ISSN: 0067-2904
GIF: 0.851

Text Hiding in Color Images Using the Secret Key Transformation Function in GF (2^n)

Nada Hussein M. Ali[1*], Abdul Monem S. Rahma[2], Abeer Salim Jamil[3]
[1]Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq
[2]Department of Computer Science, University of Technology, Baghdad, Iraq
[3]Computer Science & Information Systems Department, Al-Mansour University College (MUC), Baghdad, Iraq

Abstract

Steganography is one of the most popular techniques for data hiding in the different media such as images, audio or video files. This paper introduced the improved technique to hide the secret message using the LSB algorithm inside the RGB true color image by encrypting it using the secret key transformation function. The key is selecting randomly in the GF (2^n) with condition it has an inverse value to retrieve the encrypted message. Only two bits are used for the low byte in each pixel (the blue byte) to hide the secret message, since the blue color has a weak effect on human eyes. The message hidden by the suggested algorithm is less vulnerable to be stolen than other similar applications.

Keywords: Steganography, Color images, LSB algorithm, Secret key, GF(2^n), PSNR.

إخفاء النص في الصور الملونة باستخدام وظيفة تحويل المفتاح السري في($GF(2^n$)

ندى حسين محمد علي[1*]، عبد المنعم صالح رحمة[2]، عبير سالم جميل[3]

[1]قسم علوم الحاسبات، كلية العلوم، جامعة بغداد، بغداد، العراق

[2]قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق

[3]قسم علوم الحاسبات وانظمة البيانات، كلية المنصور الجامعة، بغداد، العراق

الخلاصة

إخفاء المعلومات هي واحدة من التقنيات الأكثر انتشارا لاخفاء للبيانات في وسائط مختلفة مثل ملفات الصور ، الصوت او الفديو. قدمت هذه الورقة تقنية محسنة لإخفاء الرسائل السرية باستخدام خوارزمية LSB داخل صور RGBذات اللون الحقيقي عن طريق تشفيرها باستخدام وظيفة تحويل المفتاح السري. يتم تحديد المفتاح عشوائيا في GF (2^n) مع شرط أنه يحتوي على قيمة المعكوس وذلك لاسترجاع الرسالة المشفرة. ويتم استخدام بت اثنين فقط من البايت الاقل تسلسل في كل بكسل (البايت الأزرق) لإخفاء الرسالة سرية، لأن اللون الأزرق له تأثير ضعيف على العين البشرية. الرسالة الخفية التي يتم اخفاءها في الخوارزمية المقترحة هي أقل عرضة للسرقة من التطبيقات الأخرى المماثلة بسبب وجود دالة التحويل.

Introduction

The security of information becomes one of the most important factors of information technology and communication because of the huge rise of the World Wide Web and the copyright laws [1]. Information hiding can be achieved into four phases are: preliminary phase in which an encryption technique is applied. Embedded phase in concerned with algorithms which are used for information hiding. The transmission phase and finally the extraction phase. For each step a security issue must be considered. Information hiding can be used in different applications include copyright, military, confidential communication, digital elections, E-commerce, copy control, authentication. Hiding

*Email: nada_husn@yahoo.com

information is better than ciphering in the aforementioned fields, because in the former, nobody can notice any information hiding for a message in image [2].

In steganography, image has become an essential, potential, and popular file to be used as the carrier file for protecting the confidential information. But actually, the theory had said that all of the digital files could be used as a carrier file or the message [3].

The secret message, cover message, embedding algorithm and the secret key are the main four terminologies used in the steganography systems. The secret message is defined as the data or information which is needed to be hidden in the appropriate digital media. While; the cover message is considered as the carrier of the hidden message such as image, video, text or any other digital media. The embedding algorithm is the most important part; it can be defined as a method or the ideas that usually used to embed the secret information in the cover message to prevent unauthorized people to get it [4].

The LSB coding is suitable to work with any type of data file format, hence easily combine with any technique, but unfortunately, LSB encoding lack robustness and security. In this technique the data are only hidden in the last bit, which lower its importance. In this technique, an attacker can easily identify and uncover the message by just removing the complete LSB plane. Many improvements have been taken to enhance the performance of LSB coding [5], for more details see [6] and [7].

Related Work

Mamta Juneja and Parvinder S. Sandhu in [8] presents an embedding algorithm for hiding encrypted messages in nonadjacent, and random pixel locations in edges and smooth areas of images.

Deepesh Rawat and Vijaya Bhandari in [9] proposed two techniques for hiding the data of the secret image in the cover image. In the first proposed technique, each of the two MSB bits of secret image plane (red, green, blue) are replaced by corresponding two LSB bits for each plane in the cover image. While in the second technique, the first LSB bit, two LSB bits and three LSB bits of the red, green and blue channels in the cover image is replaced by the first MSB bit, two MSB bits and three MSB bits of red, green and blue channels in the secret image respectively. In the both techniques, each pixel in the cover image could hide six bits of secret image.

G.R.Manjula1 and AjitDanti in [10] suggested a method for hiding secret image inside a color cover image based on the limitation of the human eyes for recognizing RGB colors. In this method, a hash function is used to insert 2-3-3 least significant bits for each channel (red, green, blue) in the color image. The mean square error (MSE) and the Peak Signal to Noise Ratio (PSNR) values of the proposed methods have been improved when using the hash function

The Transformation Operations in the Finite Fields of The Form GF(2^n)

The finite field arithmetic operations are different from the traditional integer arithmetic operations. The finite fields have two main properties: the first one is that this field has finite elements; while the second property is that all results of the operations performed in this field are resulting in an element in the same field [11]. The purpose for using the finite field; which is called also Galois Field (GF); is appropriate for computer applications since its components are 0 and 1. That is, the computer also consist of two numbers 0 and, which make them identical [12].

Let p be a prime number. The integers mod p, consisting of the integers {0, 1,2, . . ., p−1} with addition and multiplication performed mod p, is a finite field of order p. The finite field of order p^n is generally written as GF(p^n). A particular case of Finite Field that this study is interesting when the prime (p) = 2, it is conventional to express elements of GF(2^n) as binary numbers. Finite fields of order 2^n are called binary fields or characteristic-two finite fields. One way to construct GF(2^n) is to use a polynomial basis representation. Here, the elements of GF(2^n) are the binary polynomials (polynomials whose coefficients are in the field GF(2) = {0,1}) of degree at most n−1. A polynomial f(x) in GF(2^n) is presented as in equation 1 which can be uniquely represented by its n binary coefficients ($a_{n-1}\ a_{n-2}\ldots a_0$) [13].

$$f(x) = a_{n-1}\ x^{n-1} + a_{n-2}\ x^{n-2} + \cdots + a_1\ x + a_0 = \sum_{i=0}^{n-1} a_i\ x^i \quad (1)$$

Thus, every polynomial in GF (2^n) can be represented by an n-bit number. Irreducibility of f(x) of degree less than n means that f(x) cannot be factored as a product of binary polynomials [14]. Addition, multiplication, division, exponentiation and inverse multiplication are the most basic arithmetic operations in a finite field. Addition and subtraction are performed by adding or subtracting

two of these polynomials together, and reducing the result modulo the characteristic [11]. Representing data as a vector in a Galois Field allows mathematical operations to scramble data easily and effectively [12].

Color Images and Least Significant Bit Technique

The file formats with a high degree of redundancy is more suitable to use for steganography. One may define the redundancy term as the numbers of bits are greater than necessary to obtain high accuracy of the displayed objects. Any alteration in the redundant bits of an object cannot be detected easily; this property is the most appropriate for hiding the secret messages. The most suitable file formats are images and audio that is conforming to these requirements. The main four categories of the file format are shown in Figure-1 which used for steganography [15].

A digital image is considered the most common type carrier used for steganography. One can define the image as two dimension array of integer numbers, the height represents the rows and the width represents the columns. Each element in this array is denoted point of colors which called pixels, the sizes of the image is specified in pixels. Each pixel in the image is indexed by x and y coordinates and stored 24 bits in color images and 8 bits in gray scale images. The 24 bit images are extended over three bytes; these bytes are RGB (red, green, blue) colors respectively. The colors are found by mixing these three RGB colors in different properties [16].

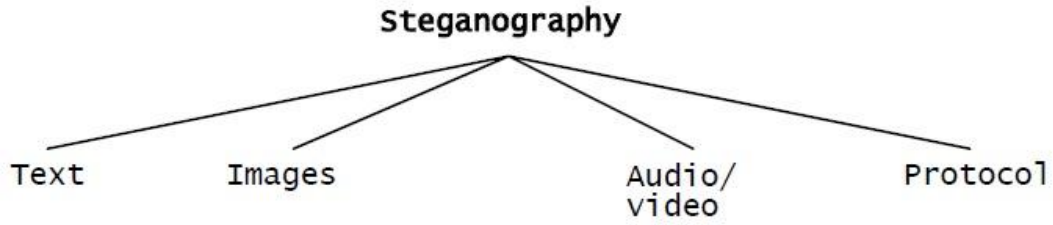


Figure 1- Categories of steganography [15]

When using a 24-bit image, a bit of each byte in the pixel red, green and blue color components can be used. In other words, one can store 3 bits in each pixel. For Example, An 800 × 600 pixel images, can store a total of 1,440,000 bits or 180,000 bytes of embedded data The LSB technique is considered the best technique to apply to BMP images. The reason is that most researchers focus for the transmission of the information amount and not on the information secrecy [15]. Encryption is used to increase the security of data embedded in the cover image. When hidden data embedded in the cover image, it must not affect in its quality or, in other word, not causing any type of image degradation. Another problem occurs when the hidden data have size larger than the cover image; this will yield increasing in the cover image distortion. Thus, the main goal is to minimize image degradation when embedded any data into the cover image [16].

Peak Signal-to-Noise Ratio (PSNR)

Nowadays, the most popular distortion measures in the field of image and video coding and compression are the signal-to-noise ratio (SNR), and the peak signal-to-noise ratio (PSNR). They are usually measured in decibels (dB) [17]. The PSNR measurements have been used to check the quality of the steganography images. PSNR is a standard quantity used in steganography technique in order to test the quality of the steganography images as shown in Equation 2. The higher the value of PSNR, the more quality the steganography image will have [18].

$$PSNR = 10.\log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (2)$$

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad (3)$$

In equation 2, Max is a maximum value in the image (for example, in a gray scale image, Max is 255) while MSN is the mean squared error. In equation 3 are the height and width of the image, respectively. I(i,j) is the original value of the image, and K(i,j) is the value after a hidden the information in the steganography image [17].

The Proposed Transformation Function in GF(2^n) for Hiding Text in Image

Only the low byte of the image (the blue bandwidth) is used to hide the raw data, since the blue color has a weak effect on human eyes. The maximum size of the characters that can be embedded into the LSB of the image is *T* bits. Two bits are extracted from each blue byte and ANDes' with the consecutive bits of each character in the text file to be hidden.

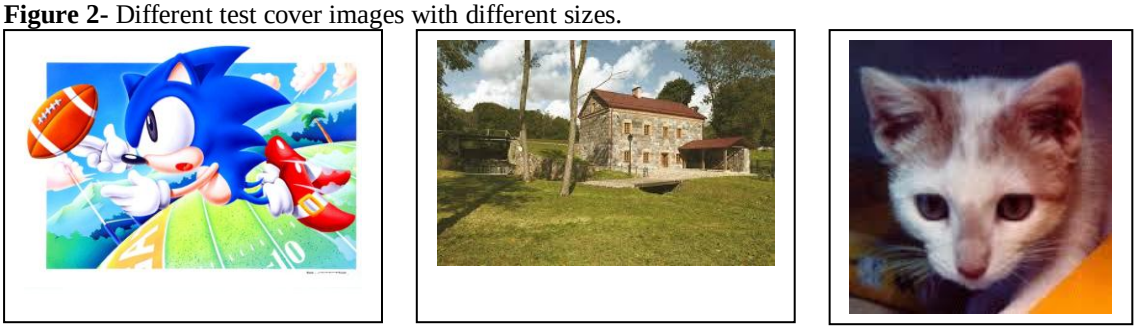
The *Hide_Text_Image_Tra_Fun* shows in details the implementations of the proposed algorithm, which transfer the data in GF (2^n) modulate irreducible polynomial, and hide it in Stego_image. Different irreducible polynomials of degree GF(2^n) could be used to implement the multiplication operations between the data and the secret key. The retrieve data and extraction operations performed in reverse order. The proposed system, which is designed in Visual C^{++} , uses different tested colored images.

Hide_Text_Image_Tra_Fun
Input: *Cover_Image, Tra_Key, Data_Text_File* ***Output***: *Stego_Image* ***Step1*** : Read *Tra_Key*, do steps 2-7 ***Step2:*** Read single character from Data_Text_File, 4 pixels from Cover_Image. ***Step3:*** Extract the last two bits from every read pixel to form single char *char_s.* ***Step4:*** Multiply (*Tra_Key* char_s*) in GF(2^n) mod irreducible polynomial. ***Step5:*** Split char_s into four parts (each part has two bits long), and save each part to every read pixels in the same order that extracted from. ***Step6:*** Save the four modified pixels in the *Stego_Image.* ***Step7*** : Repeat to multiple images until the end of Data_Text_File

The system is composed of two parts; the first one is designed to transfer the data and hide it in multiple images depend on the data size using the proposed *Hide_Text_Image_Tra_Fun* algorithm, while the second part is designed to retrieve the hidden data from the steganography images and transfer it to obtain the origin text file. The system is tested in GF (2^8), where each character in the text file is multiplied with the Tra_Key (selected by the user) mod irreducible polynomial $x^8+x^4+x^3+x+1$. Each encrypted output character is divided into four parts (2-bits for each) and hide in four successive pixels in the *Stego_Image*. Different BMP cover images of different sizes were used to test the proposed *Hide_Text_Image_Tra_Fun* algorithm, as shown in Figure-2. In Figure-3 the output images obtained when execute the proposed algorithm.

a) Cartoon Image (b) House Image (c) Cat Image

Figure 2- Different test cover images with different sizes.



(a) Cartoon Stego-Image (b) House Stego-Image (c) Cat Stego-Image

Figure 3- The Stego Images when execute the proposed algorithm *Hide_Text_Image_Tra_Fun*

Results and Discussion

The proposed algorithm ***Hide_Text_Image_Tra_Fun*** has been tested by several BMP images. For each tested image the capacity measurement has been calculated to estimate the number of stored bits that can be hidden in the image, as shown in equation 4:

$$Total\ Hidden\ Bytes = \frac{Image\ Height\ X\ Image\ Width\ X\ 2}{8} \quad (4)$$

Table-1 demonstrates the execution of the proposed algorithm for different images. The tested images are selected depending on their texture, for example the Cartoon image has a clear color, while the House image has a gradation and overlapping colors and in the Cat image the face colors are clearer than background. The second and third fields in the table give the total size of each tested color image in kilobyte (KB) and the Width * Height in pixels respectively, while the fourth field gives the estimated number of KB that could be embedded (or hide) into the image. The last field gives the approximate percentage ratio of the size of the hidden data that embedded in the original file when using two bits of blue byte for each pixel in the image.

Table-2 demonstrates the MSE and PSNR for different steganography images after change their values using the proposed algorithm Hide_Text_Image_Fun. From the obtained results in this table, the best PSNR and MSE was in the House image; the reason is that the front and background colors are overlapped and cannot distinguishable from each other.

The advantage of the proposed algorithm is to achieve higher confusion property for several pixels; this property gives high security against any attack since it needs to know the secret key to recover the original data.

Table 1- Different BMP image files and the maximum hidden data in KB.

Image file name	Image size (KB)	Width * Height In pixels	Total estimated capacity hidden (KB) in the image	Percentage ratio used from image file for hiding information
Cartoon	901	640 *480	75	12%
House	149	275* 183	13	11.5%
Cat	82	164* 170	7	11.7%

Table 2- The MSE and PSNR for different images

Image name	Image size (KB)	MSE	PSNR (dB)
Cartoon	**901**	**0.307**	**53.2**
House	**149**	**0.012**	**67.35**
Cat	**82**	**0.026**	**63.96**

Conclusions

The goal of the proposed algorithm is to increase the security of the hidden data by using a secret key transformation in GF(2^n). The first benefit of such key is to increase the confusion property by extract only two bits from consequent four pixels in RGB color images and hide the information inside. The second benefit is that the same secret key has different transformation values depend on the irreducible polynomial used in the operations performed in GF(2^n) .Another advantage, only the low byte or blue bandwidth is used to extract bits and exclusive Andes' with the origin data, since the blue color has a weak effect on human eyes. The higher the value of PSNR, the more quality the steganography image will have. The obtained PSNR results shown that quality of embedded image is good.

References

1. Firas A. J. **2013**. A Novel Steganography Algorithm for Hiding Text in Image using Five Modulus Method, *International Journal of Computer Applications*, 72(17).
2. Ahmed I. and Arwa Z. **2009**. Algorithm for Text Hiding in Digital Image for Information Security, *International Journal of Computer Science and Network Security*, 9(6).
3. Jasril, Ismail M. and Faisal R.**2013**. Capacity Enhancement of Messages Concealment in Image and Audio Steganography, *International Journal on Smart Sensing and Intelligent System*, 6(5).

4. Vipul S. and Sunny K. **2013** .A New Approach to Hide Text in Images Using Steganography, *International Journal of Advanced Research in Computer Science and Software Engineering*,3(4).

5. Pooja C., Minu C. and Chandrakant B.**2013.** Enhancement in Security of LSB based Audio Steganography using Multiple Files. *International Journal of Computer Applications*,73(7).

6. Vijay K. S. and Vishal S. **2012**. A Steganography Algorithm for Hiding Image in image by Improved LSB Substitution by Minimize Detection, *Journal of Theoretical and Applied Information Technology*, 36(1).

7. Suma S. and Dharmambal. **2015**. A Novel Image Steganography based on Secured Inversion Technique, *International Journal of Innovative Research in Computer and Communication Engineering*, 3(6).

8. Mamta J. and Parvinder S. S. **2013**. An Improved LSB Based Steganography Technique for RGB Color Images, *International Journal of Computer and Communication Engineering*, 2(4).

9. Deepesh R. and Vijaya B. **2013**. A Steganography Technique for Hiding Image in an Image using LSB Method for 24 Bit Color Image, *International Journal of Computer Applications*,64(20).

10. G.R.Manjula and AjitDanti. **2015**. A Novel Hash Based Least Significant Bit (2-3-3) Image Steganography in Spatial Domain, *International Journal of Security, Privacy and Trust Management*, 4(1).

11. Madhuri O.B.B., Rambabu E. and Malijeddi M.**2012**. Design and Implementation of Arithmetic Unit for GF(2m), *International Journal of Advanced Research in Computer Engineering & Technology*, 1(9).

12. Benvenuto, C. J.**2012**. *Galois Field in Cryptography*. 31, p: 11.

13. Stallings, W. **2012**. *Cryptography and Network Security Principles and Practice*, Fifth Edition, Prentice Hall.

14. Talbot, J. and Welsh, D. **2006**. *Complexity and Cryptography an Introduction*, Cambridge University Press.

15. Morkel T., Eloff J.H.P. and Olivier M.S. **2005.** An Overview of Image Steganography in Proceedings of the Fifth Annual Information Security South Africa Conference , Sandton, South Africa, June/July.

16. Nasser H. **2010.** Hiding Text Information in a Digital Image Based on Entropy Function, *The International Arab Journal of Information Technology*, 7(2), April.

17. Stefan K. and Fabien A. P. **2000**. *Information hiding techniques for steganography and digital watermarking*, Artech House computing library.

18. Farshid P. , Sitinorul H. S. A. and Shahnorbanun S. **2013**. Peak Signal-To-Noise Ratio Based on Threshold Method for Image Segmentation", *Journal of Theoretical and Applied Information Technology*, 57(2).