# Combined Steganography and Image Cryptography System for Secure Data Transfer

**Anitha Malaghan[1] Aditya M[2], Ajith B[3], Anjana S[4] & Karthik GPN[5]**

*[1]Assistant Professor, [2,3,4&5]Student,Computer Science and Engineering, ACS College of Engineering, India*

***DOI**: https://doi.org/10.34293/acsjse.v2i1.28*

**Abstract -** *With billions of active internet users every given moment, user data privacy is vulnerable to potential attackers, so secure digital transmissions have always been a concern. With numerous researches going on in the field of cryptography systems there is always search new methods that can improve security of data sent over the internet. In this project a model is being designedwith integrated various cryptographic techniques to make a hybrid crypto system for more reliable secure digital transmission. This project incorporates plain text cryptography, image steganography, and visual cryptography to encrypt a text all the way through getting an image that is distorted with hidden data inside. The primary step is to convert the plain text to cipher text using a symmetric cryptography algorithm, then using the method of steganography the cipher text is hidden inside the image and finally the concept of image cryptography is used to get a cipher image with encryption keys that encrypted the same. This project is going to be developed to accept globally spoken English language.*

***Keywords:** Steganography, Image Cryptography, Image Encryption, Security*

## I INTRODUCTION

The past couple of decades there is a linear increase in internet users, and time-to-time we have seen the threat of data security vulnerability and liability of data that is personal or business that could be used for vicious actions. To improve data security cryptography is a method used to encrypt and decrypt the data for secure data transfer over the internet. But recently image cryptosystems and steganography has taken some places to communicate data. Steganography has provided water marking audio and video files that can limit illegal copywrite and track digital prints. Interesting techniques have been proposed to support steganography, one such prominent method is the Least Significant Bit LSB.Image cryptography also has played a part in internet communication, digital water marking, medical imaging, and military communication. There have been many novel and efficient cryptography techniques been proposed [11][12][13].

The interest to develop this system aroused during studying the theory about image cryptography and steganography, and reference of journals such as optics communication [5], LSB technique[4], and more.

There are various methods proposed for secure image transfer that combine cryptography methods and steganography [3][4]. With many techniques of image cryptography and steganography suggesting useful applications and various techniques that use combining these various cryptography methods. In this project, an integrated system is implemented using both steganography and image cryptography that can be used for secure data transfer.

The remainder of the paper is ordered as follows. Section 2 reviews related works. A comprehensive explanation is described on implemented method is Section 3. In Section 4 review and experiments on the system is shown. And finally in section 5 we conclude with conclusion and further work.

## II LITERATURE SURVEY

Various techniques have been proposed on steganography the most popular method used is Least Significant Bit (LSB). Least Significant Bit reference from paper[2] method shows a mode of LSB substitution to hide data in images that shows a improvement in area of payload by using method of conversion of data bits to base 3, this method is also prone to detection and three colour planes is used for all LSB substation. Another work proposed by X. Zhou and W. Gong et al [2] is an improved modification to the existing LSB technique which combines hiding and cryptography similar to reference [3].A colour image LSB technique is presented in the paper[6], in this technique the pixels with only values R, G, and B are used for hiding data and the values must not be equal to one another and least a difference of two with other components. This method uses only one colour component per pixel which is used to hide data.

Image cryptography has various methods proposed that show capable results, chaos-based schemes that include two steps: chaotic pixel permutation step and pixel diffusion step. The first step consists of confusion stage, chaotic map combination is used to apprehend the pixel confusion combination, confusion key is generated here, keys here refer to the parameter of chaotic map. The second step, pixel diffusion each pixel value in the image is changed based on the first step chaotic stage. The diffusion key represents the diffusion function. These techniques of chaotic map are used in paper,[8] that improves problems in the existing system using chaotic coupled lattices, trigonometric maps are used to increase space between the chaotic confusion key [5]. Rubik’s algorithm image encryption algorithm is used in paper [7], the algorithm scrambles pixels in a gray plainimage using the Rubik’s method where each position of pixels is changed, that is only deployed changes at this step is changes to the positioning of position of pixels in the image.There are two random secrete keys used.These two keys are applied into a bitwise XOR for the odd rows and columns present in the plain-image.Also, for the even row and columns present in to plain-image, the bitwise XOR is applied using flipped secrete keys.The number of iterations is repeated until a final iteration is reached, for each step number of iterations are performed.

## III RELATED WORK

In this section the implementation of the image cryptosystem is described, both the encryption and decryption methods are described. This section is ordered as follows. Section 3.1 comprises of steganography implementation and Section 3.2 comprises of implementation of image cryptography. Each of these two sections is comprehensive of encryption and decryption algorithm. Finally, section 3.3 comprises of the final integrated system and describes the classes and its functionality. This project is implemented using the language python.

## 3.1 Steganography

In this section, Least Significant Bit (LSB) technique is implemented for hiding data in colour images. The LSB method consists of four classes to hide data in image and retrieve the same data from the image. The classes are: genData(), modPix(), encode_data(), and decode_data().

The genData() class basically generates binary code of the input data that is being hidden, modPix() class analyses the length of binary data and number of pixels in the image then three pixels are taken at once and pixels values are changed, encode_data() class is called after modPix() and basic functionality is to place the modified pixels into a new image, and decode_data() class simply extracts the data from image by reverse engineering.

**LSB message Hiding Algorithm**

1. A color image as a cover or transmission medium is selected.
2. Also, the message that is to be hidden in the image that is the secrete information is input to get a steganography image
3. Then a new instance of the image is created for the selected image.
4. Now, encode_data() class is called which accepts the new image instance and the input data.
5. In encode_data(), the modPixel() class is called which accepts individual pixel in the image with the data.
6. The modPixel(), generates codes for the data or the input message using the genData().
7. Now in the modPixel() for the length of the generated data for genData() pixel value is changed.
8. With in range of the data pixels values are made by declaring odd as values as 1 and even values as 0.
9. The reading of pixels is done as, for every eight pixels 0 or 1 is checked, if its 0 it means continue to read else to stop read or 1 says message is over
10. Back in the encode_data() class each modified pixel is again placed in the same position using for so that the image looks exactly the same.
11. Finally, the modified image is generated that contains the hidden information.

**Decoding the hidden message in image**

1. The steganography image is the input which contains the secrete.
2. This procedure is to get the string back from the image that is hidden.
3. We first declare a data field to store the retrieved data.
4. Now we loop the image for all true values.
5. So, three pixels are read at a time and also a binary string filed is declared to store odd or even values.
6. Then data is retrieved from the pixels.
7. Finally, the data retrieved is returned.

## 3.2 Image Cryptography

The implementation of image cryptography is described in this section, Rubik's Cube principal algorithm is used for image cryptography. Rubik's algorithm image encryption algorithm is used in paper [7], the algorithm scrambles pixels in a gray plain image using the Rubik's method where each position of pixels is changed, that is only deployed changes at this step is changes to the positioning of position of pixels in the image. There are two random secrete keys used. These two keys are applied into a bitwise XOR for the odd rows and columns present in the plain-image. Also, for the even row and columns present in to plain-image, the bitwise XOR is applied using flipped secrete keys. The number of iterations is repeated until a final iteration is reached, for each step number of iterations are performed. The above is divided into five classes where three classes are commonly used in both encryption and decryption. The common classes are: upshift(), downshift(), and rotate180(). The other two classes are purely named image_encrypt() and image_decrypt(). The upshift(), downshift(), and rotate180() are used to shift and reorder pixels in the image, these three classes are placed in a separate file called as shift_helper.py. The image_encrypt() class divides the image into R, G, and B pixels and the classes in shift_helper.py is used to reorder. Similarly, image_decode() decodes the shuffled image back to original image.

### Functions used to Shift and Shuffle the Pixels in Encryption and Decryption

1. There are three different functions used to shuffle and shift the pixels, they are upshift, downshift and rotate180.
2. The upshift function, this function is passed with three attributes the r, g, or b arrays extracted from the picture, row or column number and the current row or column values.
3. The upshift will append the col declared array with current j value of the length and the index of the iteration shifts the column using NumPy roll function of -n times. Also will match $a_{i,j}$to shifted column$_i$.
4. The downshift function also accepts attributes accepted by upshift that are the r, g, or b arrays extracted from the picture, row or column number and the current row or column values.
5. Here instead of shifting -n the NumPy roll shifts values n for the current col.
6. Finally, the rotate180 accepts an integer and returns a binary string and the values of it.

### Rubik's Cube Encryption System

1. In the encryption method a new image input and the new image name is passed.
2. The pixels in the image are loaded in to a variable using image load.
3. The next step is to load r, g, and b pixels, for the range of image size $_{[0]}$ all are appended the n for image size $_{[1]}$ pixel$_{i,j}$ r$_i$ , g$_i$and b$_i$ arrays.
4. New arrays m and n are set to size $_{[0]}$ and size $_{[1].}$
5. Then K$_r$ and K$_c$ values are computed using randint K$_r$ with power$_{2,\ 8}$ for range of m i.e size $_{[0]}$ , K$_c$ with power$_{2,\ 8}$ for range of m i.e size $_{[1]}$.

6. Then in range of m total sum of all three r, g and b pixels are set using sum and modulus of same is calculated and stored.
7. Now for each modulus we check condition for each i in range of mi.e size $_{[0]}$, if the modulus is equal to 0 then $r_i$ and $kr_i$ are rolled using NumPy roll.
8. Similarly for range of n i.e size $_{[1]}$ Kc values are conditioned and upshift if equal to zero and downshift function if not 0.
9. Then for range of $m_i$ and $n_j$, if modulus of i is equal to 1 $r_{i, j}$, $g_{i, j}$, and $b_{i, j}$ is powered to $k_j$ else rotate180 function of $k_i$ is passed for each row.
10. Then for range of $m_i$ and $n_j$, if modulus of j is equal to 1 $r_{i, j}$ , $g_{i, j}$, and $b_{i, j}$ is powered to $k_j$ else rotate180 function of $k_i$ is passed for each column.
11. Finally, for pixel$_{i,}$ j in range the $r_{i, j}$, $g_{i, j}$, and $b_{i, j}$ pixels are arranged and a new image with the input name is titled as the new encrypted image.

**Rubik's Cube Decryption System**

1. This process is reverse of encryption or reverse engineering of the encryption algorithm.
2. First step is to load r, g, and b pixels, for the range of image size $_{[0]}$ all are appended the n for image size $_{[1]}$pixel$_{i, j}$$r_i$ , $g_i$ and $b_i$ arrays.
3. New arrays m and n are set to size $_{[0]}$ and size $_{[1].}$
4. Then for range of $m_i$ and $n_j$, if modulus of i is equal to 1 $r_{i, j}$ , $g_{i, j}$, and $b_{i, j}$ is powered to $k_j$ else rotate180 function of $k_i$ is passed for each row.
5. Then for range of $m_i$ and $n_j$, if modulus of j is equal to 1 $r_{i, j}$ , $g_{i, j}$, and $b_{i, j}$ is powered to $k_j$ else rotate180 function of $k_i$ is passed for each column.
6. Now for each modulus we check condition for each i in range of mi.e size $_{[0]}$, if the modulus is equal to 0 then $r_i$ and $kr_i$ are rolled using NumPy roll.
7. Similarly for range of n i.e size $_{[1]}$ Kc values are conditioned and upshift if equal to zero and downshift function if not 0.
8. Finally, for pixel$_{i,}$ j in range the $r_{i, j}$ ,$g_{i, j}$, and $b_{i, j}$ pixels are arranged and a image is retrieved that is exactly same as to the plain image before encryption.

### 3.3 Combined Steganography and Image Cryptography System for Secure Data Transfer

This section talks about the integration of both steganography and image cryptography into the same system. As specified earlier python is used to implement the project, the file structure of the system is as follows. There are four python files main.py, encode.py, decode.py, shift_helper.py. After encryption and decryption is performed on an image the output image is placed inencrypted and decrypted image folder simultaneously. The encode.py consists of classes genData(), modPix(), encode_data() and image_encrypt(). In decode.py comprises of decryption classes, decode_data() and image_decode(). The shift_helper.py consists of upshift(), downshift(), and rotate180() classes. Finally, main.py is entry point of the system that takes input and takes command from the user to encrypt and decrypt.

### 3.3.1 Proposed System

The proposed system asks the user to either encrypt or decrypt, the encrypt option takes input, an image and data to be ciphered and the system outputs set of key values and cipher image. The decrypt option accepts input the cipher image and key values for the same image.

The proposed encrypt system is shown figure (1) below and described as follows:

**Step 1: Plain text and an image are taken as input**

a. The plain text is converted into binary code and is hidden in the input image using Least Significant Bit.
b. The steganography image is next subjected to image cryptography using Rubik's cube principle.
c. After the encrypt is completed, set of keys and an encrypted image is placed in encrypt image folder.

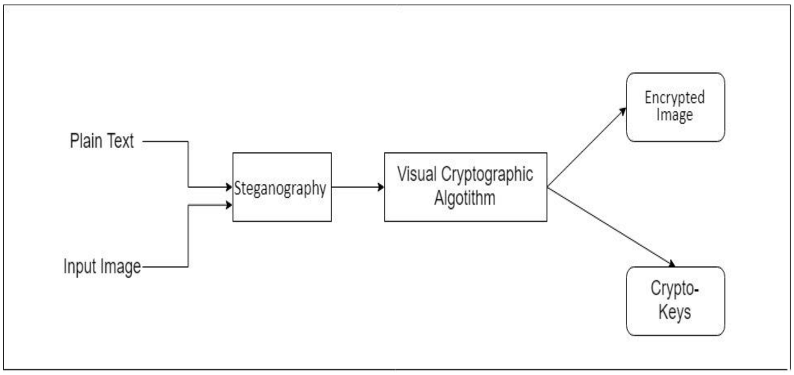


**Figure 1 Encryption Process**

The decryption system is shown in figure (2) below:

**Step 2:** The cipher image and the key values are accepted as the input.

**Step 3:** If the valid keys and image is given, plain text is obtained.

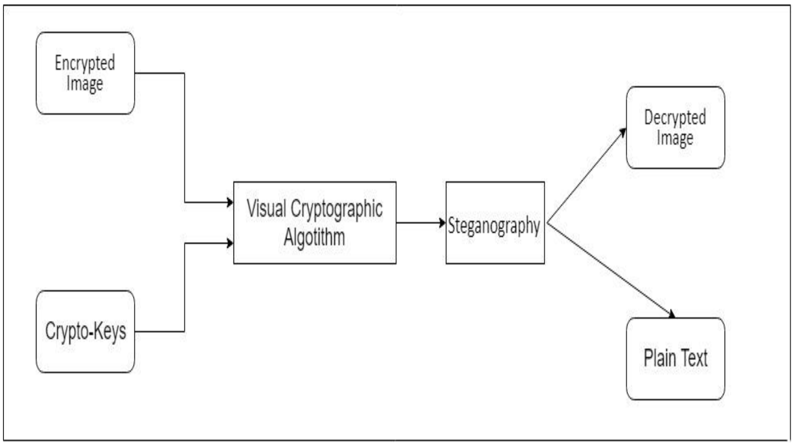


**Figure 2 Decryption Process**

## IV SIMULATION RESULTS

In this section results of the above-described system is shown with examples.

Here there are two present results with text data input with an image and the decrypted image and the plain text obtained.

The first test was a simple white png image with black dots is the first test. The plain text given to be hidden was “Hide this data, test one”. The image, the key values and the encrypted image is shown below. The accuracy was 100% the entered plain text was successfully obtained.

The second test was conducted on lenna.png lenna model image. The plain text given to be hidden was “Test tow, hide this”. The image, the key values and the encrypted image is shown below. The accuracy was 100% the entered plain text was successfully obtained.

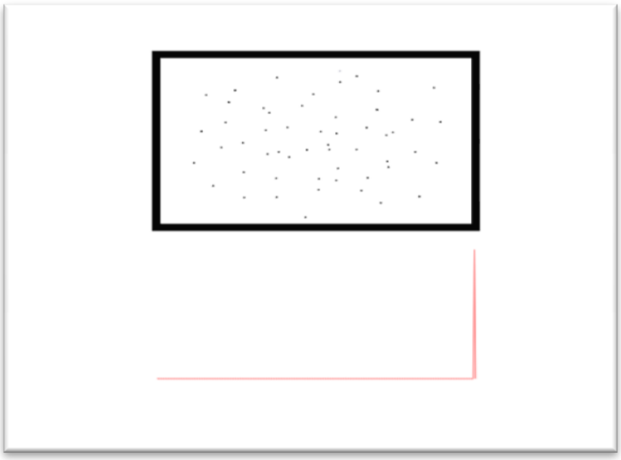
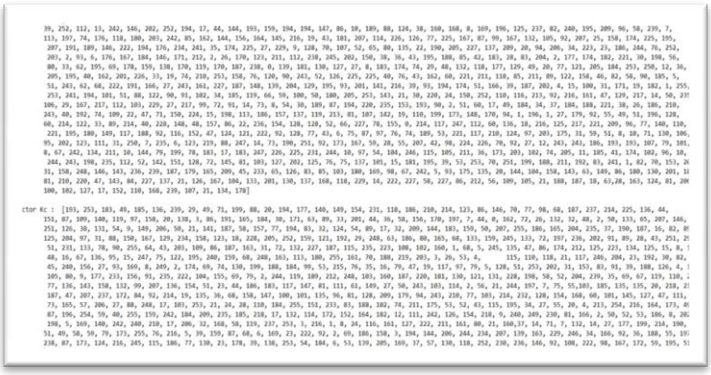


**Figure 3: Dot’s image with Histogram (dots.png)**



```
 39, 252, 112, 13, 242, 146, 202, 252, 194, 17, 44, 144, 193, 159, 194, 194, 147, 86, 10, 189, 88, 124, 38, 160, 168, 8, 169, 196, 125, 237, 82, 240, 195, 209, 96, 58, 239, 7,
 113, 197, 74, 176, 118, 180, 203, 242, 85, 162, 144, 156, 164, 145, 216, 19, 43, 181, 207, 114, 226, 126, 77, 225, 167, 87, 99, 167, 132, 105, 92, 207, 25, 158, 174, 225, 195,
  207, 191, 189, 146, 222, 194, 176, 234, 241, 35, 174, 225, 27, 229, 9, 128, 70, 107, 52, 65, 80, 135, 22, 190, 205, 227, 137, 209, 20, 94, 206, 34, 223, 23, 186, 244, 76, 252,
  203, 2, 93, 6, 176, 167, 184, 146, 171, 212, 2, 26, 170, 123, 211, 112, 238, 245, 202, 150, 38, 36, 43, 195, 188, 85, 42, 183, 28, 83, 204, 2, 177, 174, 182, 221, 30, 198, 56,
  80, 33, 62, 195, 69, 178, 159, 138, 170, 119, 170, 187, 238, 0, 139, 181, 130, 127, 27, 8, 183, 174, 74, 29, 48, 132, 118, 177, 129, 49, 20, 77, 121, 205, 184, 253, 250, 12, 36,
  205, 195, 40, 162, 201, 226, 33, 19, 74, 210, 253, 158, 76, 120, 90, 243, 52, 126, 225, 225, 40, 76, 43, 162, 60, 221, 211, 118, 85, 211, 89, 122, 158, 46, 82, 58, 90, 185, 5,
  51, 243, 62, 68, 222, 191, 166, 27, 243, 161, 227, 187, 148, 139, 204, 129, 195, 93, 201, 141, 216, 39, 93, 194, 174, 51, 166, 39, 187, 202, 4, 15, 100, 31, 171, 19, 182, 1, 255,
  253, 241, 194, 101, 51, 88, 122, 90, 91, 102, 34, 185, 119, 66, 59, 100, 50, 180, 205, 253, 143, 21, 30, 220, 24, 150, 252, 110, 116, 213, 92, 216, 161, 47, 129, 217, 14, 50, 23
 106, 29, 167, 217, 112, 103, 229, 27, 217, 99, 72, 91, 14, 73, 8, 54, 30, 189, 87, 194, 220, 235, 153, 193, 90, 2, 51, 60, 17, 49, 184, 34, 37, 184, 188, 221, 38, 26, 186, 210,
 243, 40, 192, 74, 109, 22, 47, 71, 150, 224, 15, 198, 113, 186, 157, 137, 119, 213, 81, 107, 142, 19, 110, 199, 173, 148, 170, 94, 1, 196, 1, 27, 179, 92, 55, 49, 51, 196, 128,
 60, 214, 122, 33, 89, 214, 40, 228, 148, 48, 157, 86, 22, 236, 154, 128, 128, 52, 66, 227, 78, 155, 0, 214, 117, 247, 112, 60, 136, 18, 216, 125, 217, 221, 209, 96, 77, 140, 110,
  221, 195, 180, 149, 117, 188, 92, 116, 152, 47, 124, 121, 222, 92, 128, 77, 43, 6, 75, 87, 97, 76, 74, 189, 53, 221, 117, 210, 124, 97, 203, 175, 31, 59, 51, 8, 10, 71, 130, 106,
 95, 202, 123, 111, 31, 250, 7, 235, 6, 123, 219, 88, 247, 14, 73, 190, 251, 92, 173, 167, 59, 28, 55, 207, 42, 98, 224, 226, 70, 92, 27, 12, 243, 243, 186, 193, 193, 107, 79, 101,
 8, 67, 242, 134, 211, 10, 144, 79, 199, 78, 183, 17, 183, 247, 226, 225, 231, 244, 10, 97, 54, 104, 246, 115, 105, 211, 36, 173, 203, 102, 74, 205, 11, 185, 41, 174, 102, 96, 10,
  244, 243, 198, 235, 112, 52, 142, 151, 128, 72, 145, 81, 103, 127, 202, 125, 76, 75, 137, 101, 15, 181, 195, 39, 53, 253, 70, 251, 199, 188, 211, 192, 83, 241, 1, 82, 70, 153, 2
 31, 158, 248, 146, 143, 236, 239, 187, 179, 165, 209, 45, 233, 65, 126, 83, 85, 103, 180, 169, 98, 67, 242, 5, 93, 175, 135, 20, 144, 104, 158, 143, 63, 149, 86, 180, 130, 201, 1
 81, 210, 220, 47, 143, 84, 227, 137, 21, 126, 167, 104, 133, 201, 130, 137, 168, 118, 229, 14, 222, 227, 58, 227, 86, 212, 56, 109, 105, 21, 188, 187, 18, 63,28, 163, 124, 81, 20
 180, 102, 127, 17, 152, 110, 168, 239, 107, 21, 134, 178]

ctor Kc :  [193, 253, 183, 49, 185, 136, 239, 29, 49, 71, 199, 88, 20, 194, 177, 140, 149, 154, 231, 118, 186, 210, 214, 123, 86, 146, 70, 77, 98, 68, 187, 237, 214, 225, 136, 44,
 151, 87, 109, 140, 119, 97, 158, 20, 138, 3, 86, 191, 165, 184, 30, 171, 63, 89, 33, 201, 44, 36, 58, 156, 170, 197, 7, 44, 0, 162, 72, 26, 132, 32, 48, 2, 50, 133, 65, 207, 146,
 251, 126, 38, 131, 54, 9, 149, 206, 50, 21, 141, 187, 58, 157, 77, 194, 83, 32, 124, 54, 89, 17, 32, 209, 144, 183, 159, 50, 207, 255, 186, 165, 204, 235, 37, 190, 187, 16, 82, 8
 125, 204, 97, 31, 88, 150, 167, 129, 234, 158, 123, 18, 228, 205, 252, 159, 121, 192, 29, 248, 63, 186, 80, 165, 68, 133, 159, 245, 133, 72, 197, 236, 202, 91, 89, 28, 43, 251, 2
  51, 231, 133, 78, 90, 255, 64, 43, 203, 109, 86, 187, 163, 31, 72, 132, 227, 187, 115, 235, 223, 108, 102, 160, 1, 68, 5, 245, 135, 47, 86, 174, 212, 125, 223, 134, 125, 15, 8, 1
  48, 16, 67, 136, 95, 15, 247, 75, 122, 195, 240, 159, 68, 248, 163, 113, 180, 255, 161, 70, 188, 219, 203, 3, 26, 53, 4,       115, 110, 118, 21, 117, 246, 204, 23, 192, 30, 82,
  45, 240, 156, 27, 93, 169, 8, 249, 2, 174, 69, 74, 130, 199, 188, 184, 99, 53, 215, 76, 35, 16, 79, 47, 19, 117, 97, 79, 5, 128, 51, 253, 202, 31, 153, 83, 91, 39, 188, 126, 4, 1
  105, 80, 9, 177, 233, 156, 91, 235, 222, 104, 155, 69, 79, 2, 244, 119, 189, 212, 248, 103, 160, 187, 220, 181, 130, 121, 131, 228, 198, 58, 52, 204, 239, 35, 69, 67, 119, 110,
  77, 136, 143, 158, 132, 99, 207, 136, 154, 51, 23, 44, 186, 183, 117, 147, 81, 111, 61, 149, 27, 50, 243, 103, 114, 2, 56, 21, 244, 197, 7, 75, 55,103, 185, 135, 135, 20, 218, 2
  187, 47, 207, 237, 172, 84, 92, 214, 19, 135, 36, 68, 158, 147, 100, 101, 135, 96, 81, 128, 209, 179, 94, 243, 210, 77, 103, 214, 232, 128, 154, 168, 60, 101, 145, 127, 47, 111,
  73, 165, 57, 206, 27, 88, 248, 17, 103, 253, 21, 24, 28, 110, 184, 255, 151, 233, 83, 188, 102, 74, 211, 175, 53, 52, 43, 115, 195, 34, 27, 55, 20, 4, 213, 254, 216, 164, 173, 4
  87, 196, 254, 59, 40, 255, 159, 242, 184, 209, 235, 185, 218, 17, 132, 114, 172, 152, 164, 182, 12, 111, 242, 126, 154, 218, 9, 240, 249, 230, 81, 166, 2, 50, 52, 53, 186, 8, 20
  198, 5, 169, 140, 242, 240, 210, 17, 206, 32, 168, 58, 119, 237, 253, 3, 216, 1, 8, 24, 116, 161, 127, 222, 211, 161, 80, 21, 160,37, 14, 71, 7, 132, 14, 27, 177, 199, 214, 190,
 51, 49, 58, 59, 79, 173, 255, 76, 216, 5, 39, 159, 87, 68, 6, 169, 23, 222, 92, 2, 69, 186, 158, 3, 194, 144, 206, 244, 234, 207, 139, 163, 229, 246, 34, 166, 92, 36, 188, 55, 19
 238, 87, 173, 124, 216, 245, 115, 186, 77, 130, 23, 178, 39, 138, 253, 54, 184, 6, 53, 139, 205, 169, 37, 57, 130, 118, 252, 230, 236, 146, 92, 108, 222, 98, 167, 172, 59, 195, 5
```

**Figure 4: Key values for dots.png**

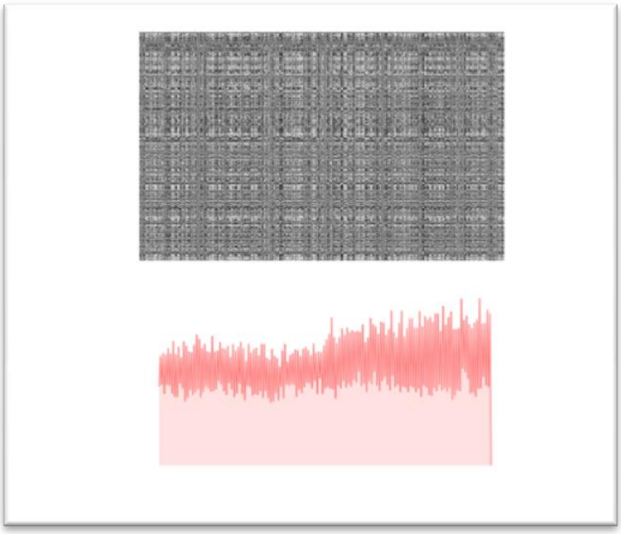


**Figure 5: Encrypted Image and Histogram (dots.png)**

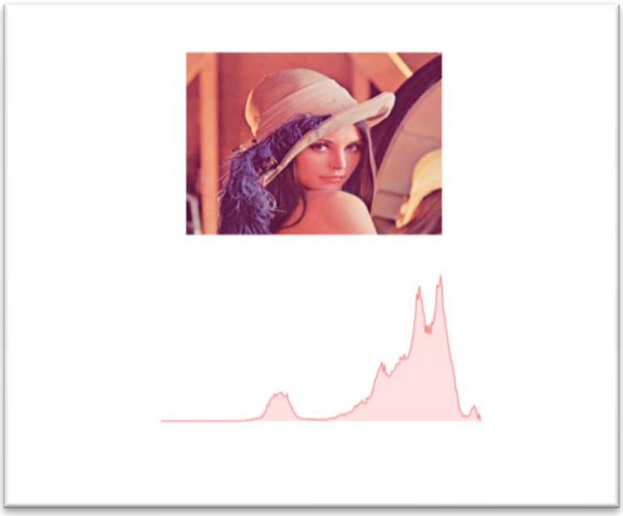
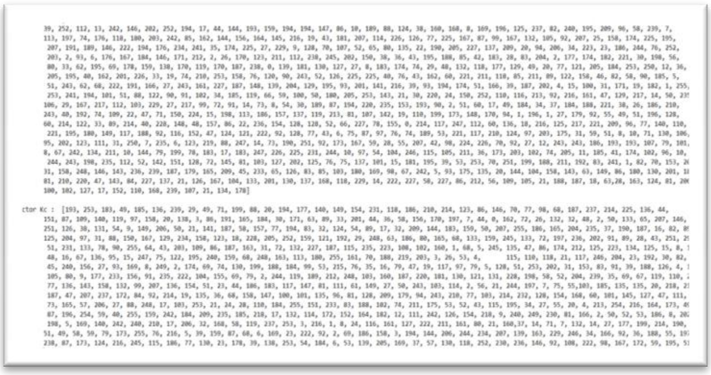


**Figure 6: Input Image and Histogram (Lenna.PNG)**



```
39, 252, 112, 13, 242, 146, 202, 252, 194, 17, 44, 144, 193, 159, 194, 194, 147, 86, 10, 189, 88, 124, 38, 160, 168, 8, 169, 196, 125, 237, 82, 240, 195, 209, 96, 58, 239, 7,
113, 197, 74, 176, 118, 180, 203, 242, 85, 162, 144, 156, 164, 145, 216, 19, 43, 181, 207, 114, 226, 126, 77, 225, 167, 87, 99, 167, 132, 105, 92, 207, 25, 158, 174, 225, 195,
 207, 191, 189, 146, 222, 194, 176, 234, 241, 35, 174, 225, 27, 229, 9, 128, 70, 107, 52, 65, 80, 135, 22, 190, 205, 227, 137, 209, 20, 94, 206, 34, 223, 23, 186, 244, 76, 252,
 203, 2, 93, 6, 176, 167, 184, 146, 171, 212, 2, 26, 170, 123, 211, 112, 238, 245, 202, 150, 38, 36, 43, 195, 188, 85, 42, 183, 28, 83, 204, 2, 177, 174, 182, 221, 30, 198, 56,
 80, 33, 62, 195, 69, 178, 159, 138, 170, 119, 170, 187, 238, 0, 139, 181, 130, 127, 27, 8, 183, 174, 74, 29, 48, 132, 118, 177, 129, 49, 20, 77, 121, 205, 184, 253, 250, 12, 36,
 205, 195, 40, 162, 201, 226, 33, 19, 74, 210, 253, 158, 76, 120, 90, 243, 52, 126, 225, 225, 40, 76, 43, 162, 60, 221, 211, 118, 85, 211, 89, 122, 158, 46, 82, 58, 90, 185, 5,
 51, 243, 62, 68, 222, 191, 166, 27, 243, 161, 227, 187, 148, 139, 204, 129, 195, 93, 201, 141, 216, 39, 93, 194, 174, 51, 166, 39, 187, 202, 4, 15, 100, 31, 171, 19, 182, 1, 255,
 253, 241, 194, 101, 51, 88, 122, 90, 91, 102, 34, 185, 119, 66, 59, 100, 50, 180, 205, 253, 143, 21, 30, 220, 24, 150, 252, 110, 116, 213, 92, 216, 161, 47, 129, 217, 14, 50, 23
106, 29, 167, 217, 112, 103, 229, 27, 217, 99, 72, 91, 14, 73, 8, 54, 30, 189, 87, 194, 220, 235, 153, 193, 90, 2, 51, 60, 17, 49, 184, 34, 37, 184, 188, 221, 38, 26, 186, 210,
243, 40, 192, 74, 109, 22, 47, 71, 150, 224, 15, 198, 113, 186, 157, 137, 119, 213, 81, 107, 142, 19, 110, 199, 173, 148, 170, 94, 1, 196, 1, 27, 179, 92, 55, 49, 51, 196, 128,
60, 214, 122, 33, 89, 214, 40, 228, 148, 48, 157, 86, 22, 236, 154, 128, 128, 52, 66, 227, 78, 155, 0, 214, 117, 247, 112, 60, 136, 18, 216, 125, 217, 221, 209, 96, 77, 140, 110,
 221, 195, 180, 149, 117, 188, 92, 116, 152, 47, 124, 121, 222, 92, 128, 77, 43, 6, 75, 87, 97, 76, 74, 189, 53, 221, 117, 210, 124, 97, 203, 175, 31, 59, 51, 8, 10, 71, 130, 106,
95, 202, 123, 111, 31, 250, 7, 235, 6, 123, 219, 88, 247, 14, 73, 190, 251, 92, 173, 167, 59, 28, 55, 207, 42, 98, 224, 226, 70, 92, 27, 12, 243, 243, 186, 193, 193, 107, 79, 101,
8, 67, 242, 134, 211, 10, 144, 79, 199, 78, 183, 17, 183, 247, 226, 225, 231, 244, 10, 97, 54, 104, 246, 115, 105, 211, 36, 173, 203, 102, 74, 205, 11, 185, 41, 174, 102, 96, 10,
 244, 243, 198, 235, 112, 52, 142, 151, 128, 72, 145, 81, 103, 127, 202, 125, 76, 75, 137, 101, 15, 181, 195, 39, 53, 253, 70, 251, 199, 188, 211, 192, 83, 241, 1, 82, 70, 153, 2
31, 158, 248, 146, 143, 236, 239, 187, 179, 165, 209, 45, 233, 65, 126, 83, 85, 103, 180, 169, 98, 67, 242, 5, 93, 175, 135, 20, 144, 104, 158, 143, 63, 149, 86, 180, 130, 201, 1
81, 210, 220, 47, 143, 84, 227, 137, 21, 126, 167, 104, 133, 201, 130, 137, 168, 118, 229, 14, 222, 227, 58, 227, 86, 212, 56, 109, 105, 21, 188, 187, 18, 63,28, 163, 124, 81, 20
100, 102, 127, 17, 152, 110, 168, 239, 107, 21, 134, 178]

ctor Kc :  [193, 253, 183, 49, 185, 136, 239, 29, 49, 71, 199, 88, 20, 194, 177, 140, 149, 154, 231, 118, 186, 210, 214, 123, 86, 146, 70, 77, 98, 68, 187, 237, 214, 225, 136, 44,
           151, 87, 109, 140, 119, 97, 158, 20, 138, 3, 86, 191, 165, 184, 30, 171, 63, 89, 33, 201, 44, 36, 58, 156, 170, 197, 7, 44, 0, 162, 72, 26, 132, 32, 48, 2, 50, 133, 65, 207, 146,
           251, 126, 38, 131, 54, 9, 149, 206, 50, 21, 141, 187, 58, 157, 77, 194, 83, 32, 124, 54, 89, 17, 32, 209, 144, 183, 159, 50, 207, 255, 186, 165, 204, 235, 37, 190, 187, 16, 82, 8
           125, 204, 97, 31, 88, 150, 167, 129, 234, 158, 123, 18, 228, 205, 252, 159, 121, 192, 29, 248, 63, 186, 80, 165, 68, 133, 159, 245, 133, 72, 197, 236, 202, 91, 89, 28, 43, 251, 2
            51, 231, 133, 78, 90, 255, 64, 43, 203, 109, 86, 187, 163, 31, 72, 132, 227, 187, 115, 235, 223, 108, 102, 160, 1, 68, 5, 245, 135, 47, 86, 174, 212, 125, 223, 134, 125, 15, 8, 1
            48, 16, 67, 136, 95, 15, 247, 75, 122, 195, 240, 159, 68, 248, 163, 113, 180, 255, 161, 70, 188, 219, 203, 3, 26, 53, 4,      115, 110, 118, 21, 117, 246, 204, 23, 192, 30, 82,
            45, 240, 156, 27, 93, 169, 8, 249, 2, 174, 69, 74, 130, 199, 188, 184, 99, 53, 215, 76, 35, 16, 79, 47, 19, 117, 97, 79, 5, 128, 51, 253, 202, 31, 153, 83, 91, 39, 188, 126, 4, 1
            105, 80, 9, 177, 233, 156, 91, 235, 222, 104, 155, 69, 79, 2, 244, 119, 189, 212, 248, 103, 160, 187, 220, 181, 130, 121, 131, 228, 198, 58, 52, 204, 239, 35, 69, 67, 119, 110, 2
            77, 136, 143, 158, 132, 99, 207, 136, 154, 51, 23, 44, 186, 183, 117, 147, 81, 111, 61, 149, 27, 50, 243, 103, 114, 2, 56, 21, 244, 197, 7, 75, 55,103, 185, 135, 135, 20, 218, 2
            187, 47, 207, 237, 172, 84, 92, 214, 19, 135, 36, 68, 158, 147, 100, 101, 135, 96, 81, 128, 209, 179, 94, 243, 210, 77, 103, 214, 232, 128, 154, 168, 60, 101, 145, 127, 47, 111,
            73, 165, 57, 206, 27, 88, 248, 17, 103, 253, 21, 24, 28, 110, 184, 255, 151, 233, 83, 188, 102, 74, 211, 175, 53, 52, 43, 115, 195, 34, 27, 55, 20, 4, 213, 254, 216, 164, 173, 4
            87, 196, 254, 59, 40, 255, 159, 242, 184, 209, 235, 185, 218, 17, 132, 114, 172, 152, 164, 182, 12, 111, 242, 126, 154, 218, 9, 240, 249, 230, 81, 166, 2, 50, 52, 53, 186, 8, 20
            198, 5, 169, 140, 242, 240, 210, 17, 206, 32, 168, 58, 119, 237, 253, 3, 216, 1, 8, 24, 116, 161, 127, 222, 211, 161, 80, 21, 160,37, 14, 71, 7, 132, 14, 27, 177, 199, 214, 190,
           51, 49, 58, 59, 79, 173, 255, 76, 216, 5, 39, 159, 87, 68, 6, 169, 23, 222, 92, 2, 69, 186, 158, 3, 194, 144, 206, 244, 234, 207, 139, 163, 229, 246, 34, 166, 92, 36, 188, 55, 19
           238, 87, 173, 124, 216, 245, 115, 186, 77, 130, 23, 178, 39, 138, 253, 54, 184, 6, 53, 139, 205, 169, 37, 57, 130, 118, 252, 230, 236, 146, 92, 108, 222, 98, 167, 172, 59, 195, 5
```

**Figure 7: Key Values for Lenna.PNG**

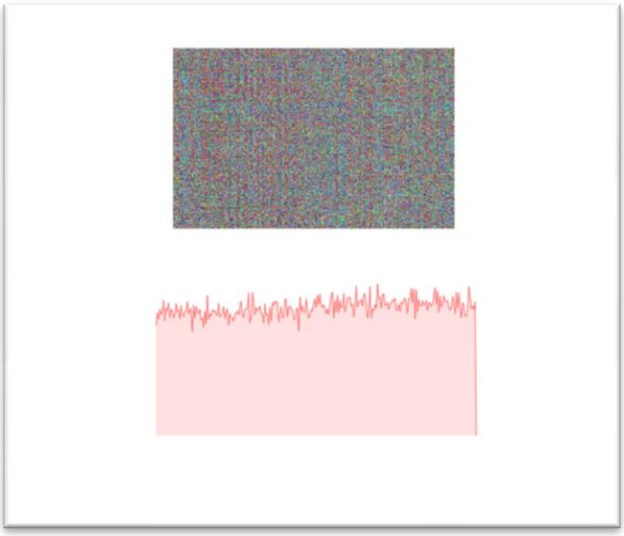


**Figure 8: Encrypted Image and Histogram (Lenna.PNG)**

## V CONCLUSION

In this project we demonstrated a method of combining steganography and image cryptography. This method comprises of first implementing steganography and then that image is subjected to image cryptography. This system can be replaced or used in all cases where steganography and image cryptography is being used. The advantage of this system over using these techniques separately is, if a middle man even gets hold of encrypted image and accurate keys, if the decoded image has a bait text on the image, the stenography text hidden in the image is not visible. Another advantage of this system is if even a smallest of mistake occurs in key values while decryption the output image even when steganography is performed on decrypted image, the plain text cannot be obtained. Pointing this we can say this system can be used for secure communications such as medical and military applications.

## VI REFERENCES

[1] Ahadpour S., Sadra Y., ArastehFrad Z., “A Novel Chaotic Encryption Scheme based on Pseudorandom Bit Padding” IJCSI International Journal of Computer Science Issues, 9(1), 449-456, 2012.

[2] X. Zhou, W. Gong, W. Fu and L. Jin, "An improved method for LSB based color image steganography combined with cryptography," 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, 2016, pp. 1-4.

[3] S. J. Fiona G. Sathiaraj, Grishma S, Pingale, SouvikMajumdar, Saniya J. Shaikh, Bhushan S. Thakare “Secure Transfer of Image-Acquired Text Using a Combination of Cryptography and Steganography” 2019 1st International Conference on Advances in Information Technology.

[4] Ahadpour S., Sadra Y., ArastehFrad Z., “A Novel Chaotic Image Encryption using Generalized Threshold Function” International Journal of Computer Applications, 42(18), pp. 25-31, 2012.

[5] Guoji Zhang, Qing Liu A novel image encryption method based on total shuffling scheme 15 February 2011

[6] Suraj Kumar , Santosh Kumar , Neeraj Kumar Singh , Anandaprova Majumder , Suvamoy Changder5 “ A Novel Approach to Hide Text Data in Colour Image” 2018

[7] Khaled Loukhaoukha, Jean-Yves Chouinard, and Abdellah Berdai “A Secure Image Encryption Algorithm Based on Rubik’s Cube Principle” Hindawi Publishing Corporation Journal of Electrical and Computer Engineering 15 November 2011

[8] Sodeif Ahadpour Yaser Sadra “A Chaos-based Image Encryption Scheme using Chaotic Coupled Map Lattices”

[9] MASOUD ALAJMI, IBRAHIM ELASHRY , HALA S. EL-SAYED , AND OSAMA S. FARAGALLAH “Steganography of Encrypted Messages Inside Valid QR Codes” January 19, 2020.

[10] B. Madhuravani Dr. D. S. R. Murthy Dr.P.Bhaskara Reddy Dr. K.V.S.N. RamaRao “Strong Authentication Using Dynamic Hashing and Steganography” International Conference on Computing, Communication and Automation (ICCCA2015)

[11] SULIMAN A. ALSUHIBANY “Developing a Visual Cryptography Tool for Arabic Text” June 5, 2019

[12] V. Nagaraj, Dr. V. Vijaylakshmi, Dr. G. Zayaraz, “Color Image Steganography based on pixel value modification method using modulus function”, IERI Procedia 4 (2013) 17-24.

[13] Pareek N. K., Patidar V., Sud K. K., “Image encryption using chaotic logistic map,” Image and Vision Computing, 24, pp. 926–934, 2006.

[14] PEI-LING CHIU1, AND KAI-HUI LEE Threshold Visual Cryptography Schemes With Tagged Shares May 11, 2020.