Indian Journal of Science and Technology, Vol 8(22), DOI: 10.17485/ijst/2015/v8i22/68626, September 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Novel Approach towards Development of Hybrid Image Steganography using DNA Sequences

Sudipta Roy[1]*, Siddhartha Sadhukhan[1], Shayak Sadhu[1] and S. K. Bandyopadhyay[2]

[1]Department of MCA, Academy of Technology, Adisaptagram, Aedconagar, Hooghly - 712121, West Bengal, India; sudiptaroy01@yahoo.com, siddhartha.sadhukhan.2014@gmail.com, shayakchemistry@gmail.com
[2]Department of Computer Science and Engineering, University of Calcutta, 92 A.P.C. Road, Kolkata - 700009, West Bengal, India; skb1@vsnl.com

Abstract

Objective: For the purpose of Data Communication Steganography is a technique which is combination of both the science and art by which we may hide information inside other covered media. **Method:** Here we use 3-layers steganographic data encryption, in the first stage data are encoded based upon DNA-Sequence and then the stego-key is added with individual byte then the byte streams are encoded within the image. First encrypt the data is more secure than encoding the raw data and authentication values which provide access only to authorized persons. **Finding:** Only Steganography is not enough secure for the present scenario. Simple Steganography is very much vulnerable in front of attack, but if we apply some encryption on the data itself and then use Steganography then that will be more secure than the use raw data in Steganography. Application: It will cover all the fields where we need Data Security and it covers a zone of data security where we need to hide the existence of data.

Keywords: 3-Layer Encryption, DNA-based Encryption, Image Steganography, Steganography

1. Introduction

Internet is rising day by day and it takes an important factor in digital communication. Different kind of Cryptography techniques was developed in the sense of data security. But every time this is not enough. Sometime we have to hide the existence of the message. Steganography allows us to make invisible communication. Because here we can encode the message in any other covers media. Steganography is the art of concealing information in ways that prevents the detection of hidden messages. Before applying Steganography on raw data we use an encryption on data itself then allow that encrypted data to be encode in image. Here we use DNA based encryption. DNA sequence maintains some properties which we used to hide data. Information store in DNA as a code made up of four chemical bases: adenine (A), guanine (G), cytosine (C), thymine (T). There are four chemical bases A, C, G, and T that can be encoded by 2 bit. They can be represented as 00,01,10,11 respectively. The resultant data is added with stego-key which gives a second level encryption. Then the encrypted data is encoded within the image pixels.

2. Proposed Method

Any image can be viewed as a 2-dimentaional matrix (m x n) where width of the image is m and height of the image is n here each cell is call a pixel i.e. total number of pixel is (m*n). In 32-bit color scale image each pixel has a combination of 3 fields (Red, Green, and Blue) and each field may have the value in between 0 to 255 that is 8 bit or 1 byte for each fields. Now if we change the Last One Bit (LSB) or last two bits then the color variation will not be

*Author for correspondence

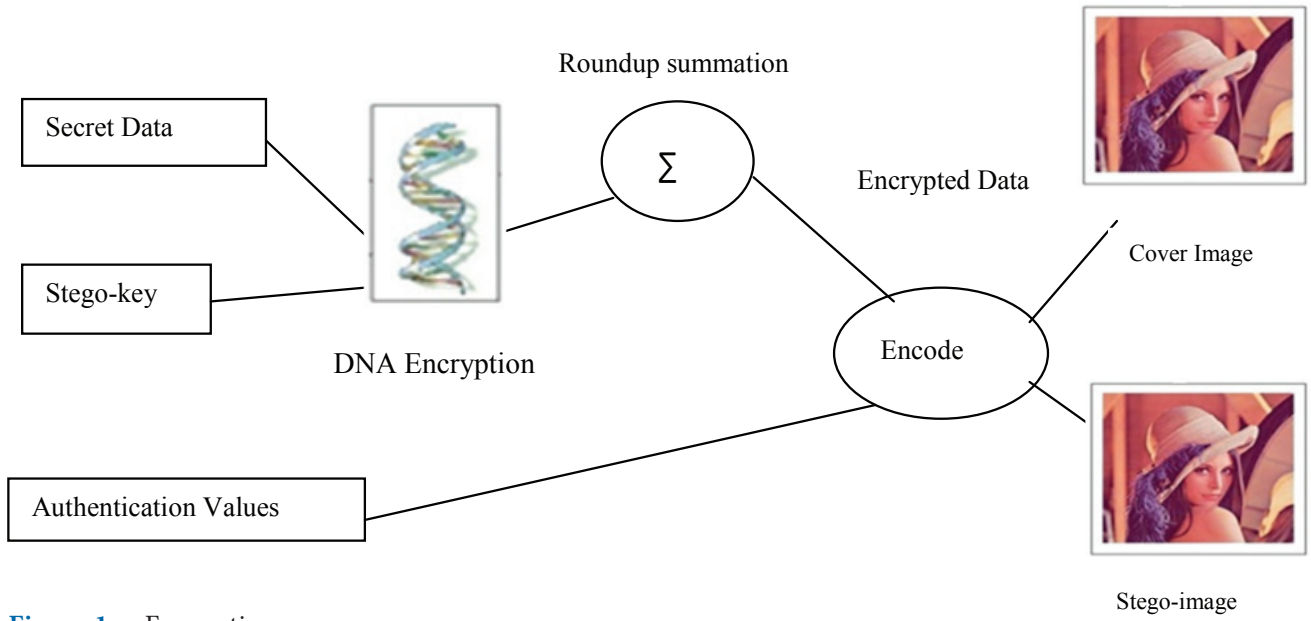

Figure 1. Encryption.

visible to human eye but these 2 bits that is 6 bits in each pixel provide to encode our data.

After encryption the original image and the encrypted image will be same for human eye. Insertion of raw data into the image will give a better scope for the hackers, but if we encrypt the data by some algorithm will give a challenge to the hackers. This paper describes an algorithm for data hiding inside a DNA sequence. Encryption is a process by which we encode the data inside the image. Decryption is the reverse process of encryption that is extraction of bits from image file.

Figure 1 shows that in fast phase a secrete data is encrypted in a DNA sequence and then second phase the encrypted data is encode in a cover image and the

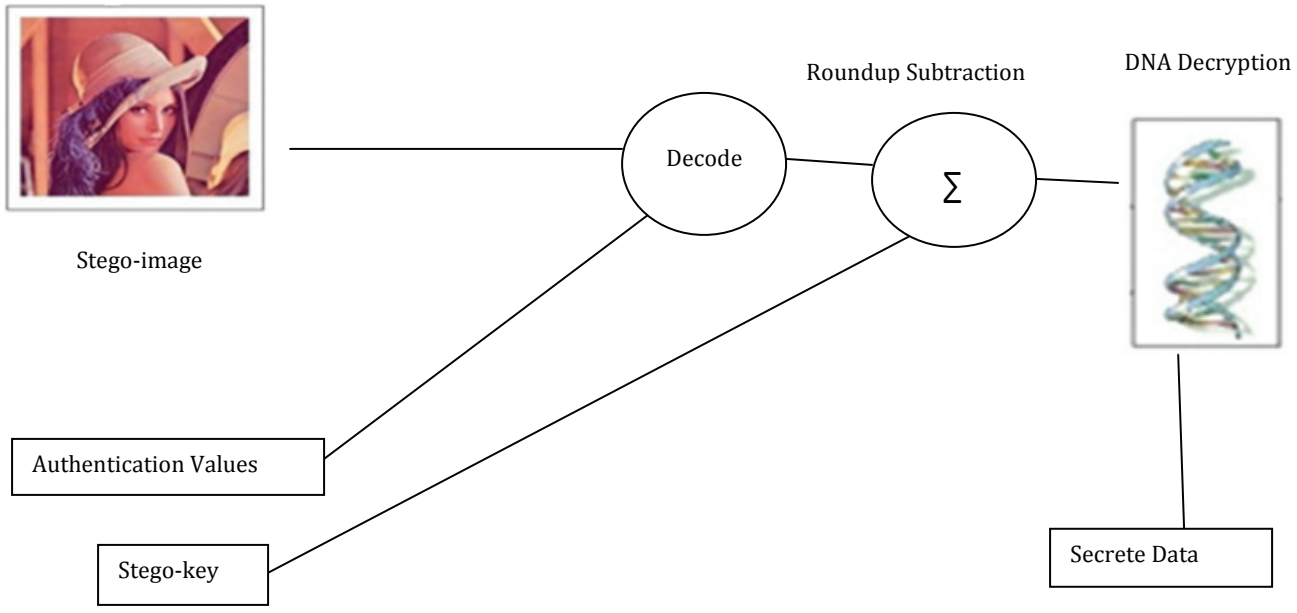

Figure 2. Decryption.

authentication values are need in second phase. The data is encoded with the stego-key inside DNA sequences. Then this encrypted data again will be encrypted with the same stego-key. Then the final data will be encoded inside the image file along with the authentication key and the encrypted image will be called stego-image.

Figure 2 shows the data is extracted from stego-image using proper authentication value and then decrypt the extracted bit using stego-key by which we get the original DNA sequence and then from the sequence get the original data. In the decryption process we have a stego-image and valid authentication key and a stego-key. Valid Authentication key will give access to actual decryption module otherwise return with failure. In decryption module DNA-encrypted bits are extracted from the image and using valid stego-key encrypted bits are decoded into actual raw bits. If the stego-key is wrong then decrypted module generate a garbage bit stream.

The stego-image must be formed with raw pixel values, because almost every image compression algorithms are loss, so if you apply such algorithms then the pixel values will be altered and create a corrupted stego-image. We have to avoid image type like 'jpg', 'jpeg' and prefer image type like 'bmp'.

As early I mention that it gives 3-lavel of security and an authenticity mechanism.

Three levels are:

- Data Hiding in DNA Sequence or Data Retrieval from DNA Sequence.
- Roundup Summation or Subtraction.
- Encoding The Byte Streams in Pixel or Decode the Byte Streams From pixels.

DNA consist of four chemical bases namely A, C, G, T. Coded them as - A (00), C (01), G (10), T (11).

3. Authentication

The authenticity can be provided by the following way:

- At the time of encryption:

  Step 1: Take 3 integer values within 0 to 255.
  Step 2: Consider a pixel which is in the position {(width/3), (height/3)} where width stands for width of the image and height stands for height of the image.
  Step 3: Replaced the RGB of the chosen pixel with the inputted 3 integer.
  Step 4: End.

- At the time of decryption:

  Step 1: Take 3 integer values within 0 to 255.
  Step 2: Select the pixel which is in the position {(width/3), (height/3)} where width stands for width of the image and height stands for height of the image.
  Step 3: Perform a matching operation with the red, green, blue part of the selected pixel with the inputted 3 integer.
  Step 4: If matched then perform decryption operation otherwise show a message and ignore decryption operation.
  Step 5: End.

Remember that, ignore the pixel which is in {(width/3), (height/3)} position for both the encoding and decoding purpose.

3.1 Encryption Process

New approach of Insertion method is used to encrypt the data by DNA sequence.

Do the following:

Step 1: Calculate the total number of byte and multiply with 8, which give total number of bits.
Step 2: Replicate the stego-key and store in a character array until it contain the number of character equal to the (number of bits in the secrete message/4).
Step 3: Read 2 character from character array. Calculate the ASCII values of those characters and represent in their binary form, and concatenate these two binary strings.
Step 4: Read the binary string left to right and place in an array say tem, except those blocks of the array tem whose index value mod 3 gives 0.
Step 5: Read all byte, but one at a time and represent in a binary string.
Step 6: Read all bits left to right and placed them in the blocks of array tem whose index value mod 3 gives 0.
Steps 7: Separate the tem array by each 8 bit and create 3 byte.
Step 8: Add these byte in a byte array.
Step 9: End.

Now do the following for Round up summation:

Step 1: Read each byte in btd.
Step 2: btd: = btd+key
Step 3: If btd>127 then

tem: = btd - 128
btd: = tem - 128
End if
Step 4: If btd = 128 then keep it as 128
Else If btd<0 then
btd: = btd*(-1)
x: = x + 128
End if,
Step 5: Store the value of btd in the corresponding to byte index
Step 6: End

Second step is complete. Now in third step the data are encoded inside the image pixel:
Step 1: Read the each pixel of the cover image one at a time.
Step 2: Get the color code of that pixel that is the value of red, green, blue.
Step 3: Represent data byte in their binary form.
Step 4: Represent each red, green, blue in there binary form and then replace last
2-bit (LSB and previous one of LSB) with the data bit of each red, green, blue.
Step 5: Reform a bitmap type image with the new pixels.
Step 6: End.

3.2 Decryption Process

We do following to extract the bits from the stego-image:
Step 1: Read the each pixel of the cover image one at a time.
Step 2: Get the color code of that pixel that is the value of red, green, blue.
Step 4: Represent each red, green, blue in there binary form and then receive last
2-bit (LSB and previous one of LSB) of each red, green, blue.
Step 5: Create a byte array with each 8 retrieval bit.
Step 6: End.

Now do the following for Roundup subtraction:
Step 1: Read each byte in btd.
Step 2: btd: = btd-key
Step 3: If btd < -128 then
Tem: = (-1*btd)-129
btd = 127 - tem
End if
Step 4: Store the value of btd in the corresponding to byte index
Step 5: End
Now decode the data bits from the DNA-sequence:
Do the following:
Step 1: Represent the bytes in their binary form.
Step 2: Create an array by inserting bits of each byte.
Step 3: Retrieve the bits from the array those index mod 3 produce 0.
Step 4: Create byte array from the retrieved bits.
Step 5: Construct the file from the new byte array.
Step 6: End.

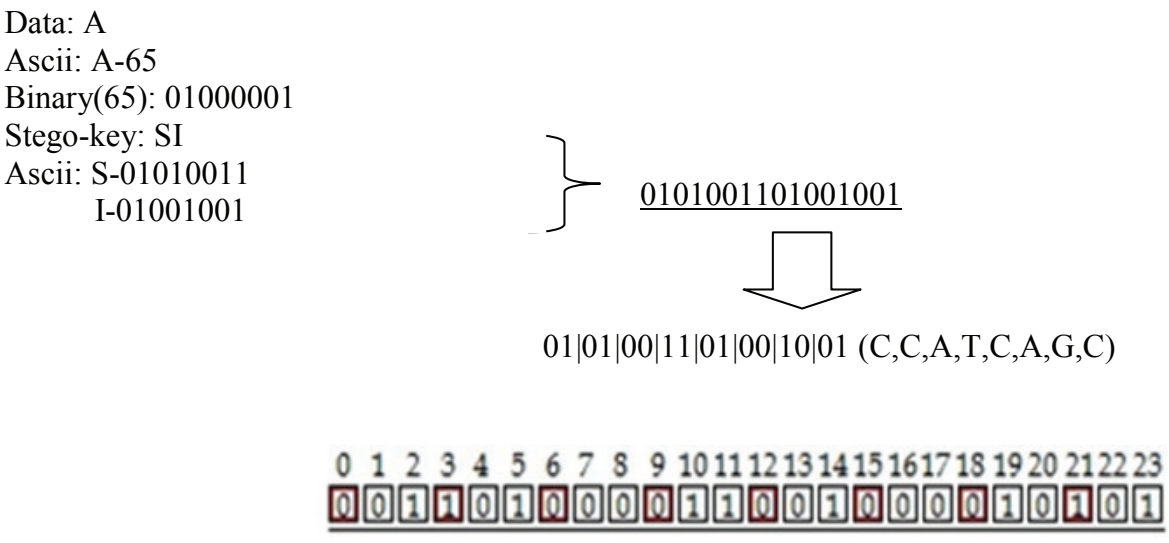

Figure 3. Mathematical demonstration of the encryption algorithm.

4. Result and Discussion

4.1 Encryption

First maintain the algorithms for Encryption process:
Encode by DNA-sequence:

Here red bits are data bit and by this encryption method if we have 1 byte of data then it will be 3 byte after DNA encoding, i.e. the file size will be increased 3-times of the original data.

First step is complete. Second step is Roundup Summation:
From Figure-1, we have 3 byte 00110100, 00110010, 00010101

Our stego-key- SI
ascii: S-83, I-73
So, summation(S+I) is (83+73) key = 157
Key = key mod 128;
Key = 29
byte[0] = 00110100, byte[1] = 00110010, byte[2] = 00010101
byte[0] = 52, byte[1] = 50, byte[2] = 21
Roundup summation:
byte [0] = 52+29 = 81<127
byte [1] = 50+29 = 79>127
byte [2] = 21+29 = 50>127
New encrypted bytes are- byte [0] = 81, byte [1] = 79, byte [2] = 50
Now we have 3 byte. Each pixel can hide 6 bit (2 bit each R, G, B). Therefore we require 4 pixels (2 x 2 images).
Suppose, Pixel [index] = (R, G, B)-values.
Let,
Pixel [0] = 120, 90, 140
Pixel [1] = 130, 60, 70
Pixel [2] = 110, 140, 70
Binary form of the pixels is:
Pixel [0] = 01111000, 01011010, 10001100
Pixel [1] = 10000010, 00111100, 01000110
Pixel [2] = 01101110, 10001100, 01000110
Pixel [3] = 11010010, 11111010, 10111110
Data bytes are: byte [0] = 81, byte [1] = 79, byte [2] = 50
byte[0] = 01000001
byte[1] = 01001111
byte[2] = 00110010
Data bit stream: 01000001 01001111 00110010
Encrypt data bit within the pixels
Pixel [0] = 01111001, 01011000, 10001100
Pixel [1] = 10000001, 00111101, 01000100
Pixel [2] = 01101111, 10001111, 01000100
Pixel [3] = 11010011, 11111000, 10111110
New encrypted pixels are:
Pixel [0] = 121, 88,140
Pixel [1] = 129, 61, 68
Pixel [2] = 111, 143, 68
Pixel [3] = 211, 248, 190
Now reform the bitmap type image with the new pixel.
Completion of Encryption method, I maintain the algorithm.

4.2 Decryption

In this process secrete message is decoded from the encrypted image or stego-image.
We have (2 x 2) encrypted image.
Now, encrypted pixels are:
Pixel [0] = 121, 88,140
Pixel [1] = 129, 61, 68
Pixel [2] = 111, 143, 68
Pixel [3] = 211, 248, 190
In binary representation:
Pixel [0] = 01111001, 01011000, 10001100
Pixel [1] = 10000001, 00111101, 01000100
Pixel [2] = 01101111, 10001111, 01000100
Pixel [3] = 11010011, 11111000, 10111110
Retrieve the underline bits.
So bits stream is: 01000001 01001111 00110010
Bytes are: byte [0] = 81, byte [1] = 79, byte [2] = 50
Stego-key: SI
Key = (int)S+(int)I = 157
Key = key mod 128
i.e. key = 157 mod 128 = 2
Round up subtraction:
byte [0] = 81-29 = 52 >-128
byte [1] = 79-29 = 50 >-128
byte [2] = 50-29 = 21 >-128
byte [0] = 52, byte [1] = 50, byte [2] = 21

Binary representation of retrieved bytes is:
After this, create new byte array from each retrieve 8 bit.
Completion of whole decoding constructs the file with the newly created byte array.

Now, bits stream is: 01000001. So the value is 65 i.e. 'A'.

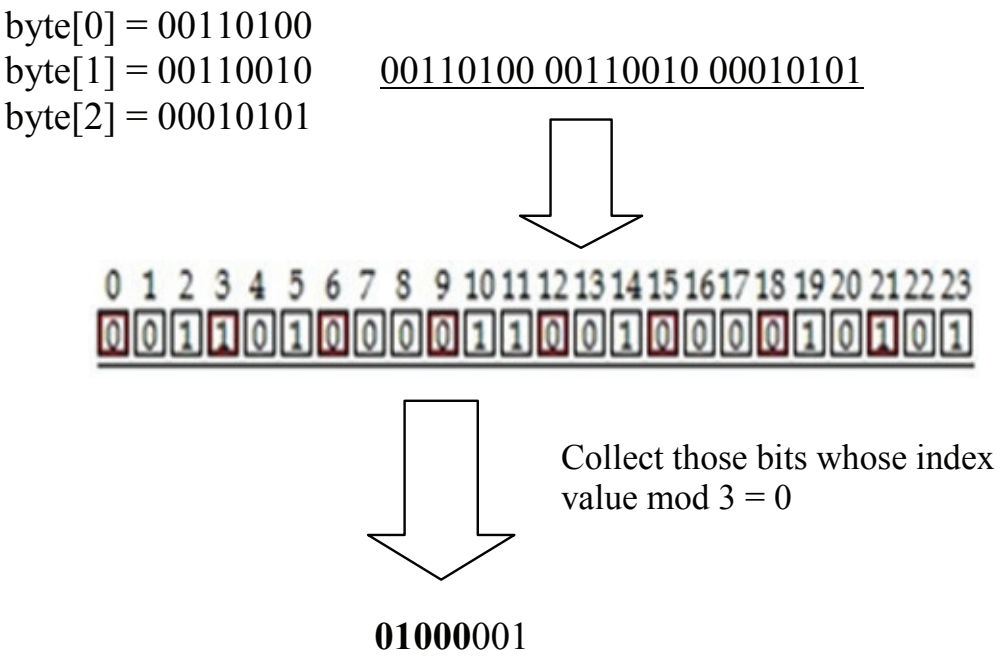

Figure 4. Mathematical demonstration of the decryption algorithm.

In order to retrieve the secret message by an unauthorized person, they must do the following[7]. Firstly, there are commonly one hundred and sixty three million DNA sequences present for public use. Thus, the probability of success is. Secondly, the random number generator and the two seeds may be required. Thirdly, the attacker has to know the Binary Coding Scheme.

For the second case: suppose there is a binary sequence S_1 which is handled with size n when it is in the data recovery stage. S_1 contain the secret message

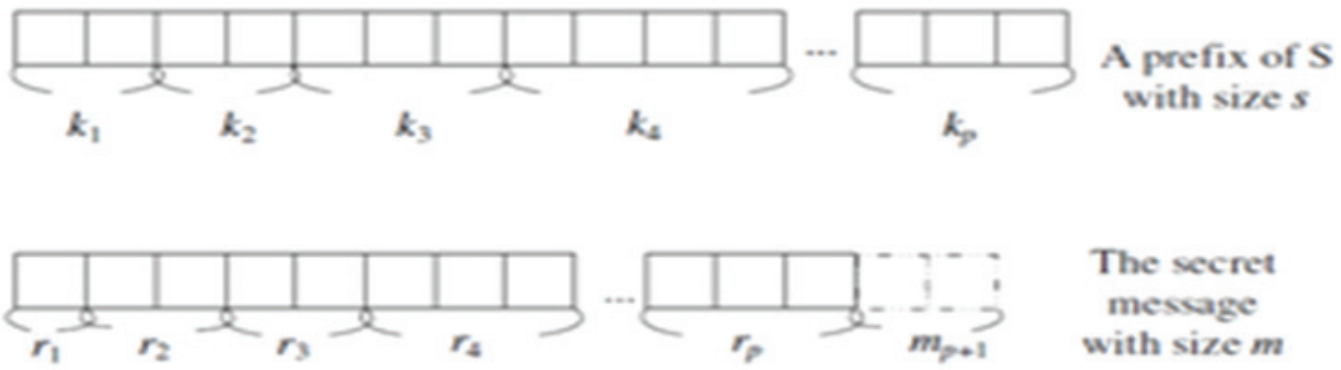

Figure 5. Secret massage generation using keys.

M and the prefix substring in reference sequence S. The size of M is m and the prefix of S is s, respectively. It is hard for an intruder to know the size of M and S. It can consider that an attacker could guess the value m and s first. It is known that m + s = n, where m, s, n ≥ 1 and there will be:

$$C\binom{2+n-2-1}{n-2} = C\binom{n-1}{n-2} \models n-1$$

Possibilities: the probability of the intruder's successful guessing m and s is. It is not enough for the intruder to recover the data. The problem for the attacker that he/she does not know the number sequences which will be generated by the random number seeds r and k denoted as $r_1, r_2,\ldots,r_p$ and $k_1,k_2,\ldots,k_p$, which will be used to break the secret message and the reference sequence S. The addition of r_i's and m_{p+1} is equal to m, and the addition of k_i's is equal to s. Figure 3 shows the relationship between m(s) and r_i's (k_i's). This is difficult for an intruder to know how many segments are divided. Thus, he/she have to try 2 segments, 3 segments, and then 4 segments and so on. Let the length of the remaining part in Step 2 of Algorithm 1-1 be q. Then, there may be the following cases:

$$\begin{aligned}
&r_1+q=m, \quad r_1 \geqslant 1,\ q \geqslant 0\\
&r_1+r_2+q=m, \quad r_1,r_2 \geqslant 1,\ q \geqslant 0\\
&r_1+r_2+r_3+q=m, \quad r_1,r_2,r_3 \geqslant 1,\ q \geqslant 0\\
&r_1+r_2+r_3+r_4+q=m, \quad r_1,r_2,r_3,r_4 \geqslant 1,\ q \geqslant 0\\
&r_1+r_2+r_3+\cdots+r_m+q=m, \quad r_1,r_2,r_3,\ldots,r_m \geqslant 1,\ q \geqslant 0
\end{aligned}$$

Thus, the probability of the intruder to make a successful guess for m at this stage is $\frac{1}{(2^s-1)}1/(2^m-1)$. Similarly, the probability of the intruder to make a successful guess for s at this stage is $\frac{1}{(2^s-1)}$. For the third situation: the number of the binary coding rules is four factorial is twenty four that the probability of the intruder to make a successful guess at this stage is $\frac{1}{24}$. Therefore the probability of the attacker to make a successful guess by this method is given by Insertion Method is $\frac{1}{1.63\times 10^8} \times \frac{1}{n-1} \times \frac{1}{2^m-1} \times \frac{1}{(2^s-1)} \times \frac{1}{24}$.

5. Conclusion

This paper is presenting a technique of image steganography, where the secrete message or data first encoded by DNA-sequence and then sum up each byte values with the stego-key. So the data retrieval will become tougher to the hackers. Authentication mechanisms which will allow for decryption that has the proper authentication values. Therefore it will give a better security rather than traditional LSB image steganography.

6. References

1. Bandyopadhyay SK, Bhattacharyya D, Ganguly D, Mukherjee S, Das P. A tutorial review on steganography. IC3. 2008; 106–14.
2. Torkaman MRN, Kazazi NS, Rouddini A. Innovative approach to improve hybrid cryptography by using DNA steganography. International Journal on New Computer Architectures and Their Applications. 2012; 2(1):225–36.
3. Guangzhao C, Cuiling L, Haobin L, Xiaoguang L. DNA computing and its application to information security field. 2009 5th International Conference on Natural Computation.
4. Marvel LM, Boncelet CG, Retter CT. Spread spectrum image Steganography. IEEE Transactions on Image Processing. 1999 Aug; 8(8).
5. Cheddad A, Condell J, Curran K, Kevitt PM. Digital image Steganography: Survey and analysis of current methods. Signal Processing. 2010 Mar; 90(3):727–52.
6. Zheng L, Cox I. JPEG based conditional entropy coding for correlated Steganography. Beijing, China: Proceedings of IEEE International Conference on Multimedia and Expo. 2007 Jul 2–5. p. 1251–4.
7. Shiu HJ, Ng KL, Fang JF, Lee RCT, Huang CH. Data hiding methods based upon DNA sequences. Information Sciences. Elsevier. 2010; 180:2196–208.
8. Manna S, Roy S, Roy P, Bandyopadhyay SK. Modified technique of insertion methods for data hiding using DNA sequences. 1st International Conference on Automation, Control, Energy and Systems. p. 1–5.
9. Srinivasan B, Arunkumar S, Rajesh K. A novel approach for color image, steganography using nubasi and randomized, secret sharing algorithm. Indian Journal of Science and Technology. 2015 Apr; 8(supl 7).
10. Vidya G, Hema PR, Shilpa GS, Kalpana V. Image steganography using Ken Ken Puzzle for secure data hiding. Indian Journal of Science and Technology. 2014 Sep; 7(9).