*International Journal of*
*Emerging Research in Management &Technology*
*ISSN: 2278-9359 (Volume-3, Issue-9)*

Research Article | September 2014

# Scattered Information Using Steganography

**Ali Abidalkarem Habib**
MSc.IT
Faculty of computer Science and mathematics
Department of Computer Science
University of Kufa. Iraq

**Farah Abbas Obiad**
MSc.IT
Faculty Of computer Science and mathematics
Department of Statistics and Informatics
University of AlQadisiyah, Iraq

***Abstract:***

***We have merge two methods to increase security of information the second one complements the work of the other, where first time the information will be codded using difference coding algorithm then embed the codded information in such image using LSB method with new modification in features of it, the difficulty decoding algorithm is one of the important characteristics in this paper.***

***Keyword: Information , codding ,decoding ,embedding algorithm , extracting algorithm***

## I. INTRODUCTION

Traditionally hidden information in image without encrypted it , or sometimes theembedded data encrypted then processedin such digital media cover , the encoding information rather than ciphering make hiddenoperation complex and any onecan't guess these data codded in new method , when searching in compression algorithms ,we find best one to make new data need to complex probabilities and mathematics operations to get original data with some modification in features of it to be suitable to create scattered information and generate new generation fig.1**[2].**

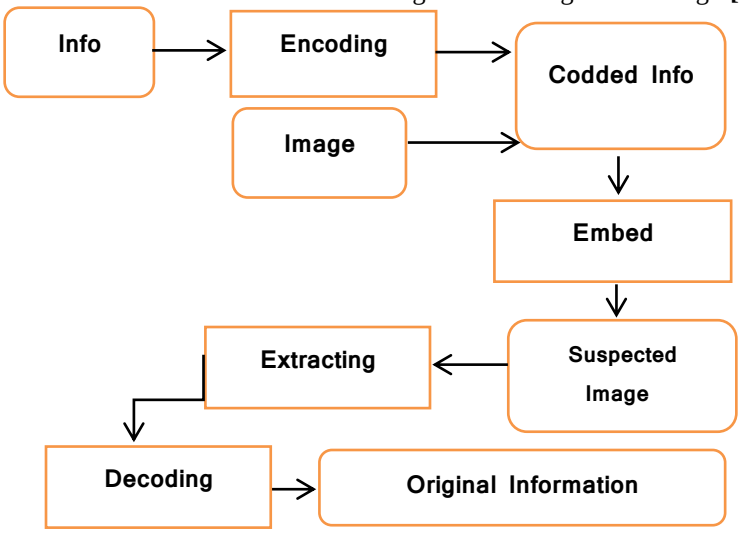


Fig. 1General Structure

## II. CREATE SCATTERED INFORMATION

We use in this section of paper , difference codding algorithm to generate scattered information**[3]** , this algorithm depend on bitwise operations as a mathematic operations , it save the first byte as it is then subtract second byte from first one and saveresult in second position and so on for other bytes .

The problem lies in the negative results because the byte represent an 8-bit unsigned integer so will complement result and put sign bit 1 for negative and 0 for positive .

**Ex:** b1=4 , b2=7 both as byte
Result= b1 subtract b2
Result =-3 Result as byte ……..error
Complement(-3 ) = 253
~ 00000011= 11111101 sign bit .

**1. Coding algorithm**

1- ByteArrayTxt=Read Text As Byte Array
2- coddedByte[0]= ByteArrayTxt[0]
3- For i =1 To (ByteArrayTxt.length-1)
4- coddedByte[i]= ByteArrayTxt[i]- ByteArrayTxt[i-1]
5- Next i

## III. EMBEDDING

Select the sequential bytes of image make themexpected and easy to get codded data so we select random positions of bytes in image , the number of these positions depend on the ( size of codded data * 8 ) then embed each bit of codded byte in LSBof byte for image**[1]**.

Ex: if coming bit (0) of codded byte then and cover byte by 254
00001101AND 11111110 = 00001100
Otherwise(1) then OR byte with 1
00001100 OR 00000001=00001101
Finally save the size of codded information in image and the positions of selected bytes using same manner.

## IV. EXTRACTING

Which means get the codded data from image, first time we have to get the size of codded data to know number of byte which used to hide codded data, the size embedded in the last bytes of image to prevent lost it**[2]**.

After get the size most be extract the positions of selected byte which embedded directly after bytes of size , now start with first byte and extract the LSB bit by right shifting to one bit then put it in byte buffer so each 8 byte of image create one byte for codded data till get all bytes .

These operation create array of bit then generate one byte at a time.

**Ex :**

First byte of image 10100101 >> 1 =01010010
Buffer=1
Second byte of image 10011110 >> 0
=01001111
Buffer =10 until complete 8 bytes and Buffer have 8 bit .

## V. DECODING

After get codded data , next and last step it generate the original data which can be understood from receiver , this step take the bytes of codded data and reverse subtract or add from one (previous decoded byte ) to second codded byte , these both operation(add , subtract) depend on specific rules**[3]**.

When the coming byte is greater than 127 then there is problem and the operation become ambiguous we can omit this complex After some experiments to understand the result of casting a negative integer which has a value in the range [-255 : -1], I got the following results:

**byte result = (byte) (-6); // result = 250**
**byte result = (byte) (-50); // result = 206**
**byte result = (byte) (-17); // result = 239**
**byte result =(byte)(-20);// result = 236**

So, provided that -256 < a < 0 , I was able to determine the result by:
result=256- a; a= codded byte .Now we getdata the real codded byte to found the original byte this determine by:
decoded byte =prev_decoded byte - result
Otherwise we can find the original byte directly by: add previous decoded byte to codded byte.
**Note: - If the codded byte = 0 so will follow first procedure.**

**2. Decoding Algorithm**

1- Coddedbyte =Get Codded byte Array
2- decodeArray[0]= Coddedbyte[0]
3- For i= 1 To Coddedbyte.length-1
4- If (Coddedbyte [i] > 127 **Or**
Coddedbyte [i]=0) Then **GoTo** :( 6 )

 

5- **Else**decodeArray[i] =
decodeArray [i - 1] + Coddedbyte[i]
6- b = 256 - Coddedbyte[i]
7- decodeArray [i] =decodeArray [i -1] – b

## VI. RESULTS AND ANALYSIS

The (table .1) contains samples of byte codes corresponding to each byte that is in itself byte also.

Table 1. Sample Results of codded bytes

| **Code** | **Original** |
|---|---|
| 104 | 104 |
| 253 | 101 |
| 7 | 108 |
| 0 | 108 |
| 3 | 111 |
| 158 | 13 |
| 253 | 10 |

possible go till 255 not more as show in fig 2.

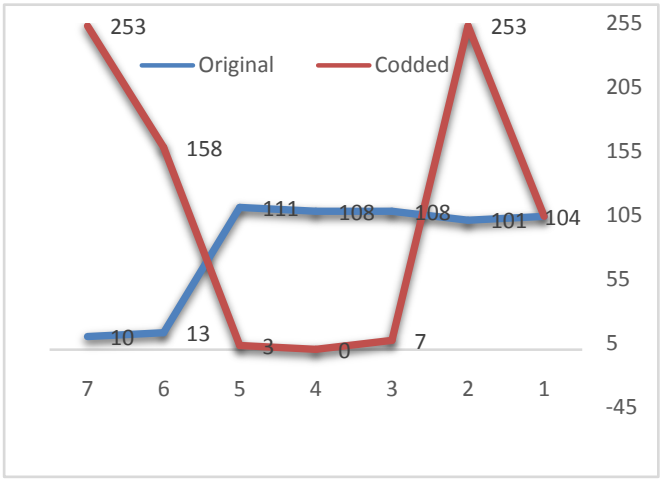


Fig .2 Sample Results of codded

the original data can't be more than 127 and codded

**3.MSE & PSNR :-**

We measure signal which represent original image to the noise or error which may be introduced in suspected image using PSNR (peak signal-to-noise ratio)**[4]**.

Before that *MSE* should be measured which means the ratio of error between original image and suspected image.as shown in table 2.

**Note**: - When the value of PSNR near or more than 80 dB then suspected image its (higher quality).

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

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

 

Table 2. PSNR for suspected images

| Original Image | Suspected Image | PSNR |
|---|---|---|
|  |  | 78.7447dB |
|  |  | 79.2499 dB |

## VII. CONCLUSION

What we conclude from this paper can be exhibited in following points.

1- Blending new method with steganography makes process of hidden information robust and need more probabilities from hackers.
2- Codding method used complement for negative bytes, and saved them as codded bytes.
3- Embedding method consumed bytes in addition to the bytes that was used to hide the coddedbytes, These bytes used to hide size of information without sending it to receiver, finally select Non-sequential bytes to hide codded information, so that no one can trace these bytes.
4- Decoding strategy have problem when get byte can't generate the original byte because we make complement for negative byte which create another byte completely different from codded byte.

## REFERENCES

[1] Nagaraj V. Dharwadkar and B. B. Amberker .,*"Secure Watermarking Scheme for Color Image Using Intensity of Pixel and LSB Substitution "* ., Journal Of Computing , Vol . No .1, PP. (1- 6 ) , December 2009.

[2] Stefan Katzenbeisser And Fabien A. P. Petitcolas ., *"Information Hiding Techniques forSteganography and Digital Watermarking "*.,Library of Congress Cataloging-in-Publication Data ., New Edition , PP. 95 – 117 .

[3] David Salomon ., *" Data CompressionThe Complete Reference"*., British Library Cataloguing in Publication Data ., Fourth Edition, PP.22 – 40 .

[4] A.Umamageswari ,M.FerniUkrit and Dr.G.R.Suresh.; *" A Survey on Security in Medical Image Communication "*, Journal of IJCA , Vol .No.3 , PP. 41- 45 , paper No .8887 ,September 2011.

[5] HediehSajedi,*" RECENT ADVANCES IN STEGANOGRAPHY"*,Published by InTechJanezaTrdine 9, 51000 Rijeka, Croatia , New Edition , 2012,PP.38 – 4

 