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 MODULAR ARITHMETIC ADDITION

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 investigation, we have developed a procedure for the steganography of a plaintext in an image. The process involved in this is fully based upon a key. In this analysis, firstly, the plaintext is modified by permuting with the key and by carrying out the modular arithmetic addition with the key. The transformed plaintext obtained in the above manner is concealed in the image by following a procedure which is based upon the key. Here the image is divided into 256 blocks and modified plaintext containing 256 characters is hidden in 4 blocks. The strength of this steganography is found to be quite significant.
Keywords: steganography, plaintext, permutation, modified plaintext, key, modular arithmetic addition.

INTRODUCTION

The blending of the basic concepts of the image processing and the information security has led to the development of image steganography [1-3]. In a recent investigation [4], we have studied the steganography of a plaintext in a gray level image basing upon a key. In this the key is consisting of 256 numbers (0 to 255) arranged in a random manner. A given plaintext, converted into decimal numbers by using EBCDIC code, is permuted by using the key. Further this is modified by XORing with the key. The plaintext obtained in this manner is hidden in the image. In this analysis, the process of concealment is totally guided by the key and the modified plaintext under consideration is placed in different columns of the image depending upon the key. In this case, the plaintext under consideration occupies completely four consecutive rows of the image. In this process, the last two binary bits of each pixel value, in the columns, are replaced by an appropriate pair of binary bits of the numbers corresponding to the modified plaintext. As we have shown in the analysis, this process can be applied for steganography of 64 plaintexts at the most.

In the present paper, our objective is to develop a novel procedure for the steganography of a plaintext in an image. In this also we use a key containing the numbers 0 to 255 generated randomly. Here also the plaintext is permuted by using the key. But, in the present analysis, we use modular arithmetic addition between the modified plaintext and the key. Here the gray level image is divided into 256 square blocks, wherein each block is containing 256 pixels. Taking the first four numbers of the key into consideration, we identify 4 blocks in the image and hide the plaintext of length 256 characters in an appropriate manner. This process can be adopted to 64 plaintexts, at the most, by considering 4 succeeding numbers, every time, in the key.

Now we mention the plan of the paper. In section 2, we deal with the development of the method for the key based steganography. 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 arrive at the conclusions obtained from this analysis.

DEVELOPMENT OF THE METHOD FOR KEY BASED STEGANOGRAPHY

Let us consider a plaintext T, which is containing 256 characters. On using EBCDIC code, it can be represented in the form of a matrix given by $T = [T_{ij}]$, i=1 to 16, j= 1 to 16. Here all the elements the T are lying in [0 255]. Let us form a key K by selecting 256 numbers, lying in [0 255] at random. This can be represented in the form $K= [K_{ij}]$, i= 1 to 16, j= 1 to 16. Consider the numbers in the key K, one after another, and permute the plaintext T basing upon the numbers in the key. Let $P = [P_{ij}]$, i=1 to 16, j= 1 to 16 be the matrix of the plaintext obtained after permutation. Let us now explain the procedure for permutation. Let $K_{ij} = N$. As N lies in [0 255], it can be represented in the form

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

where m and n are integers lying in [0 15]. When n=0, $P_{ij}= T(m,16)$ .On the other hand, when $n\neq 0$, $P_{ij} = T(m+1, n)$. For a clear visualization of this process, we may refer to [4]. Now in order to modify the permuted plaintext in a thorough manner, so that the strength of the steganography enhances, we perform the modular arithmetic addition given by

$$P = (P + K) \bmod 256. \tag{2.2}$$

Now let us see how steganography can be carried out. Consider an image F as shown in Figure 1.

Figure 1. Gray level image of a cricketer

This image 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]. Now, we divide the image into 256 sub images, and label them as 1,2,3, …256 in a row wise manner. Here it is to be noted that each sub image consists of pixels occupying 16 rows and 16 columns. Let us now consider the key matrix. Let $K_{11} =$ M. Then let us look at the sub image labeled by M. This may be called as M^{th} block. As we already pointed out, this consists of 256 numbers (gray values) which are occupying 16 rows and 16 columns wherein, each one is a decimal number lying in the interval [0 255]. Thus each number can be represented in terms of 8-binary bits. Let us now consider the modified plaintext matrix P. On converting each number into its binary form, we get a string containing 2048(=16x16x8) binary bits. This can be seen as four sub strings wherein each sub string contains 512 bits.

Now let us focus our attention on the numbers (converted into binary bits) in M. On keeping the first 6 binary bits of the first number as it is, we concatenate the resulting substring, having 6 binary bits, with the first two binary bits of the 512 bits (corresponding to the modified plaintext). Then we proceed in a column wise manner and consider the second row first column element of the M^{th} block. We consider the first 6 binary bits of this element and concatenate the second two binary bits of the substring containing 512 bits. Similar process is carried out for all the elements in that column of M, and the same procedure is repeated in a column wise manner for all the elements in the other columns. In this process, all the 256 numbers are modified with the 512 binary bits in an appropriate manner. Then we consider the blocks corresponding to K_{12}, K_{13} and K_{14}, and concatenate pairs of binary bits corresponding to the second, third and fourth substrings respectively. However, if we have one more plaintext, we can work out in the same manner by considering the next 4 numbers in the key. This process can be carried out for 64 plaintexts at the most. This completes the process of the key based steganography.

In what follows, we present the algorithm for the key based the steganograpy. We also mention the algorithm for obtaining the original plaintext hidden in the image by following the reverse process.

Algorithm for Key based Steganography

// NT is the number of plaintexts. 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. The Dec( ) is used to convert a binary string into its decimal form.

```
1.  Read  T, K, F, NT
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) mod 256.

       //Process of hiding the modified plaintext in the image
4.  for NI=1 to NT
    {
    r=0;
    for  i=1 to 16
    {
    for  j= 1 to 16
    {
    r=r+1;
    H(r)=K(i,j);
    }
    }
    r=0;
    for  i=1 to 64
    {
    for j= 1 to 4
    {
    r=r+1;
    L(i,j)=H(r);
    }
    }
    for NJ=1 to 4
    {
    N=L(NI, NJ);
    m=N/16;
    n=mod(N,16);

    for i=1 to 16
    {
    for j=1 to 16
    {
     if (n=0)
    M(i,j)=F(16(m-1)+i,240+j);
    else
    M(i,j)=F(16m+i,(n-1)16+j);

    M(i,j)= Bin(M(i,j));
    M(i,j)=Six(M(i,j));
    }
    }
```



```
    S=0;
    for i=1 to 4
    {
    for j=1 to 16
    {
    P(i,j)=Bin(P(i,j));
    S=Concat(S,P(i,j);
    }
    }
    t=0; u=1;
    for i=1 to 16
    {
    for j=1 to 16
    {
    Q(i,j)=(2t+1)th bit to 2uth bit of S;
    t=t+1;
    u=u+1;
    }
    }
    Q=Transpose(Q);
    for i=1 to 16
    {
    for j=1 to 16
    {
    F(i,j)=Concat (M(i,j),Q(i,j));
    F(i,j) = Dec(F(i,j));
    }
    }
5. Write F
```

ALGORITHM FOR OBTAINING THE ORIGINAL PLAINTEXT

// Extract ( ) is used to obtain the 7th and 8th bits of the binary string under consideration.

```
1. Read  the matrices K , F , NT
2. for NI=1 to NT
     {
    r=0;
    for i=1 to 16
    {
    for j= 1 to 16
    {
    r=r+1;
    H(r)=K(i,j);
    }
    }
    r=0;
    for i=1 to 64
    {
    for j= 1 to 4
    {
    r=r+1;
    L(i,j)=H(r);
    }
    }
    for NJ=1 to 4
    {
    N=L(NI, NJ);
    m=N/16;
    n=mod(N,16);
    for i=1 to 16
    {
    for j=1 to 16
    {
     if (n=0)
    M(i,j)=F(16(m-1)+i,240+j);
    else
    M(i,j)=F(16m+i,(n-1)16+j);

    M(i,j)= Bin(M(i,j));
    M(i,j)=Extract(M(i,j));
    }
    }
    E(NJ)=0;
    for i=1 to 16
    {
    for j=1 to 16
    {
    E(NJ)=Concat(E(NJ),M(j,i));
    }
    }
    v=0;
    for i=(1+v) to (4+v)
    {
    for j=1 to 16
    {
    P(i,j)=Dec(Eight(E(1));
    v=v+4;
    }
    }
3. P = (P - K) mod 256.
4. //Inverse permutation
   for i=1:16
     {
    for j=1to16
    {
    N=K(i,j);
    m=N/16;
    n=N mod16;
     if (n=0)
    T(m,16)= P(i,j);
    else
    T(m+1,n)= P(i,j);
    }
    }
    }
5. Write T
```

This algorithm is written by reversing the process in the preceding algorithm.

ILLUSTRATION OF THE STEGANOGRAPHY

Consider the plaintext given below.

"Dear brothers, the government says that we are terrorists. After all we have become like this on account of poverty and many atrocities done by the society to us. We all know very well how floods affected our fertile lands, and how our crop was totally ruined. See the government is not able to come to our rescue; the scientists are not able to help us in controlling floods. See the misfortunes. Our country is very keen about space research. But they do not try to control the floods in any way." (3.1)

Let us consider the first 256 characters of the plaintext given by (3.1). This is given by "Dear brothers, the government says that we are terrorists. After all we have become like this on account of poverty and many atrocities done by the society to us. We all know very well how floods affected our fertile lands, and how our crop was totally rui” (3.2)

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

$$T=\begin{pmatrix}
196 & 85 & 81 & 99 & 40 & 82 & 99 & 96 & 163 & 88 & 85 & 99 & 162 & 107 & 40 & 163 \\
88 & 85 & 40 & 87 & 96 & 165 & 85 & 99 & 95 & 94 & 85 & 95 & 163 & 40 & 162 & 81 \\
168 & 162 & 40 & 163 & 88 & 81 & 163 & 40 & 166 & 85 & 40 & 81 & 99 & 85 & 40 & 163 \\
85 & 99 & 99 & 96 & 99 & 89 & 162 & 163 & 162 & 75 & 40 & 193 & 86 & 163 & 85 & 99 \\
40 & 81 & 93 & 93 & 40 & 166 & 85 & 40 & 88 & 81 & 165 & 85 & 40 & 82 & 85 & 83 \\
96 & 94 & 85 & 40 & 93 & 89 & 92 & 85 & 40 & 163 & 88 & 89 & 162 & 40 & 96 & 95 \\
40 & 81 & 83 & 83 & 96 & 164 & 95 & 163 & 40 & 96 & 86 & 40 & 97 & 96 & 165 & 85 \\
99 & 163 & 168 & 40 & 81 & 95 & 84 & 40 & 94 & 81 & 95 & 168 & 40 & 81 & 163 & 99 \\
96 & 83 & 89 & 163 & 89 & 85 & 162 & 40 & 84 & 96 & 95 & 85 & 40 & 82 & 168 & 40 \\
163 & 88 & 85 & 40 & 162 & 96 & 83 & 89 & 85 & 163 & 168 & 40 & 163 & 96 & 40 & 164 \\
162 & 75 & 40 & 230 & 85 & 40 & 81 & 93 & 93 & 40 & 92 & 95 & 96 & 166 & 40 & 165 \\
85 & 99 & 168 & 40 & 166 & 85 & 93 & 93 & 40 & 88 & 96 & 166 & 40 & 86 & 93 & 96 \\
96 & 84 & 162 & 40 & 81 & 86 & 86 & 85 & 83 & 163 & 85 & 84 & 40 & 96 & 164 & 99
\end{pmatrix} \quad (3.3)$$

Let us choose the key matrix K in the form ,

$$K=\begin{pmatrix}
104 & 69 & 203 & 164 & 241 & 192 & 92 & 231 & 28 & 97 & 5 & 16 & 215 & 29 & 137 & 126 \\
48 & 56 & 255 & 30 & 70 & 103 & 180 & 113 & 66 & 21 & 72 & 176 & 109 & 83 & 36 & 73 \\
37 & 195 & 163 & 101 & 111 & 22 & 102 & 34 & 112 & 114 & 240 & 64 & 188 & 143 & 145 & 207 \\
183 & 131 & 115 & 134 & 182 & 249 & 201 & 243 & 124 & 85 & 116 & 58 & 55 & 125 & 179 & 119 \\
253 & 236 & 214 & 61 & 44 & 68 & 171 & 120 & 202 & 212 & 135 & 251 & 86 & 91 & 12 & 165 \\
75 & 2 & 67 & 150 & 155 & 130 & 98 & 118 & 223 & 117 & 166 & 197 & 39 & 154 & 18 & 1 \\
148 & 234 & 158 & 49 & 54 & 50 & 245 & 254 & 144 & 239 & 59 & 221 & 90 & 15 & 136 & 199 \\
141 & 139 & 71 & 200 & 220 & 242 & 157 & 105 & 161 & 162 & 94 & 78 & 81 & 211 & 230 & 95 \\
156 & 170 & 178 & 63 & 24 & 88 & 227 & 186 & 241 & 8 & 151 & 4 & 108 & 190 & 198 & 23 \\
60 & 123 & 14 & 129 & 159 & 248 & 210 & 42 & 153 & 52 & 7 & 76 & 238 & 19 & 96 & 142 \\
132 & 233 & 110 & 228 & 256 & 38 & 62 & 219 & 237 & 167 & 205 & 40 & 82 & 106 & 184 & 6 \\
217 & 93 & 191 & 177 & 33 & 138 & 51 & 149 & 13 & 79 & 100 & 187 & 122 & 128 & 10 & 218 \\
35 & 107 & 244 & 9 & 247 & 27 & 45 & 43 & 174 & 20 & 160 & 175 & 3 & 194 & 41 & 213 \\
80 & 146 & 173 & 172 & 216 & 89 & 57 & 99 & 17 & 11 & 65 & 74 & 225 & 53 & 209 & 87 \\
31 & 26 & 193 & 224 & 222 & 181 & 152 & 246 & 226 & 232 & 47 & 147 & 25 & 121 & 185 & 127 \\
208 & 204 & 252 & 206 & 84 & 169 & 192 & 229 & 189 & 168 & 250 & 77 & 46 & 140 & 133 & 235
\end{pmatrix} \quad (3.4)$$

We now adopt the process of permutation mentioned in section 2. Before we go ahead with this process, let us consider some key numbers as examples. In the key given by (3.4), we have K(1,1)=104. Here we get m=6, and n=8. Thus we have P(1,1) =T(7,8). Now we consider K(2,1)=48. For this m=3 and n=0.Thus we get P(2,1)=T(3,16).

By considering the other elements and applying the afore mentioned procedure, we get all the other elements of the permuted matrix P. Thus we have

$$P=\begin{pmatrix}
163 & 40 & 85 & 230 & 40 & 81 & 89 & 166 & 95 & 40 & 40 & 163 & 93 & 163 & 84 & 81 \\
163 & 163 & 164 & 40 & 166 & 95 & 40 & 99 & 81 & 96 & 40 & 165 & 97 & 85 & 163 & 88 \\
88 & 162 & 40 & 96 & 165 & 165 & 164 & 162 & 85 & 163 & 97 & 99 & 166 & 168 & 163 & 164 \\
93 & 89 & 168 & 85 & 85 & 81 & 83 & 81 & 168 & 93 & 40 & 75 & 162 & 40 & 168 & 84 \\
40 & 40 & 89 & 86 & 81 & 93 & 92 & 40 & 163 & 99 & 162 & 93 & 89 & 88 & 99 & 85 \\
165 & 85 & 93 & 96 & 168 & 83 & 81 & 95 & 107 & 81 & 40 & 81 & 163 & 163 & 85 & 196 \\
40 & 164 & 96 & 85 & 89 & 99 & 40 & 99 & 40 & 96 & 40 & 84 & 163 & 40 & 40 & 86 \\
40 & 95 & 85 & 85 & 95 & 166 & 163 & 40 & 162 & 75 & 40 & 82 & 96 & 85 & 96 & 96 \\
40 & 40 & 99 & 85 & 99 & 85 & 84 & 88 & 40 & 96 & 83 & 99 & 40 & 86 & 86 & 85 \\
193 & 95 & 107 & 96 & 40 & 163 & 86 & 85 & 85 & 96 & 99 & 85 & 99 & 40 & 95 & 82 \\
163 & 96 & 96 & 40 & 89 & 81 & 163 & 81 & 83 & 81 & 40 & 40 & 94 & 96 & 93 & 82 \\
40 & 162 & 93 & 85 & 168 & 96 & 99 & 162 & 162 & 85 & 83 & 96 & 81 & 99 & 88 & 93 \\
40 & 86 & 162 & 163 & 96 & 85 & 99 & 40 & 166 & 87 & 164 & 40 & 81 & 84 & 166 & 163 \\
83 & 88 & 96 & 95 & 85 & 40 & 162 & 83 & 88 & 85 & 40 & 81 & 81 & 99 & 40 & 92 \\
162 & 94 & 96 & 40 & 162 & 166 & 89 & 163 & 95 & 40 & 40 & 85 & 95 & 94 & 40 & 163 \\
99 & 84 & 168 & 96 & 40 & 93 & 96 & 88 & 40 & 93 & 93 & 40 & 85 & 85 & 89 & 99
\end{pmatrix} \quad (3.5)$$

On using (2.2), we get the modified plaintext P in the form,

$$P=\begin{pmatrix}
11 & 109 & 32 & 138 & 236 & 113 & 181 & 141 & 123 & 137 & 45 & 179 & 52 & 192 & 221 & 207 \\
211 & 219 & 163 & 70 & 236 & 198 & 220 & 212 & 147 & 117 & 112 & 85 & 206 & 168 & 199 & 161 \\
125 & 101 & 203 & 197 & 20 & 187 & 10 & 196 & 197 & 21 & 81 & 163 & 98 & 55 & 52 & 115 \\
20 & 220 & 27 & 219 & 11 & 74 & 28 & 68 & 36 & 178 & 156 & 133 & 217 & 165 & 91 & 203 \\
37 & 20 & 47 & 147 & 125 & 161 & 7 & 160 & 109 & 55 & 41 & 88 & 175 & 179 & 111 & 250 \\
240 & 87 & 160 & 246 & 67 & 213 & 179 & 213 & 74 & 198 & 206 & 22 & 202 & 61 & 103 & 197 \\
188 & 142 & 254 & 134 & 143 & 149 & 29 & 97 & 184 & 79 & 99 & 49 & 253 & 55 & 176 & 29 \\
181 & 234 & 156 & 29 & 59 & 152 & 64 & 145 & 67 & 237 & 134 & 160 & 177 & 40 & 70 & 191 \\
196 & 210 & 21 & 148 & 123 & 173 & 55 & 18 & 25 & 104 & 234 & 103 & 148 & 20 & 28 & 108 \\
253 & 218 & 121 & 225 & 199 & 155 & 40 & 127 & 238 & 148 & 106 & 161 & 81 & 59 & 191 & 224 \\
39 & 73 & 206 & 12 & 89 & 119 & 225 & 44 & 64 & 248 & 245 & 80 & 176 & 202 & 21 & 88 \\
1 & 255 & 28 & 6 & 201 & 234 & 150 & 55 & 175 & 164 & 183 & 27 & 203 & 227 & 98 & 55 \\
75 & 193 & 150 & 172 & 87 & 112 & 144 & 83 & 84 & 107 & 68 & 215 & 84 & 22 & 207 & 120 \\
163 & 234 & 13 & 11 & 45 & 129 & 219 & 182 & 105 & 96 & 105 & 155 & 50 & 152 & 249 & 179 \\
193 & 120 & 33 & 8 & 128 & 91 & 241 & 153 & 65 & 16 & 87 & 232 & 120 & 215 & 225 & 34 \\
51 & 32 & 164 & 46 & 124 & 6 & 32 & 61 & 229 & 5 & 87 & 117 & 131 & 225 & 222 & 78
\end{pmatrix} \quad (3.6)$$

On applying the algorithm of the key based steganography given in section 2, we get the image given in Fig. 2.

Figure.2. Image of the person after hiding the first plaintext

Figure.3. Image of the person after hiding the entire plaintext

To assure that the procedure applied in the key based steganography is correct, we have applied the algorithm for obtaining the original plaintext mentioned in section 2, and got back the original plaintext.

STRENGTH OF THE STEGANOGRAPHY

In this steganography process, the key is containing the numbers 0 to 255(256 numbers), which 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 key value is 10^{-7}sec, then the total time required for the execution with all possible keys in the key space is
$(256!) \times 10^{-7} / (365 \times 24 \times 60 \times 60) = (256!) \times 3.17 \times 10^{-15}$ years.
As this time is a formidable one, it is impossible to find the key with which the steganography is carried out. Thus the strength of the steganography is very significant.

COMPUTATIONS AND CONCLUSIONS

In this paper, we have devoted our attention to the study of the steganography of a plaintext in a gray level image. In this investigation, the key is playing a predominant role not only in modifying the plaintext but also in hiding it in the image. In this analysis, basing upon the numbers in the key, portions of the image (blocks) are used to conceal the plaintext. The programs required in the process of the steganography are written in MATLAB.
The remaining portion of the plaintext (3.1) contains 242 characters. Thus we have added 14 blanks to make it a full block containing 256 characters. On performing the steganography of the entire plaintext (3.1). The resulting image is shown in Figure 3. In this analysis, it is interesting to note that the gray level image does not have any change even though we went on hiding more number of plaintexts.

Here hiding different plaintexts in different blocks which are located in different positions of the image is the key factor for the security of information. In this analysis, we find that we can hide 64 plaintexts at the most in the entire image.

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] VUK Sastry, Ch.Samson," Key Dependent Steganography involving permutation and XOR Operation ", Journal of Global Research in Computer Science(JGRCS).

AUTHORS

Dr. V. U. K. Sastry is presently working as Professor in the Dept. of Computer Science &Engineering (CSE), Director (SCSI), Dean (R & D), SreeNidhi Institute of Science and Technology (SNIST), Hyderabad, India. He was Formerly Professor in IIT, Kharagpur, India and worked in IIT, Kharagpur during 1963 – 1998. He guided 12 PhDs, and published more than 40 research papers in various international journals. His research interests are Network Security & Cryptography, Image Processing, Data Mining and Genetic Algorithms.

Ch. Samson obtained his Diploma from Govt Polytechnic, Hyderabad in1994, B. E. from Osmania University in 1998 & M. E from SRTM University in 2000. He is pursuing Ph.D. from JNTUH since 2009. He is currently working as Associate Professor in the Dept. of Information Technology (IT), SNIST since June 2005.