LSP.2005.862603

Ross J. Anderson, & Fabian A.P. Petitcolas. (1998). On the Limits of steganography. *IEEE Journal of selected Areas in communication*, 16(4), 474-481. Special Issue on Copyright and Privacy protection. ISSN 0733-8716.

Sorina Dumitrescu, & Xiaolin. (2005). A New Framework of LSB Steganalysis of Digital Media. *IEEE Transactions on Signal Processing*, 53(10), 3936-3947.http://dx.doi.org/10.1109/TSP.2005.855078

Xinpeng Zhang, Shuozhong Wang, & Zhenyu Zhou. (2008). Multibit Assignment Steganography in Palette Images. *IEEE Signal Processing Transactions*, 15, 553-556. http://dx.doi.org/10.1109/LSP.2008.2001117

Table 1. Selected Segments of ASCII Code Table

Character	Decimal Value	Description
	32	Space
(	40	Open parenthesis (or open bracket)
)	41	Close parenthesis (or close bracket)
,	44	Comma
.	46	Period, dot or full stop
a	97	Lowercase a
b	98	Lowercase b
c	99	Lowercase c
d	100	Lowercase d
e	101	Lowercase e
f	102	Lowercase f
g	103	Lowercase g
h	104	Lowercase h
i	105	Lowercase i
j	106	Lowercase j
k	107	Lowercase k
l	108	Lowercase l
m	109	Lowercase m
n	110	Lowercase n
o	111	Lowercase o
p	112	Lowercase p
q	113	Lowercase q
r	114	Lowercase r
s	115	Lowercase s
t	116	Lowercase t
u	117	Lowercase u
v	118	Lowercase v
w	119	Lowercase w
x	120	Lowercase x
y	121	Lowercase y
z	122	Lowercase z

Table 2. Modified Values of the ASCII Code in Table 1

Character	Decimal Value	Description	Character	Decimal Value	Description
	32	Space	l	11	Lowercase l
(	40	Open parenthesis (or open bracket)	m	12	Lowercase m
)	41	Close parenthesis (or close bracket)	n	13	Lowercase n
,	44	Comma	o	14	Lowercase o
.	46	Period,dot or full stop	p	15	Lowercase p
a	00	Lowercase a	q	16	Lowercase q
b	01	Lowercase b	r	17	Lowercase r
c	02	Lowercase c	s	18	Lowercase s
d	03	Lowercase d	t	19	Lowercase t
e	04	Lowercase e	u	20	Lowercase u
f	05	Lowercase f	v	21	Lowercase v
g	06	Lowercase g	w	22	Lowercase w
h	07	Lowercase h	x	23	Lowercase x
i	08	Lowercase i	y	24	Lowercase y
j	09	Lowercase j	z	25	Lowercase z
k	10	Lowercase k			

Table 3. Bitmap Stego-Images (Width × Height × Palette)

Beach (360×540×3)	Lake (256×256×3)	Baby-Girl (768×1024×3)
Bridge (600×394×3)	Taj-Mahal (600×494×3)	Lighthouse (1920×1200×3)

Table 4. Experimental Results

Bitmap Stego-Image	Beach	Bridge	Lake	Taj-Mahal	Baby-Girl	Lighthouse
Image Size (byte)	583200	709200	196608	889200	2359296	6912000
Number of Unused Bytes	426	378	970	602	970	970
Size on Disk (byte)	583680	709632	197632	889856	2360320	6913024
Length of Secret Message (Characters)	300	220	810	460	845	875
Size of Embedded Information (byte)	405	340	945	575	947	950
Time of Embedding Process (Second)	0.296	0.327	0.093	0.343	0.826	2.402
Time of Extracting Process (Second)	0.150	0.165	0.047	0.172	0.420	1.203

Notes:

- Image Size = Width × Height × Palette
- Header Size of the (.bmp) image type = 54 bytes
- Size on Disk = Image Size + Header Size + Number of Unused Bytes

Table 5. Time compression between the proposed technique and the simple LSB technique

		Beach	Bridge	Lake	Taj-Mahal	Baby-Girl	Lighthouse
The proposed technique	Time of Embedding Process (Second)	0.296	0.327	0.093	0.343	0.826	2.402
	Time of Extracting Process (Second)	0.150	0.165	0.047	0.172	0.420	1.203
Technique that is using Least Significant Bit (LSB)	Time of Embedding Process (Second)	0.156	0.202	0.062	0.234	0.577	1.700
	Time of Extracting Process (Second)	0.153	0.200	0.060	0.231	0.574	1.698