Journal of Al-Qadisiyah for Computer Science and Mathematics
Vol. 3 No.1 Year 2011

Page 242-253

Enhancement an Algorithm to Hide a Text into a Digital Image as a Steganography Technique

Salman Abd Kadum
Tameem Hameed Abaidah

Abstract

In this paper we enhance an algorithm to hide a text into a digital image as steganography technique. This algorithm consists of two major parts, the first, is procedure to hide a text into a digital image, and the second, is procedure to get a hide text from it, further than this algorithm consists of many parts or procedures to enhance this algorithm to complete this process. We use the Least Significant Bit method to hide the message or text, in order to keep the features and characteristics of the original image. By experiment and implement of this algorithm by visual basic 6.0, the algorithm seems work in a butter case by hiding and getting text from a digital image which is used as a carrier of this text, it means that we use two stages in this paper to accomplish it: 1) detecting that steganography has been used, and then 2) by reading the embedded message.

Keywords

Steganography techniques, stego, steganalysis, digital images, stego-image, LSB.

I. Introduction

Steganography is derived from the Greek word Steganos which means Covered or Secret, and Graphy means written or drawn. The objective of steganography is to send a message through some media known as a carrier, to a receiver, while preventing anyone else from knowing that the message exists. The carrier can be one of many different digital media, but the most common is the image. The image should not attract any attention as a carrier of a message and should compare as

close as possible to the original image by the human eye. When images are used as the carrier in steganography, they are generally manipulated by altering one or more bits of the byte that make up the pixels of the image. The (Least Significant Bit, LSB) may be used to encode the bits of the message. These LSB's can then be read by the recipient of the stego-image and put together as bytes to reproduce the hidden message, providing they have the stego key – the password for the stego-image. Steganalysis is the art of discovering a message. Breaking a steganography has been used, reading the embedded to third parties. Steganalysis methods are also used by the steganographer to determine whether the message is secure and whether the process has been successful. [1].

Steganography is used for transmitting data in a media such as image. Cryptography and steganography are different in their methods of hiding information. Cryptography scrambles a message and hides it in a carrier, so that if it is intercepted it would be generally impossible to decode. Steganography hides the very existence of the message in the carrier. When the message is hidden in the carrier a stego-carrier is formed e.g. a stego-image. If successful, it would be perceived to be as close to the original carrier or cover image by the human eye. Images are the most widespread carrier medium [2]. They are used for steganography in the following way:

The message may firstly be encrypted. The sender embeds the secret message to be sent into a graphic file [3]. This results in production of what is called the stego-image. Additional secret data may be needed in hiding process e.g. a stego key [4]. The stego-image is then transmitted to the recipient.

The recipient or extractor extracts the message from the carrier image. The message can only be extracted if there is a shared secret between the sender and the recipient. This could be the algorithm for extraction or a special parameter such as a key (the stego-key).

To make the steganographic process even more secure the message may be compressed and encrypted before it is hidden in the carrier. Figure (1) illustrate the principles of steganography where a carrier message has a message is added and put through a Stegosystem Encoder. The stego-image is then sent through the appropriate channels to Stegosystem Decoder [5].

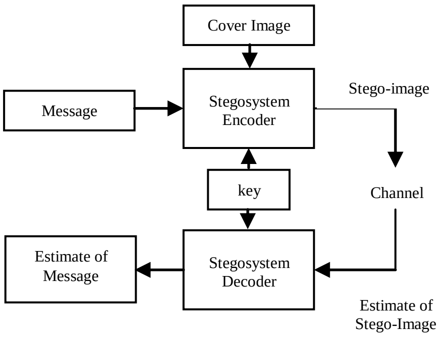


Figure (1): Steganographic System [5]

For grayscale images each pixel has a value between 0 and 255. The image is broken down into co-ordinates and pixels. The carrier image must be either the same size or larger than the message. The LSB of each pixel of the carrier is changed to the LSB if each of the message to hidden. This has the effect of hiding the message but making it appear to be the carrier. The human eye cannot detect a message or any difference to the carrier. It then has to be passed through a stego-image decoder for the hidden message to be extracted. A username and password is required at this stage.

This is where Cryptography and Steganography can be used together. When the message is compressed it takes up less space in the carrier and will minimize the amount of information to be sent.

II. Image Steganography

In LSB substitution, it is changed because this as shown in figure (2).

Image 1: The Grayscale pixel bit size is (128)

1 0 0 0 0 0 0 0

Image 2: The Grayscale pixel bit size here is (129) with the LSB changed.

1 0 0 0 0 0 0 **1**

Image 3: By changing the MSB here the bit size has changed from (128) to (0).

0 0 0 0 0 0 0 0

Figure (2): LSB and MSB Substitution.

This shows that the grayscale image would changed significantly if there were any other bit changed than the LSB. It changes more and more the closer you get to the MSB. When the LSB is changed, the pixel bit value change from 128 to 129, which is undetectable with the human eye. With the MSB changed, the pixel bit value changes from 128 to 0, which makes a significant change to the grayscale view, and change the LSB of image one to the LSB of image two for each co-ordinate or pixel, image two will be hidden in image one as illustrated in Figure (3).

LSB of Grayscale Image 1
+
LSB of Grayscale Image 2
=
Grayscale of Image 1 and Image 2

Figure (3): Image Processing Results

III. Mechanism of Steganography

When an image is used as carrier in steganography, it is generally manipulated by changing one or more bits of the byte, in our case the LSB. If it corresponds to the bit to be hidden or embedded it is left unchanged. Otherwise it is changed correspond to the hidden bit. These LSB's can then be read by the recipient of the stego-image and put together as byte to reproduce the hidden message. In a grayscale image, each pixel is either black or white and has a level between 0 and 255, as each pixel has eight bits. Steganography is carried out by changing the low order bit of a pixel, and using it to encode one bit of a character. There are two stages in the steganalysis system:

- Detecting that steganography has been used.
- Reading the embedded message.

Steganalysis is used by a steganographer in order to determine whether a message is secure and consequently whether the steganographic method has been successful. That aim of a steganalysis is to detect stego-image, find and read the embedded message, and prove that the message has been embedded to third parties. Detection involves observing relationships between combinations of cover, ,message, stego-media and steganographic tools [6]. Active interference by the steganalysis involves removing the message without altering the stego-image too much, or removing the message without considering to the stego-image appearance or structure [7]. There are two necessary conditions to be fulfilled for a secure steganographic process. The key must remain unknown or undetectable to the attacker, and the attacker should not be familiar with the cover image. If the cover image is known and it is impossible to keep it unknown from the attacker, the message could be embedded in a random way so that it is secure, as long as the key remains unknown. However, it is preferable that the cover image remains a secret to obtain maximum security.

Steganalysis techniques can be divided into five categories: *stego-only*, *known cover*, *known message*, *chosen stego,* and *chosen message*. In a stego-only attack, only the stego-image is available for steganalysis. This is similar to the *cipher text only* attack in cryptanalysis and is the weakest form of attack. In a known cover attack, both the original cover and a corresponding stego-image is available. The known message attack is when the steganalyst knows the secret message embedded in a stego-image. A chosen-stego attack (similar to a chosen *cipher text* attack in cryptanalysis) is when access to the message extraction tool is available so the attacker does not have to deduce the decoding algorithm. The most powerful attack is the chosen message attack, where the steganalyst has access to the steganography encoding tool itself and can embed and analyze messages of his own choosing.

Destroying the presence of embedded data without destroying the perceptual quality of the stego-image can be a trivial or a very difficult task depending on the steganographic method employed to embed the data. For any LSB embedding or simple bit-wise modulation schemes, destruction of the message can be performed by zeroing the entire LSB plane. For attacking non-robust steganographic methods, anti-watermarking software such as UnZign [8] or Stirmark [9] has been shown to be effective in destroying an embedded message.

IV. Discussion and Implementation

In this research we enhance an algorithm to hide a text in a digital image, we named this algorithm the Stego_algorithm, we divided it to many parts, such as algorithm of loading stego-image, algorithm of hide a text in stego-image, algorithm of convert decimal to binary, algorithm of analysis text as steganalysis in stego-image, and algorithm of convert binary to decimal. We use the Least Significant Bit, LSB technique to hide a text in this paper, because it is undetectable by the human age .

The steps of this algorithm are illustrated as below:

```
Step 1:  Open a digital image as a carrier  of message.
Step 2:   a) Hide a text into a digital image,
        b)Function to convert decimal to   binary,
Step 3:   a)Get this text from a digital image.
        b)Function to convert binary to decimal,
End.
```

The implementation of this algorithm by visual basic 6.0 illustrated below.

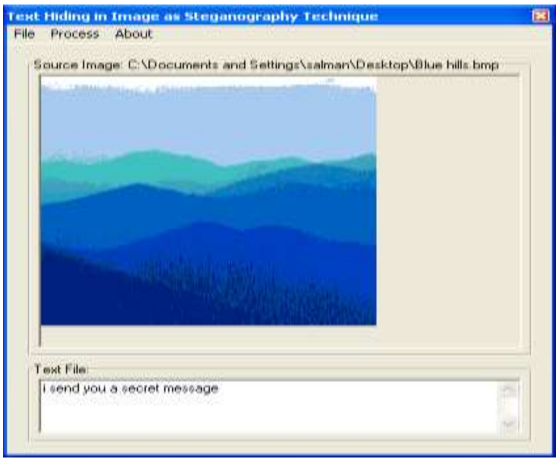
```
BIN_PIX, NEW_BIN_PIX, BIN_TXT ← String
    RGBC ←  RGBComponent
    SplitColor ←  Single
    MSLEN ←  Integer
```

The algorithm of loading stego-image is shown as fellow:

```
Open an Image as Carrier Image;

    i ← Integer, j ← Integer
    red ← Integer, green ← Integer, blue ← Integer
    pixel ← Long
    PictureName ← String
    Com.ShowOpen
        PictureName ←  Com.FileName
        If PictureName ← "" Then Exit
        Picture1.Picture ← LoadPicture(PictureName)
        Frame1.Caption ← "Source Image: " & PictureName
    End,
```

The image which is loaded is shown in the figure (4), as below:



Source Image: C:\Documents and Settings\salman\Desktop\Blue hills.bmp

Text File:

i send you a secret message

Figure (4): The carrier Image.

The algorithm of hide a text in stego-image is shown as below:

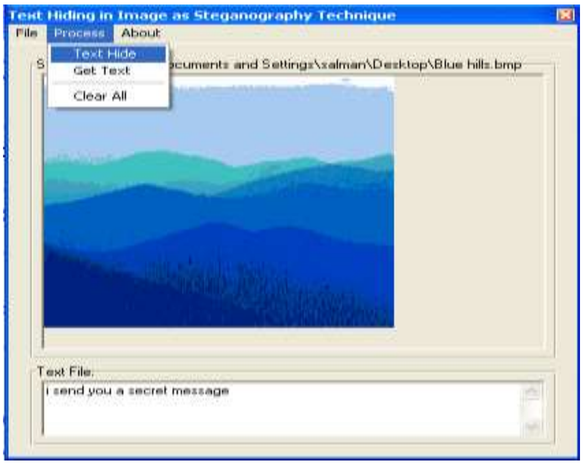
```
Hide a Text into an Image;
    If Trim(Text1) ← "" Then Exit
    Picture1.Picture ← Picture1.Image
    Picture1.Refresh
    Convert text to  binary
    For i ← 1 To Len(Text1)
    BIN_TXT ← BIN_TXT + DEC2BIN(Asc(Mid(Text1, i, 1)))
   Next
   BIN_TXT ← Trim(BIN_TXT)
   MSLEN ← Len(BIN_TXT)
   R ← Integer
    For i ← 1 To Len(BIN_TXT)
      SplitColor ← GetPixel(Picture1.hdc, i, 1)
   With RGBC
   .R ← SplitColor And &HFF
   .G ← (SplitColor \ &H100) And &HFF
   .B ← (SplitColor \ &H10000) And &HFF
      BIN_PIX ← DEC2BIN(.R)
    NEW_BIN_PIX  ←  Mid(BIN_PIX,  1,  Len(BIN_PIX)  -  1)  +
      Mid(BIN_TXT, i)
    . R ← BIN2DEC(NEW_BIN_PIX)
```

```
        SetPixel Picture1.hdc, i, 1, RGB(.R, .G, .B)
        End With
        DoEvents
        Next i
      Picture1.Refresh
      Text1 ← ""
End,
```

The algorithm of converting decimal to binary is shown as below:

```
Convert decimal to Binary using Function DEC2BIN;
Public Function DEC2BIN ← string
     ByVal x ← Single
     If x ← 0 Then
     DEC2BIN ←  "00000000"
     Else
     S ← ""
     Do While x <> 0 And Len(s) < 8
     s ← Trim(Str(x Mod 2)) + s
     x ← x \ 2
     Loop
     DEC2BIN ←  s
     End If
End Function,
```

A hide text in stego-image as shown in figure   (5) below:



ocuments and Settings\xalman\Desktop\Blue hills.bmp

Text File:
i send you a secret message

Figure (5): Hide a text into a carrier

The algorithm of analysis text as Steganalysis in stego-image is shown as below:

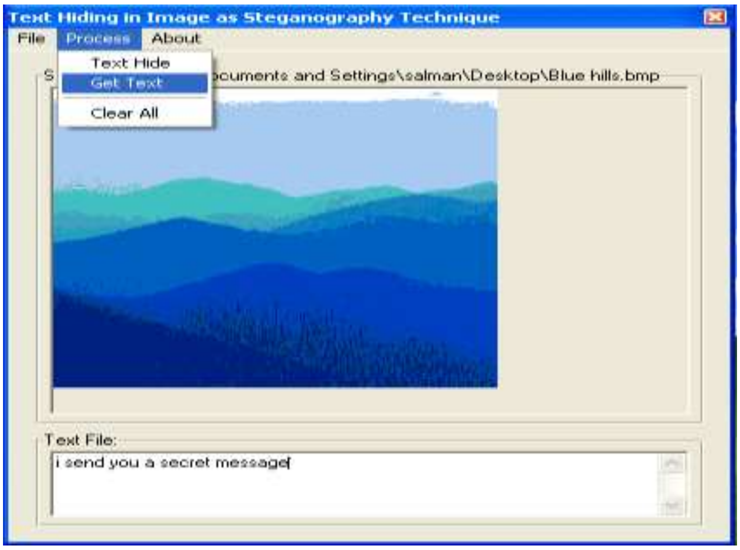
```
Get a Hide Text from an image;
    Text1 ← ""
    For i ← 1 To MSLEN
    SplitColor ← GetPixel(Picture1.hdc, i, 1)
    With RGBC
    .R ← SplitColor And &HFF
    .G ← (SplitColor \ &H100) And &HFF
    .B ← (SplitColor \ &H10000) And &HFF
    BIN_PIX ← DEC2BIN(.R)

    S ← s + Right(BIN_PIX, 1)
    End With
    DoEvents
    Next i
    Text1 ← "": no ← Len(s) \ 7
    For i ← 1 To no
    s1 ← "0" + Mid(s, 1, 7)
    s ← Mid(s)
    Text1 ← Text1 + Chr$(BIN2DEC(s1))
    Next
End,
```

The algorithm of converting binary to decimal shown as below:

```
Convert Binary to decimal using Function BIN2DEC;
Public Function BIN2DEC ← Single
    ByVal x ← String
    Dim sum, us As Integer
    If x  ←"00000000" Or x ← "" Then
      BIN2DEC ← 0
    Else
    us ← 1: sum ← 0
    x ← Left(x, Len(x) - 1)
    Do While x <> ""
      If Right(x, 1) ← "1" Then sum ← sum + us
         us ← us * 2
         x ← Left(x, Len(x) - 1)
    Loop
       BIN2DEC ← sum
  End If
End Function,
```

A Get text in stego-image as Steganalysis
is shown in figure (6) below:



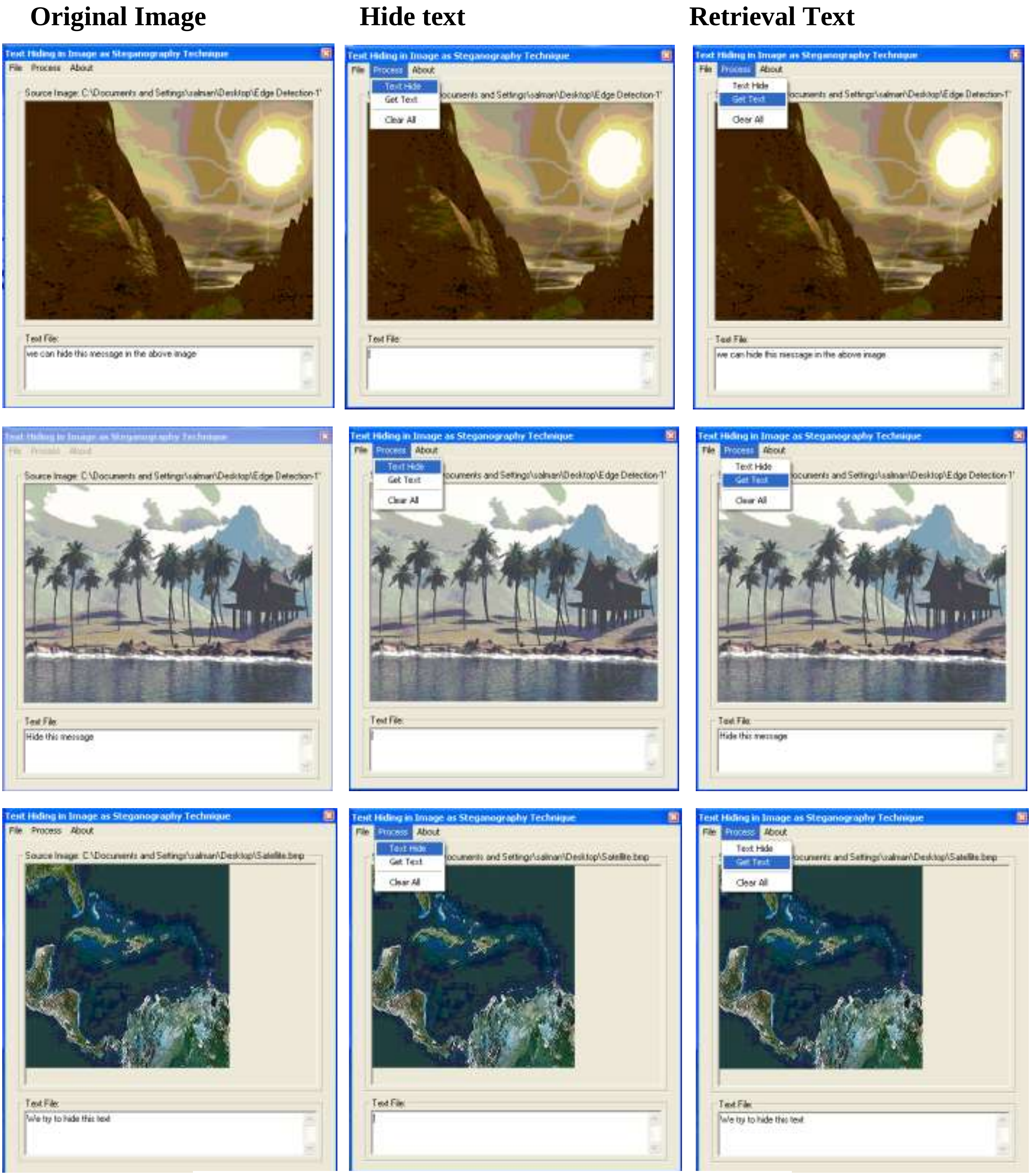
S ocuments and Settings\salman\Desktop\Blue hills.bmp

Text File:

i send you a secret message

Figure (6): Extracting a text from a carrier image.

We applied this algorithm to many images such as illustrated in the following figure:



Original Image

Hide text

Retrieval Text

Source Image: C:\Documents and Settings\salman\Desktop\Edge Detection-1'

Text File: we can hide this message in the above image

ocuments and Settings\salman\Desktop\Edge Detection-1'

Text File:

ocuments and Settings\salman\Desktop\Edge Detection-1'

Text File: we can hide this message in the above image

Source Image: C:\Documents and Settings\salman\Desktop\Edge Detection-1'

Text File: Hide this message

ocuments and Settings\salman\Desktop\Edge Detection-1'

Text File:

ocuments and Settings\salman\Desktop\Edge Detection-1'

Text File: Hide this message

Source Image: C:\Documents and Settings\salman\Desktop\Satellite.bmp

Text File: We try to hide this text

ocuments and Settings\salman\Desktop\Satellite.bmp

Text File:

ocuments and Settings\salman\Desktop\Satellite.bmp

Text File: We try to hide this text

Figure (7) application this method to many digital images.

When we applied this method to many different digital images, we get the same result to hide and retrieval a text. It means that this method works in a good way.

V. Conclusion

From the information that has been presented in this paper, the steganography technique is a very good method to secrete message by sending it through digital image, and when we applied this method to many different digital images, we get the same result to hide and retrieval a text. It means that this method works in a good way. The image we use to hide text it seems by compare as close as possible to the original image by the human eye. That means that to a great deal of advantage for those who hide secrets using steganography. And a huge disadvantage for the forensic analysts, who has the challenge of detecting and retrieving the hidden messages without destroying it;

The algorithm which is enhanced here, it's so good and sufficient to hide a text in different images we use, by implementing this algorithm in visual basic programming and it give a good result in a secrete message as a steganography technique.

References

[1] K. Curran, L. Xuelong, R. Clarke, 2005, "An Investigation into the use of the least Significant Bit Substitution Technique in digital Watermarking", International Technologies Research Group, University of Ulster, magee Campus, Northland Road, Northern Irland, UK, American Journal of Applied Science 2(3), pp: 648-654.

[2] E. Cole, 2003, "Hiding in Plain Sigt", John W. Wiley, ISBN:0-471-44449-9.

[3] A. Wstfield, and Pfitzmann, 1999, "Attacks on Steganographic System", Third International Workshop, IH'99 Dresden Germany, October Proceedings, Computer Science, 1768:61-76.

[4] J. Zollner, H. Federrath, H. Klimant, A. Pfitzmann, R. Piotraschke, A. Westfeld, G. Wicke and G. Wolf, 1998, "Modeling the Security of Steganographic System", Information Hiding, 2nd International Workshop, IH'98 Portland, Oregon, USA, Computer Science, 1525: 344-254.

[5] B. Pfitzmann, 1996, "Information Hiding Terminology", collected by Birgit Pfitzmann Information Hiding First International Workshop, Cambridge.

[6] L. M. Marvel, C. G. Boncelet, and C. T., Retter, 1998, "Reliable Blind Information Hiding for Images", Proc. Of Information Hiding Workshop, pp: 48-62.

[7] N. F. Johnson, D. Zoran, and J. Sushil, 2001, "Information Hiding, and Watermarking- attacked", Countermeasures, Kluwer.

[8] P. Wayner, 2002, "Disappearing Cryptography, Information Hiding Steganography and Watermarking", second edition, Morgan Kaufmann.

[9] D. Bret, 2002, "A detailed look at Steganographic Techniques and their use in an Open-Systems Environment", SANS Institute.

تحسين خوارزمية اخفاء النص في الصورة الرقمية في تقنية ستيكانوكرافي

سلمان عبد كاظم *تأميم حميد عبيدة*

الخلاصة

في هذا البحث تم تصميم وتحسين خوارزمية لإخفاء النص في الصورة الرقمية ضمن تقنيات الإخفاء التي تدعى بالاستيكانوكرافيٍ Steganograph . هذه الخوارزمية تتكون من جزئيين رئيسين، وهما الجزء الأول الذي يستخدم في إخفاء النص في الصورة الرقمية والجزء الثاني الذي يستخدم لاسترجاع النص منها، بالإضافة إلى عدة أجزاء اخرى في هذه الخوارزمية تكون مساعدة لانجاز البحث. بالتجربة والتنفيذ لهذه الخوارزمية تحت بيئة لغة البرمجة المرئية بيسك الإصدار 6.0، تبين ان هذه الخوارزمية تعمل في حالة ممتازة من حيث إخفاء واسترجاع النص من الصورة الرقمية، تتضمن هذه الخوارزمية مرحلتين مهمتين لانجاز البحث وهما 1) اكتشاف بان الإخفاء موجود،
2) قراءة الرسالة المخفية.