Volume 2, No. 6, June 2011
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

KEY BASED STEGANOGRAPHY IN A GRAY LEVEL IMAGE INVOLVING PERMUTATION AND XOR OPERATION

V.U.K.Sastry[1*], Ch.Samson[2]
[1]Department of computer Science and Engineering, SNIST
Hyderabad, AP, India
vuksastry@rediffmail.com
[2]Department of Information Technology, SNIST
Hyderabad, AP, India
samchepuri@gmail.com

Abstract: In this paper, we have developed a procedure for the steganography of a plaintext in a gray level image. This procedure is totally based on a key. The plaintext is modified by permuting with the key and by performing the XOR operation with the key. The modified plaintext obtained in the afore mentioned manner is hidden in the image by taking the help of the key. The process adopted in this investigation is found to be quite interesting as the original image has practically no change even when a long plaintext is introduced into the image.
Keywords: steganography, plaintext, permutation, modified plaintext, key, XOR operation.

INTRODUCTION

Though the study of steganography [1] had its origin several centuries back, it has gained considerable importance in recent years. In this process, the plaintext of a message is concealed either by applying a mechanism such as character marking, invisible ink etc., or by placing it in another plaintext or in an image (gray level or colored). The method of placing one plaintext in another plaintext, for hiding the former in the latter, requires a lot of overhead, while placing a plaintext of several megabytes requires a snapshot of an image. Thus steganography in an image is preferred to all the other methods available in the literature.

In the science of steganography [2-3], it is very well noticed that the strength of the steganography can be enhanced by using a key and encrypting the plaintext. Some of the ideas of the key based steganography are found in [4].

In the present paper, our objective is to develop a novel procedure for the steganography of a plaintext in a gray level image, by using a key, wherein the numbers in the key are chosen in a random manner. Here the plaintext is converted into numbers by using the EBCDIC code, and these numbers are permuted by using the key. Then the permuted plaintext is XORed with the key, and the resulting numbers are placed in the image in an appropriate manner by using the key. In this analysis, the key is used not only in modifying the plaintext but also in hiding the contents of the modified plaintext in the image under consideration. This salient feature is expected to strengthen the steganography in a remarkable manner.

In what follows we present the plan of the paper. In section 2, we introduce the development of a method for the key based steganography, and present a pair of algorithms which offer a clear insight into the procedure. Section 3 is devoted to an illustration. In section 4 we examine the strength of the steganography. Finally in section 5, we discuss the computations and conclusions.

DEVELOPMENT OF THE METHOD FOR KEY BASED STEGANOGRAPHY

Consider a plaintext, T consisting of 256 characters. On using EBCDIC code, T can be written in the form

$T = [T_{ij}]$, i=1 to 16, j= 1 to 16, where each T_{ij} is lying in [0 255].

Let us consider a key K which can be written in the form

$K= [K_{ij}]$, i= 1 to 16, j= 1 to 16.

Here each K_{ij} is an integer lying in [0 255], and all the K_{ij}'s are chosen at random.

Let the plaintext T be permuted by using the numbers in the key K. Let the permuted plaintext be denoted by P, where $P = [P_{ij}]$, i=1 to 16, j= 1 to 16.

The procedure for permutation can be explained as follows. Let K_{ij} =N. Let us suppose that N can be written in the form

$$N= 16m+n, \tag{2.1}$$

Where m and n are integers lying in the interval [0 15]. When n=0, the m^{th} row last column element of the T will be placed as the i^{th} row j^{th} column element of the permuted matrix P. On the other hand, when $n\neq 0$, the $(m+1)^{th}$ row nth column element of the T will be placed as the i^{th} row and j^{th} column element of P. For example, when N=144 we have n=0, and m=9. Then $P_{ij=}$ T(9,16). On the other hand when N=156, we have m=9 and n=12. Then $P_{ij=}$ T(10,12). This is the process of the permutation. The details of this permutation are clearly illustrated in section 3. In order to strengthen the procedure of steganography, let us write

$$P = P \oplus K. \tag{2.2}$$

Thus we get the elements of P in their modified form. Then, these new elements of the P are to be placed in a gray level image.

Let us consider a gray level image F as shown in figure. 1.

© JGRCS 2011, All Rights Reserved

Figure.1. Image of a person

This can be represented in the form

F= [F_{ij}], i=1 to 256, j= 1 to 256, where F_{ij} are the gray level values of the image. Here each F_{ij} lies in the interval [0 255]. Thus each one can be represented in terms of 8-binary bits. Now the process of hiding the modified plaintext in the image can be described as follows.

Let K_{11} = r. Let us focus our attention on the rth column of the image, i.e., j=r. Now, let us convert P_{11} into its binary form. we get a string containing 8 binary bits. Then keeping the first 6-bits of each gray level value of F_{ir}, i=1 to 4 as it is, we go on concatenating the first two bits of the binary string corresponding to P_{11} in the first row, the next two bits of P_{11} in the next row, etc., till we reach the 4^{th} row and exhaust all the 8 binary bits. Then we proceed to the column corresponding to the value of $K_{12,}$ and deal with the binary string obtained from $P_{12,}$ and carry out the concatenation process as we have mentioned earlier. This process is to be carried out for all the columns corresponding to all the 256 numbers in the key, and of course, till we exhaust all the numbers. In the modified plaintext. Here it is to be noted that the modified plaintext is accommodated in the first four rows of the image. This procedure is explained very clearly in the illustration presented in the next section. In what follows, we present an algorithm describing the method of steganography discussed so far. We have also presented an algorithm for obtaining the original plaintext from the image in which it is hidden.

Algorithm for Key Based Steganography

// Bin( ) is used to convert a decimal number into its binary form. Six( ) is used to take only the first 6 bits into consideration. Concat ( ) is utilized to concatenate a string with another string. Dec ( ) is used to convert a binary string into its decimal form.

```
1.  Read  the matrices T, K and  F
2.  // Permutation
    for i=1 to 16
    {
    for j=1 to 16
    {
      N=K(i,j);
      m=N/16;
      n=N mod 16;
    if( n=0)
     P(i,j)=T(m,16);
    else
    P(i,j)=T(m+1,n);
     }
     }

3.  P=P ⊕ K;
4.  for NI=1 to 16
    {
     u=0;
     for i=1:16
     {
     for j=1:16
      {
     r = K(i,j);

      P(i,j)=Bin(P(i,j));
      t=0;
     for s=(1+u) to (4+u)
      {
     G(s,r)=Bin(F(s,r));
     G(s,r)=Six(G(s,r));
     G(s,r)=Concat( G(s,r), (2t+1)th
                          bit and (2s)th bit of P)
      t=1;
      F(s,r)=Dec(G(s,r));
        }
        }
        }
     u=u+4;
       }
5. Write F
```

Algorithm for obtaining the original plaintext

//Extract ( ) is used to get the 7^{th} and 8^{th} bits of the binary string under consideration.

```
1.  Read  the matrices K and  F
2.  for i=1:16
     {
     for j=1:16
     {
     r = K(i,j);
     P(i,j)=0;
     s=1:4
     {
     G(s,r)=Bin(F(s,r));
     P(s,r)=Extract(G(s,r));
     P(i,j)=Concat( P(i,j), P(s,r));
     }
     }
3. P=P ⊕ K
4.//Inverse permutation
    for i=1:16
   {
   for j=1:16
   {
   N=K(i,j);
     m=N/16;
     n= N mod 16;
     if( n=0)
     T(m,16)= P(i,j);
     else
```

© JGRCS 2011, All Rights Reserved

```
T(m+1,n)= P(i,j);
}
}
}
```

5. Write P

It is worth noticing that the process involved in the present algorithm is obtained by reversing the steps in the steganography.

ILLUSTRATION OF THE STEGANOGRAPHY

Consider the plaintext given below.

"Dear father! I have visited almost all parts of India. You told me that all Indians are highly religious and they are ethical. I have seen that it is true up to a large extent. Here in this country, there are a large number of engineering colleges, and of course, a large number of liquor shops. It is really interesting. Here we find every day a strike conducted by one party or the other. The shops are broken, the buses are damaged and 144 sections are implemented very often. Many changes are coming now! " (3.1)

Let us focus our attention on the first 256 characters of the plaintext given by (3.1). Thus we have "Dear father! I have visited almost all parts of India. You told me that all Indians are highly religious and they are ethical. I have seen that it is true up to a large extent. Here in this country, there are a large number of engineering colleges, and of ," (3.2)

On adopting the EBCDIC code, (3.2) can be written in the form of a matrix T, given by

$$T = \begin{pmatrix}
196 & 85 & 81 & 99 & 40 & 86 & 81 & 163 & 88 & 85 & 99 & 79 & 40 & 201 & 40 & 88 \\
81 & 165 & 85 & 40 & 165 & 89 & 162 & 89 & 163 & 85 & 84 & 40 & 81 & 93 & 94 & 96 \\
162 & 163 & 40 & 81 & 93 & 93 & 40 & 97 & 81 & 99 & 163 & 162 & 40 & 96 & 86 & 40 \\
201 & 95 & 84 & 89 & 81 & 75 & 40 & 232 & 96 & 164 & 40 & 163 & 96 & 93 & 84 & 40 \\
94 & 85 & 40 & 163 & 88 & 81 & 163 & 40 & 81 & 93 & 93 & 40 & 201 & 95 & 84 & 89 \\
81 & 95 & 162 & 40 & 81 & 99 & 85 & 40 & 88 & 89 & 87 & 88 & 93 & 168 & 40 & 99 \\
85 & 93 & 89 & 87 & 89 & 96 & 164 & 162 & 40 & 81 & 95 & 84 & 40 & 163 & 88 & 85 \\
168 & 40 & 81 & 99 & 85 & 40 & 85 & 163 & 88 & 89 & 83 & 81 & 93 & 75 & 40 & 201 \\
40 & 88 & 81 & 165 & 85 & 40 & 162 & 85 & 85 & 95 & 40 & 163 & 88 & 81 & 163 & 40 \\
89 & 163 & 40 & 89 & 162 & 40 & 163 & 99 & 164 & 85 & 40 & 164 & 97 & 40 & 163 & 96 \\
40 & 81 & 40 & 93 & 81 & 99 & 87 & 85 & 40 & 85 & 167 & 163 & 85 & 95 & 163 & 75 \\
40 & 200 & 85 & 99 & 85 & 40 & 89 & 95 & 40 & 163 & 88 & 89 & 162 & 40 & 83 & 96 \\
164 & 95 & 163 & 99 & 168 & 107 & 40 & 163 & 88 & 85 & 99 & 85 & 40 & 81 & 99 & 85 \\
40 & 81 & 40 & 93 & 81 & 99 & 87 & 85 & 40 & 95 & 164 & 94 & 82 & 85 & 99 & 40 \\
96 & 86 & 40 & 85 & 95 & 87 & 89 & 95 & 85 & 85 & 99 & 89 & 95 & 87 & 40 & 83 \\
96 & 93 & 93 & 85 & 87 & 85 & 162 & 107 & 40 & 81 & 95 & 84 & 40 & 96 & 86 & 40
\end{pmatrix} \tag{3.3}$$

Let us take the key K in the form

$$K = \begin{pmatrix}
208 & 41 & 248 & 43 & 65 & 40 & 222 & 99 & 233 & 67 & 66 & 124 & 78 & 119 & 149 & 183 \\
12 & 50 & 10 & 105 & 226 & 204 & 80 & 102 & 197 & 17 & 104 & 150 & 51 & 83 & 137 & 175 \\
162 & 241 & 87 & 154 & 19 & 169 & 64 & 193 & 13 & 256 & 37 & 198 & 85 & 207 & 170 & 148 \\
183 & 131 & 115 & 134 & 182 & 249 & 201 & 243 & 124 & 85 & 116 & 58 & 55 & 125 & 179 & 119 \\
231 & 56 & 15 & 122 & 9 & 129 & 5 & 182 & 33 & 100 & 126 & 63 & 84 & 196 & 101 & 127 \\
106 & 93 & 181 & 61 & 120 & 77 & 178 & 4 & 89 & 36 & 94 & 195 & 153 & 199 & 123 & 246 \\
11 & 158 & 90 & 108 & 116 & 76 & 21 & 138 & 216 & 227 & 91 & 54 & 230 & 135 & 161 & 128 \\
235 & 245 & 96 & 237 & 141 & 188 & 133 & 200 & 236 & 250 & 62 & 79 & 14 & 157 & 155 & 156 \\
53 & 201 & 131 & 221 & 144 & 38 & 168 & 174 & 210 & 3 & 20 & 52 & 45 & 29 & 240 & 28 \\
130 & 179 & 187 & 177 & 147 & 242 & 103 & 70 & 16 & 111 & 255 & 185 & 167 & 98 & 173 & 239 \\
220 & 25 & 114 & 186 & 134 & 225 & 232 & 209 & 35 & 68 & 217 & 212 & 49 & 109 & 159 & 205 \\
202 & 189 & 213 & 48 & 31 & 30 & 117 & 191 & 143 & 215 & 95 & 223 & 7 & 176 & 81 & 34 \\
253 & 180 & 206 & 18 & 229 & 228 & 145 & 23 & 132 & 152 & 160 & 22 & 27 & 1 & 110 & 73 \\
26 & 88 & 244 & 125 & 55 & 190 & 165 & 42 & 164 & 71 & 218 & 74 & 247 & 211 & 243 & 86 \\
249 & 146 & 47 & 172 & 112 & 184 & 194 & 92 & 192 & 171 & 59 & 60 & 251 & 238 & 234 & 224 \\
254 & 32 & 69 & 97 & 58 & 57 & 142 & 82 & 72 & 8 & 46 & 44 & 163 & 214 & 24 & 6
\end{pmatrix} \tag{3.4}$$

The numbers in (3.4) are lying in the interval [0 255], and they are chosen at random. On carrying out the permutation process mentioned in section 2, we get

$$P = \begin{pmatrix}
85 & 81 & 107 & 163 & 94 & 97 & 85 & 89 & 85 & 40 & 85 & 81 & 95 & 85 & 162 & 89 \\
79 & 95 & 85 & 40 & 86 & 85 & 89 & 96 & 168 & 81 & 162 & 40 & 84 & 162 & 85 & 163 \\
81 & 96 & 85 & 85 & 85 & 40 & 40 & 164 & 40 & 40 & 93 & 107 & 81 & 99 & 85 & 89 \\
89 & 81 & 81 & 40 & 40 & 40 & 88 & 93 & 81 & 81 & 99 & 164 & 40 & 93 & 85 & 85 \\
89 & 232 & 40 & 89 & 88 & 40 & 40 & 40 & 162 & 87 & 75 & 84 & 40 & 99 & 89 & 40 \\
81 & 93 & 85 & 96 & 163 & 201 & 200 & 99 & 88 & 81 & 168 & 163 & 164 & 40 & 83 & 85 \\
99 & 40 & 89 & 84 & 99 & 40 & 165 & 95 & 85 & 40 & 87 & 75 & 87 & 162 & 40 & 201 \\
99 & 87 & 99 & 95 & 88 & 89 & 85 & 163 & 89 & 81 & 93 & 84 & 201 & 97 & 40 & 164 \\
81 & 88 & 81 & 82 & 40 & 93 & 85 & 95 & 81 & 81 & 40 & 89 & 40 & 81 & 83 & 40 \\
88 & 85 & 88 & 40 & 40 & 93 & 164 & 81 & 88 & 88 & 86 & 40 & 87 & 93 & 85 & 40 \\
94 & 163 & 40 & 163 & 40 & 96 & 95 & 40 & 40 & 163 & 40 & 93 & 201 & 40 & 163 & 40 \\
85 & 162 & 81 & 40 & 94 & 93 & 85 & 83 & 163 & 87 & 40 & 99 & 81 & 75 & 81 & 163 \\
40 & 99 & 81 & 165 & 95 & 85 & 89 & 162 & 165 & 99 & 96 & 89 & 84 & 196 & 163 & 81 \\
85 & 40 & 85 & 93 & 40 & 40 & 81 & 99 & 93 & 163 & 95 & 93 & 162 & 40 & 93 & 99 \\
40 & 163 & 86 & 163 & 85 & 95 & 95 & 88 & 96 & 167 & 40 & 163 & 95 & 87 & 85 & 40 \\
96 & 96 & 88 & 85 & 164 & 96 & 81 & 95 & 40 & 163 & 96 & 162 & 40 & 99 & 89 & 86
\end{pmatrix} \tag{3.5}$$

On using the XOR operation, given by (2.2), we get the modified plaintext P in the form

$$P = \begin{pmatrix}
133 & 120 & 147 & 136 & 31 & 73 & 139 & 58 & 188 & 107 & 23 & 45 & 17 & 34 & 55 & 238 \\
67 & 109 & 95 & 65 & 180 & 153 & 9 & 6 & 109 & 64 & 202 & 190 & 103 & 241 & 220 & 12 \\
243 & 145 & 2 & 207 & 70 & 129 & 104 & 101 & 37 & 40 & 120 & 173 & 4 & 172 & 255 & 205 \\
238 & 210 & 34 & 174 & 158 & 209 & 145 & 174 & 45 & 4 & 23 & 158 & 31 & 32 & 230 & 34 \\
190 & 208 & 39 & 35 & 81 & 169 & 45 & 158 & 131 & 51 & 53 & 107 & 124 & 167 & 60 & 87 \\
59 & 0 & 224 & 93 & 219 & 132 & 122 & 103 & 1 & 117 & 246 & 96 & 61 & 239 & 40 & 163 \\
104 & 182 & 3 & 56 & 23 & 100 & 176 & 213 & 141 & 203 & 12 & 125 & 177 & 37 & 137 & 73 \\
136 & 162 & 3 & 178 & 213 & 229 & 208 & 107 & 181 & 171 & 99 & 27 & 199 & 252 & 179 & 56 \\
100 & 145 & 210 & 143 & 184 & 123 & 253 & 241 & 131 & 82 & 60 & 109 & 5 & 76 & 163 & 52 \\
218 & 230 & 227 & 153 & 187 & 175 & 195 & 23 & 72 & 55 & 169 & 145 & 240 & 63 & 248 & 199 \\
130 & 186 & 90 & 25 & 174 & 129 & 183 & 249 & 11 & 231 & 241 & 137 & 248 & 69 & 60 & 229 \\
159 & 31 & 132 & 24 & 65 & 67 & 32 & 236 & 44 & 128 & 119 & 188 & 86 & 251 & 0 & 129 \\
213 & 215 & 159 & 183 & 186 & 177 & 200 & 181 & 33 & 251 & 192 & 79 & 79 & 197 & 205 & 24 \\
79 & 112 & 161 & 32 & 31 & 150 & 244 & 73 & 249 & 228 & 133 & 23 & 85 & 251 & 174 & 53 \\
209 & 49 & 121 & 15 & 37 & 231 & 157 & 4 & 160 & 12 & 19 & 159 & 164 & 185 & 191 & 200 \\
158 & 64 & 29 & 52 & 158 & 89 & 223 & 13 & 96 & 171 & 78 & 142 & 139 & 181 & 65 & 80
\end{pmatrix} \tag{3.6}$$

© JGRCS 2011, All Rights Reserved

As explained in section 2, let us now see how the steganography of the modified plaintext (3.6) can be carried out basing upon the key K given by (3.4). Here we have $K_{11=}$ 208. From (3.6), we have P_{11} =85. On converting 85 into binary form, we get a string of 8 binary bits, given by 01010101. Now we focus our attention on the 208 th column of the image. We consider the element F(1,208) and convert it into its binary form. Then we take the first 6 bits of the binary string, and concatenate with the first 2 bits (01) of the binary string corresponding to 85.Similarly the next 2 bits (01) of P_{11} are placed in F(2,208). The same procedure is carried out with the 3^{rd} and 4^{th} pairs of bits by putting them in F(3,208) and F(4,208). Now, consider K12 = 41 and $P_{12=}$ 81. By following the procedure, discussed earlier, we concatenate the strings of binary bits and obtain F(i,41), i=1 to 4.
The afore mentioned process is repeated for the rest of the columns of the image by considering the succeeding values of the key elements (in a row wise manner) and the corresponding values of the modified plaintext elements ,one after another, till we exhaust all the columns of F, and the elements of K and P. However, it is to be remembered that we carry out the steganography of the next portion of the plaintext (consisting 256 characters) by considering the next 4 rows of the image. The resulting image after carrying out the steganography of a plaintext in the first 4 rows is shown in Fig. 2

Figure 2. Image after hiding the first plaintext

To assure that the procedure applied in the key dependent steganography is working accurately, we have reversed the entire procedure of the key based steganography (by applying the algorithm given in section 2) and obtained the original plaintext.

STRENGTH OF THE STEGANOGRAPHY

In this steganography process, the key is containing 256 numbers (0 to 255). Here these numbers are arranged in a random manner. Thus the size of the key space is 256!. If the time required for processing the steganography with one value of the key is 10^{-7}sec, then the time required for the execution with all possible keys in the key space is

$$(256!)\ x10^{-7}/\ (365x24x60x60) = (256!)x3.17x10^{-15} \text{ years.}$$

As the time required is a formidable one, it is impossible to find the key with which the steganography is carried out. Thus the process of steganography is a very strong one.

COMPUTATIONS AND CONCLUSIONS

In this paper, we have developed a procedure for carrying out the steganography of a plaintext in a gray level image. In this analysis, the key is playing a prominent role in modifying the plaintext, and in the process of hiding the plaintext in the image. As the numbers in the key are in a random fashion, the portions of the plaintexts are concealed in different columns of the image in a thorough haphazard manner.
The programs required in this analysis are written in MATLAB.
The remnant of the plaintext (3.1) contains 250 characters. We make this also a string of 256 characters by appending six more blanks. This plaintext is hidden in the image, in rows 5 to 8, by introducing it, as discussed earlier. The image containing both the plaintexts is shown in Fig. 3.

Figure 3. Image containing the entire plaintext

Here it is to noted that 64 plaintexts, each consisting of 256 characters, can be concealed in the image, one after another.
Now we conclude that the modification of the plaintext which is achieved by the permutation, using the key, and by applying the XOR operation is quite significant. Further, we notice that the placement of the plaintext in the image, being guided by the key, strengthens the steganography in a remarkable manner.

REFERENCES

[1] William Stallings, Cryptography and Network Security, Principles and Practice, Fourth Edition, Pearson, 2006.
[2] Katzenbeisser, S., ed. Information Hiding Techniques for Steganography and Digital Watermarking. Boston: Artech House, 2000.
[3] Wayner, P. Disappearing Cryptography. Boston: AP Professional Books, 1996.
[4] Atul Kahate, Cryptography and network security, First Edition, TMH,

© JGRCS 2011, All Rights Reserved