Chapter 1

Introduction

One of the reason that meddlers can be successful is most information they acquire is in the form that they can read and encompass. Meddlers or intruders may modify the information and reveal it to an individual or organization. They may also use it to launch an attack.One solution to this problem is through “STEGANOGRAPHY”.

Steganography is the art or technique of hiding information. Steganography word is the combination of two Greek words “Stego” meaning cover and “grafia” meaning writing. In contrast to cryptography, it is not keep people from knowing the hidden information but it is to keep people from thinking that the information even exists. Therefore, steganography is the art of concealing the information that prevents revelation of hidden message. For better security, both steganography and cryptography are combined.

In ancient time, steganography was used in Greece for the first time in 5th century. Greek people used to hide the information on the head of their slave. In the same time of Greece, Spartans used wood wax tablet and covered a new plane layer of wax on it to hide the information. During World War 2, information was written on paper using invisible ink or microdots were used to write secret message which was hidden in innocent message.

Steganography becomes of greater significance in the digital era as more people are joining cyber space revolution. Computer network requires special means of security as the number of data being exchanged on the internet is increasing. Therefore, confidentiality and data integrity plays a major role to protect against unauthorized access. Information hiding is an emerging research area in modern communication. This includes applications such as watermarking, fingerprinting, copyright protection and steganography.

The most popular cover media for steganography are the images. One of the popularly accepted technique of steganography is least significant bit [LSB].It has its origins dating back to the 17th century. Least significant bit or bits of pixel are replaced by the bits of data to be hidden in the so called LSB method.The most popular cover media for steganography are the images. One of the popularly accepted technique of steganography is least significant bit [LSB].It has its origins dating back to the 17th century. Least significant bit or bits of pixel are replaced by the bits of data to be hidden in the so called LSB method.

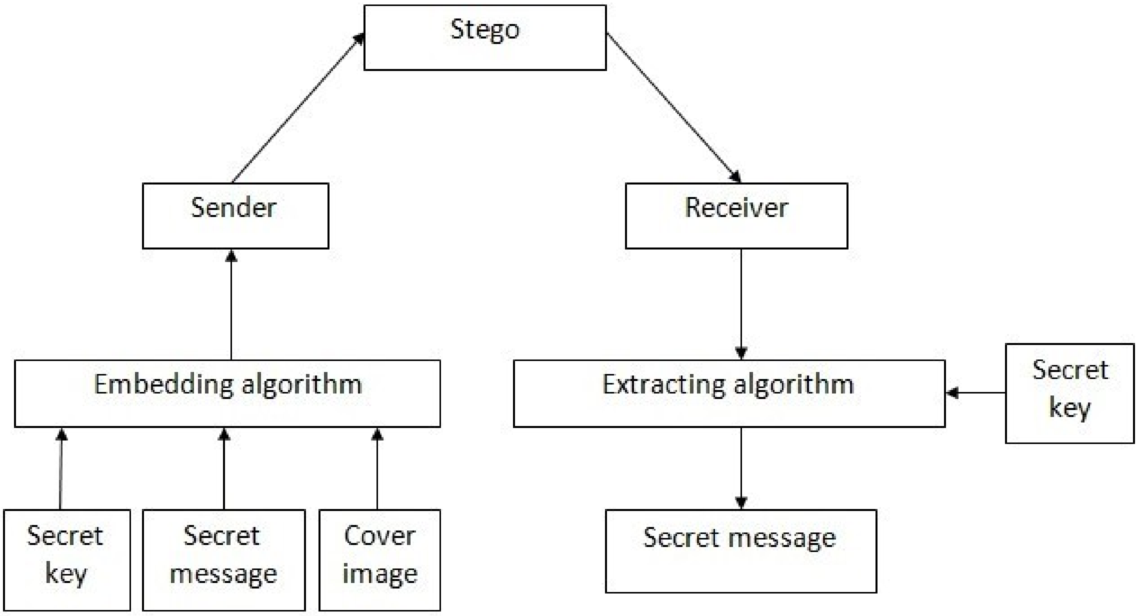


Figure 1.1: Steganography System Scenario

1.1 Classification of Steganography

Steganography is classified into 3 categories:
i. Pure Steganography: In this there is no stego key. It is based on no other party is aware of communication.
ii. Secret key Steganography: The stego key is exchanged prior to communication between two parties.
iii. Public key Steganography: In this approach a public and a private key is used for secure communication.

1.2 Classification based on Domain

Steganography is divided into two main categories:
1. Spatial domain approach: The secret message is directly hidden by modifying the pixels of the image. Examples are LSB technique, watermarking etc.
2. Transform domain approach: The secret message is indirectly embedded by taking transforms such as DCT, DWT etc.

1.3 Steganography types based on cover medium

Almost all digital formats can be used for Steganography. It can be divided into 5 main categories.
i. Text steganography: The most common method of steganography is in text file for hiding information. The method is to hide a secret message into a text message. Text steganography using digital files is not used very often because text file has a very small

amount of excess data.
ii. Image steganography: Images are used as a popular cover medium in which a message is embedded using embedding algorithm. Image steganography is commonly used in websites-mail attachments where the images are of great importance.
iii. Audio steganography: It is concerned with embedding information within a speech in a secure and robust manner. It involves digital representation of sound in the form of intensity at a certain point of time.
iv. Video steganography: It is a technique to hide any kind of files in any extension in a video file. It is generally collection of image and sound steganography and all the property of image and sound is applied in video steganography. Here too in video steganography large amount of data can be embedded.
v. Protocol steganography: It is a technique to embed information within network protocols such as TCP/IP.

1.4 Steganography Properties

Any Steganography system model must satisfy the following properties:
i. Invisibility: With naked eye view the secret data mustn't be invisible.
ii.Capacity: A cover image must hold more embedded secret data and the quality of image should not be degraded.
iii. Robustness: The stego image should hold the secret data even after some noise gets added to it.

1.5 Motivation

Steganography provides hiding sensitive or private information within an image so as it appears no information is hidden at all. If a person views an image, the information hidden inside it cannot be perceived by human eye, therefore the person will not attempt to decrypt the information. Image steganography using LSB technique is used as the changes in the cover and the encrypted image is invisible to human eye, as the process is done in spatial domain. Steganography find application in various domain like military, medical, copyright and one time password etc.

1.6 Problem Statement

Steganography Technique Using AES Algorithm.

1.7 Objectives

The main objectives of our project :
- To hide the text or image inside the other image, text using LSB technique and Array Method.
- To provide security for the hidden information using AES Algorithm.

1.8 Scope

The project is developed for hiding secret information either in the form of text or images in an image. The project implementation is using of steganography tool for hiding information which includes any type of text or image files and the path where the user wants to save image or extrude file.

1.9 Outcome of project

The main outcome of the project is to retrieve back the data i.e. the text and the image from the cover image.

1.10 Methodology

• Technique used is LSB [Least significant bit] which uses LSB encoder and LSB decoder.
• Advance Encryption Standards [AES] encryption and AES decryption for more security with the key length of 256 bit.
• Implementation of the Python code using LSB and AES algorithm.
• Retrieving back the text or image from the image.

Chapter 2

Literature survey

The purpose of literature survey is to identify the information relevant to “Advanced Encryption Standard [AES] and Least Significant Bit [LSB] technique”.

Gurpreet Singh and Supriya[1] proposed a paper “A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for information security” in which it is described that asymmetric algorithms such as RSA(Rivest–Shamir–Adleman) are slower than that of symmetric algorithm and RSA is least secure as compared to DES(Data Encryption Standard), 3DES and AES(Advanced Encryption Standard). The paper presents detail about popular Encryption algorithm. According to the details provided in the paper, AES algorithm is most efficient in terms of speed, time, throughput and avalanche effect.

Qi Zang and Qunding[2] proposed paper by name “Digital Image Encryption based on Advanced Encryption Standard [AES] algorithm” which puts forward the usage of AES algorithm in Python with the key control to encrypt the image. According to the result and analysis, coupled with the histogram and key sensitivity analysis, AES can achieve very good effect on image encryption and, the decryption essence has same structure with the encryption, so it can easily restore the original image. The infrastructure of the AES algorithm uses matrix as basic unit, therefore Python software tool which has powerful numerical calculation function, especially for arrays and matrix calculation is used in this paper as it is easier to implement.

Harpreet Kaur and Ajay Kakkar[3] proposed a paper “Comparision of different image formats using LSB steganography” in which LSB substitution scheme is used for embedding any secret data into gray scale images with different formats. The result in the paper shows that the stego image is obtained without making a perceptible distortion. Comparison of various images with different formats and their PSNR and MSE values are calculated and described in the paper. S.M Masud Karim, Md.Saifur Rahman, Md.

Ismail Hossain[4] presented a paper “A new approach for LSB based image steganography using secrete key” in which a secret key is used to hide the information in cover image reporting without significant distortion and it is very difficult for unauthorized users to identify the changes in stego image. The use of secret key gives a way to secure the information from illegal user and provides better PNSR value.

Aman Arora, Manish Pratap Singh, Prateek Thakral, Naveen Jarwal[5] proposed a paper by name “Image steganography using Enhanced LSB substitution technique” in which the embedding capacity is increased to significant levels. In this paper the data is first encrypted and then embedded in to the cover image, therefore it offers double security because the data is encrypted first. The base algorithm used in this paper is LSB substitution technique.

Priya Deshmukh[6] presented a paper on “An image encryption and decryption using AES algorithm” in which Advanced encryption standard is used to secure the data from unauthorized access. With the help of Python coding, implementation of AES and simulation for image encryption and decryption is performed. The results in the paper show that the AES algorithm have extremely large security key space and can withstand most common attacks by intruders.

Chapter 3

Methodology

3.1 Design Methodology And Implementation

The chapter deals with method used for implementing image steganography using LSB and AES algorithm. The two algorithms used are:
1. Least Significant Bit (LSB)
2. Advanced Encryption Algorithm (AES)

3.2 Text Inside Image

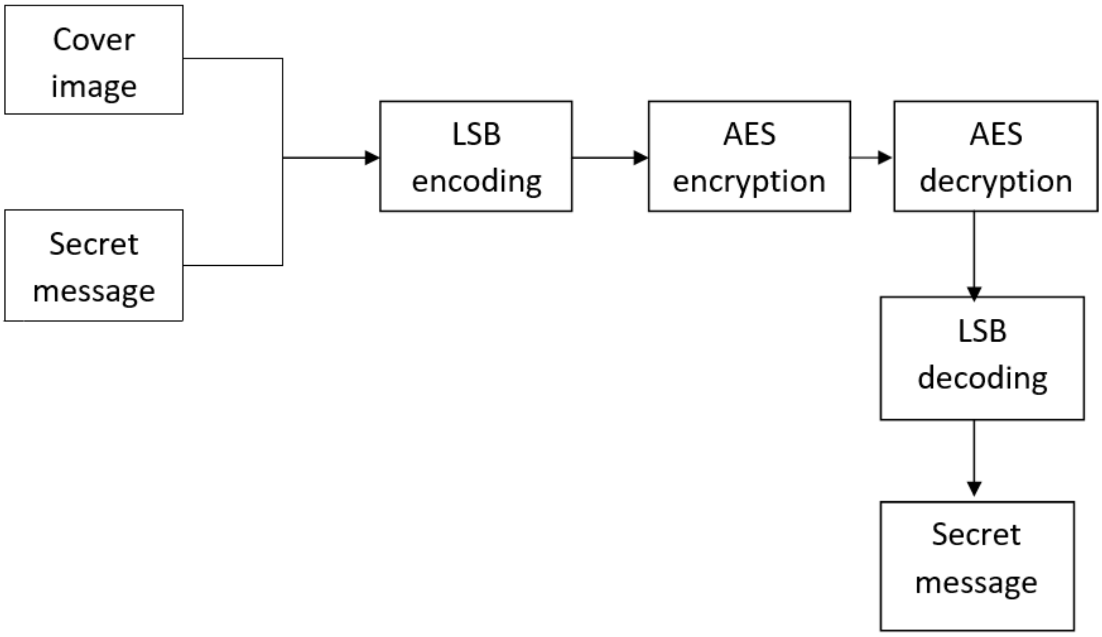


Figure 3.1: Block Diagram of combined LSB with AES algorithm of Text inside Image

The general block in Fig tells about how the secret data is hidden in the cover image and extracted by providing security. The stage involved in this are:
• Input the cover image. If it is a RGB image, convert it into grey scale image and resize it to a particular size of 320*175.

• Input the secret data to be hidden. Hide the data using LSB encoder in which the least significant bit is of the cover image is substituted.
• Apply AES encryption to the cover image embedded with the hidden data.
• To the obtained AES encrypted image, perform AES decryption.
• Extract the data embedded using LSB decoding technique. The size of the original image is of 320x175 where 320x175 gives the number of rows and columns with the total size of the image to be 55999.The text and the cover image are given to the LSB encoder where the LSB encoder hides each of the bit of the character into least significant bit of the cover image. The process continues till every character gets substituted in the pixel value of the original image. The resultant image formed is the stego-image. Here in the next step the stego-image is encrypted with AES algorithm to provide the security for the hidden text inside the cover image. Once the encrypted image is formed the stego image needs to be retrieved which is dine with the help of AES decryption. The function of the LSB decoder is to retrieve back the text from the image. The final output is the secret text.

3.3 Image Inside Image

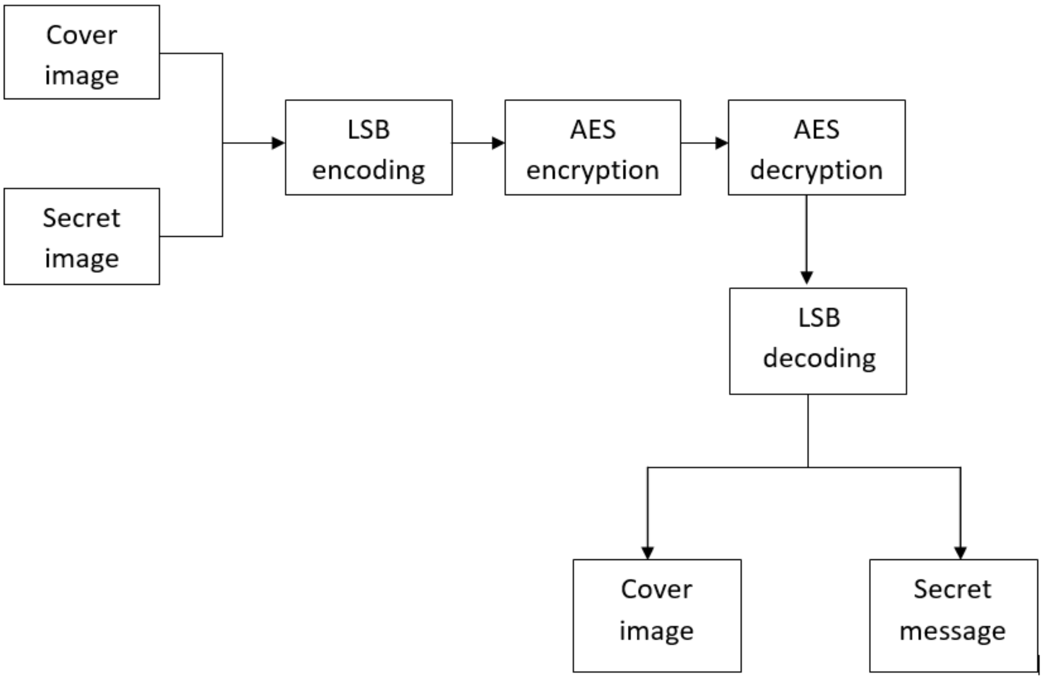


Figure 3.2: Block Diagram of combined LSB with AES algorithm of Image inside Image

3.4 Least Significant Bit

Least Significant Bit technique is one of the most popular existing spatial techniques used in steganography. Our work focuses on LSB embedding with gray scale images in which large amounts of data can be embedded without observable changes. The technique works by replacing some of the information in a given pixel with information from data in the least significant bit. Since all digital information are stored in bits, any data can be embedded bit-by-bit into the pixels of least significant bit as a series of zeros and ones. It requires a much larger cover, since a single gray pixel in the data to embed would be spread over eight pixels in the cover image. LSB embedding approach has become the basis of many techniques that hide messages within multimedia carrier data. First subsection explains the encoding process used i.e. it deals with the information regarding hiding data inside image. Second subsection explains the decoding process i.e. extraction of the data from the image. LSB technique comprises of
LSB Encoder
LSB Decoder

3.5 LSB Encoding Process

1) Input the cover image [original image]. Each of the pixel value of cover image is converted to binary form.
2) Input the secret data which is to be hidden and convert the data to binary form.
3) Apply LSB encoder.
4) Convert the obtained result into pixel values.
5) Stego-image is the output image.

3.6 Block diagram for encoding process

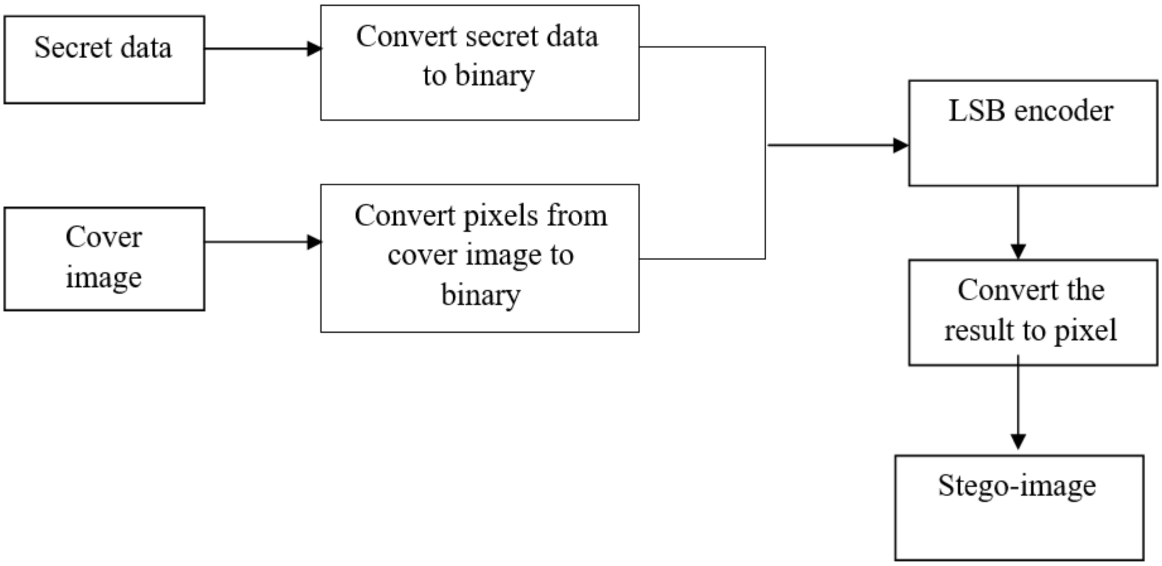

Secret data
Convert secret data to binary
Cover image
Convert pixels from cover image to binary
LSB encoder
Convert the result to pixel
Stego-image

Figure 3.3: Block diagram of LSB Encoder

From the fig 3.3, the original image and the secret data either text or the image are first converted to binary representation in form of ones and zeros. The resultant is given to the LSB encoder where the function of the encoder is to hide the data in the least significant bit of the pixel values of the cover image. The binary format of the pixel values are again converted into the pixel values to get the stego image.

3.7 LSB Decoding Process

1. The output stego image is considered and the pixel values of the image are converted to binary representation.
2. Apply LSB decoder.
3. Convert the binary values back to obtain the data.
4. Convert the binary value of the image into pixels.
5. Obtain the secrete data and the cover image at the output.

3.8 Block Diagram For Decoding

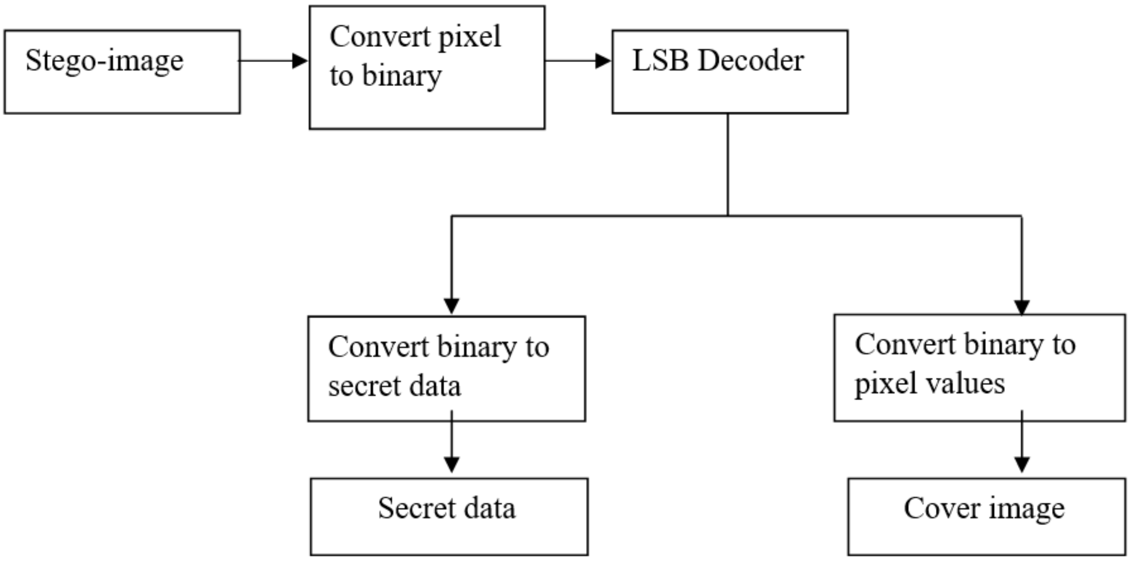


Figure 3.4: Block diagram of LSB Decoder

3.9 LSB Method with Examples

LSB replacement steganography changes the last bit of each pixel values to reflect the message that needs to be hidden. Suppose the first eight pixels of the cover image have the following gray color values:
10011100
10011111
10011110
10011011
10011110
10011100
10011111
10011110
To hide letter 'h' which has ASCII value of 104, we would replace the LSB's of these pixels to have the following new values:
10011101
10011111
10011110
10011011
10011110
10011100
10011110

3.10 Encryption Algorithm

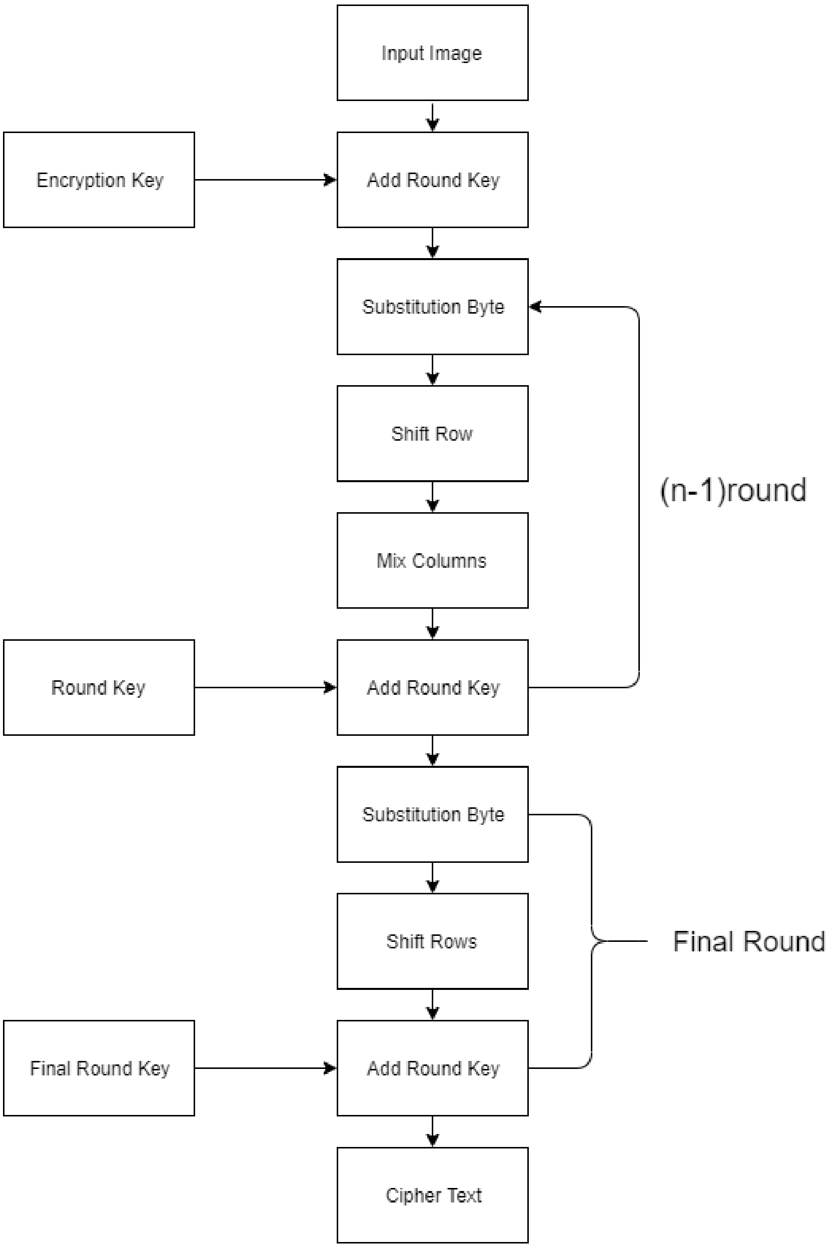


Figure 3.5: AES Encryption Algorithm

3.11 Decryption Algorithm

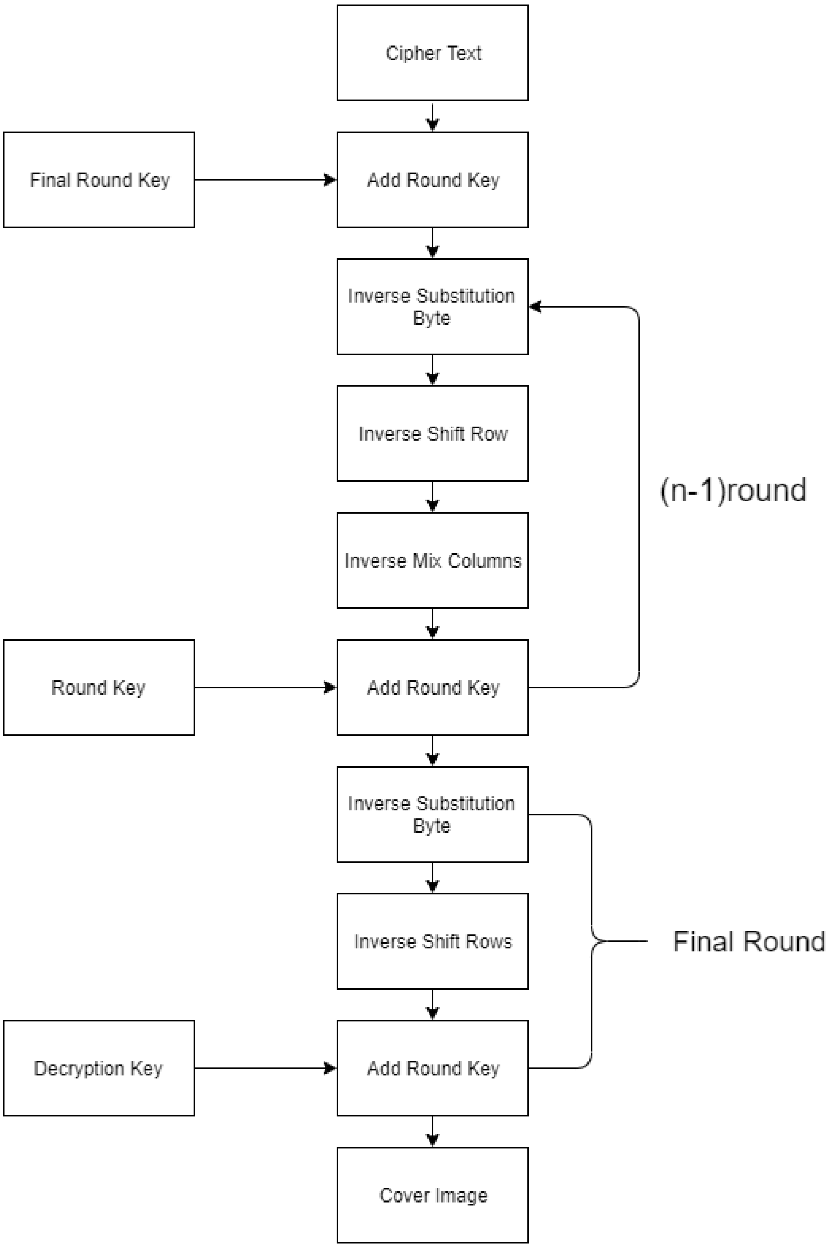


Figure 3.6: AES Decryption Algorithm

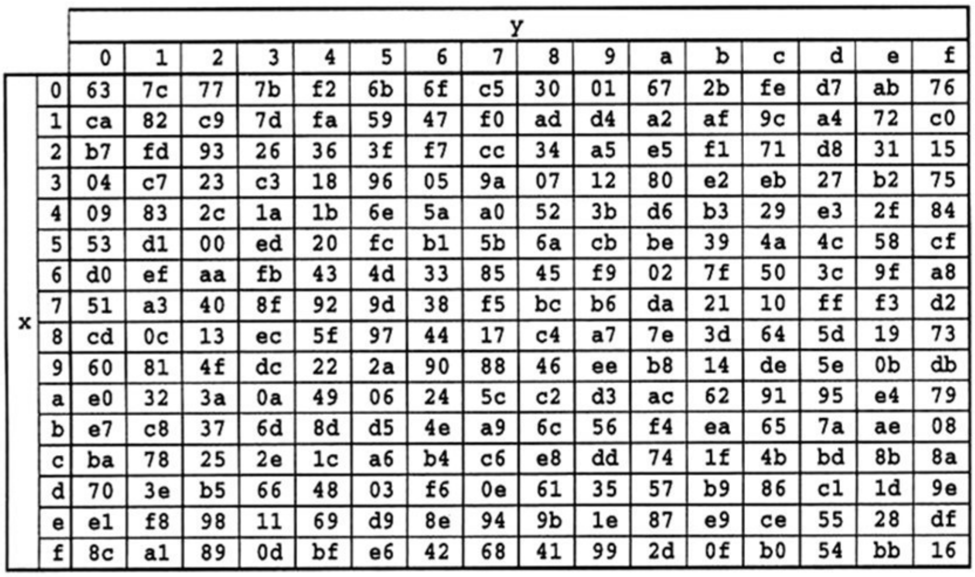
3.12 Advanced Encryption Standard

National Institute of Standard and Technology (NIST) selected Rijndael as the proposed AES algorithm in the year 2001. Dr. John Daemon and Dr. Vincent Rijmen are the two researchers who developed and submitted Rijndael for the AES. AES is used instead of DES and is one of the most popular symmetric algorithms. The block length is 128 bits with 128, 192 or 256 bit key length. The input is a single 256 bit block for encryption and decryption algorithm, frequently used key length of 128 bits. The AES algorithm uses a round function that is composed of four different byte orientation transformations. Encryption uses four stages that include:
i. Substitution byte
ii. Shift rows
iii.Mix columns
iv.Add round key

Decryption is the reverse process of encryption which uses:
i.Inverse shift rows
ii.Inverse sub bytes
iiiInverse mix columns
iv.Inverse round key

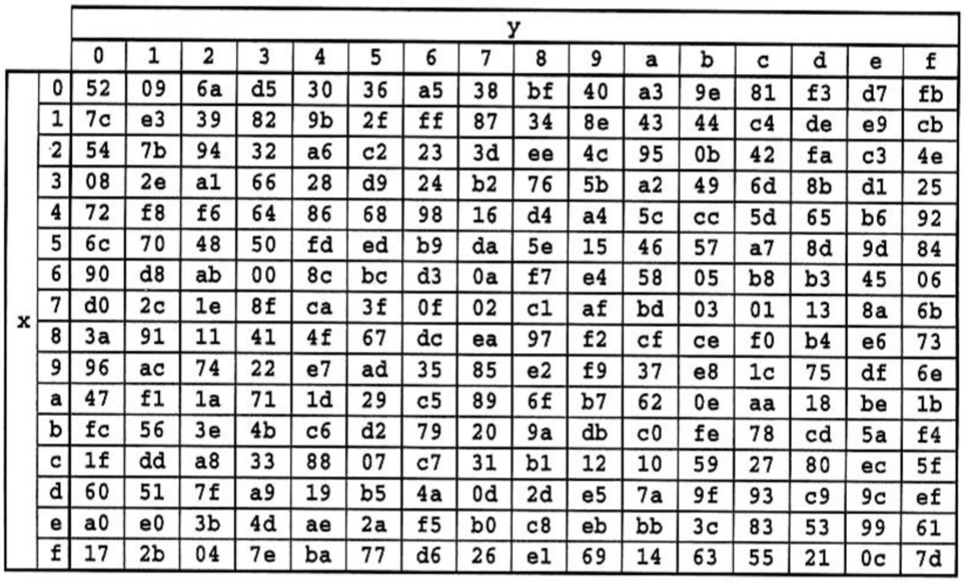
3.13 Substitution Byte Transformation

The forward substitute byte transformation, called Sub Bytes is a simple lookup table. AES defines a 16 X 16 matrix of byte values, called an S-box that contains a permutation of all possible 256 8-bit values. It uses this Sbox which is constructed by multiplicative inverse and affine transformation to perform byte to byte substitution of the block .The inverse substitute byte transformation, called InvSubBytes, makes use of inverse S-box.



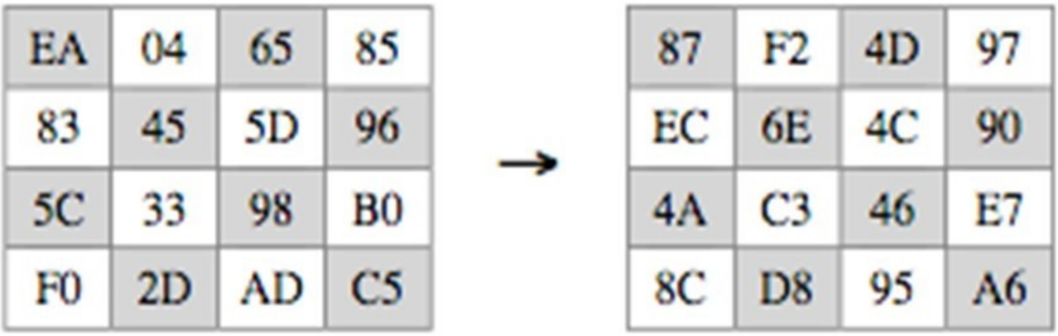
		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 3.7: S-box look up table



		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figure 3.8: Inverse S-box look up table



EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

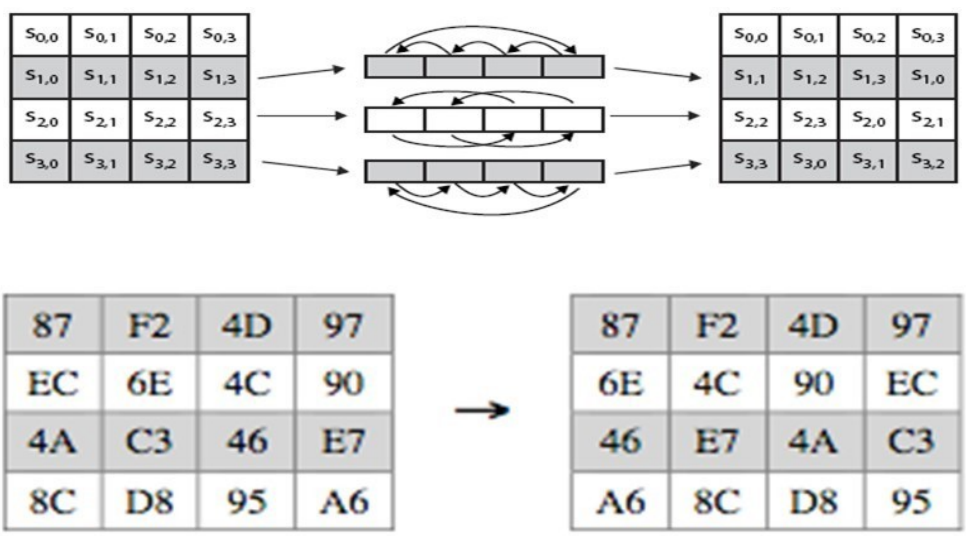
→

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

Figure 3.9: Substitute byte transformation

3.14 Shift Row Transformation

It is a simple byte transformation. For the Forward shift row transformation, called Shift Rows the bytes in the last three rows of the state are cyclically shifted i.e., the first row of the state is unaltered, for the second row 1-byte circular left shift is performed and for third row 2-byte circular shift is performed and for the last row 3-byte circular left shift is performed. For Inverse shift row transformation, called Inverse Shift Rows, the circular shifts in opposite direction for each of last three rows, with a one-byte circular right shift for second row, and so on are performed.



87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

Figure 3.10: Shift Row Transformation

3.15 Mix Column Transformation

The forward mix column transformation is called Mix Columns and inverse mix column transformation, is called InvMixColumns. It operates on every column independently. Each byte of a column is mapped into a new value that is a function of all four bytes in that column. The substitution makes use of arithmetic over GF(28). It is designed as a matrix multiplication where each byte is treated as a polynomial in GF(28). The inverse used for decryption involves a different set of constants. This gives good mixing of the bytes within each column. Combined with the "shift rows" step provides good avalanche, that reflect within a few rounds, all output bits depend on all input bits.

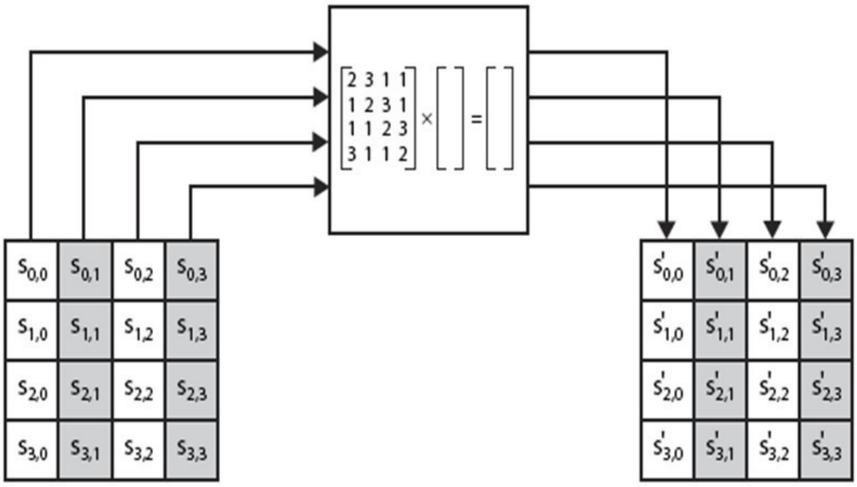
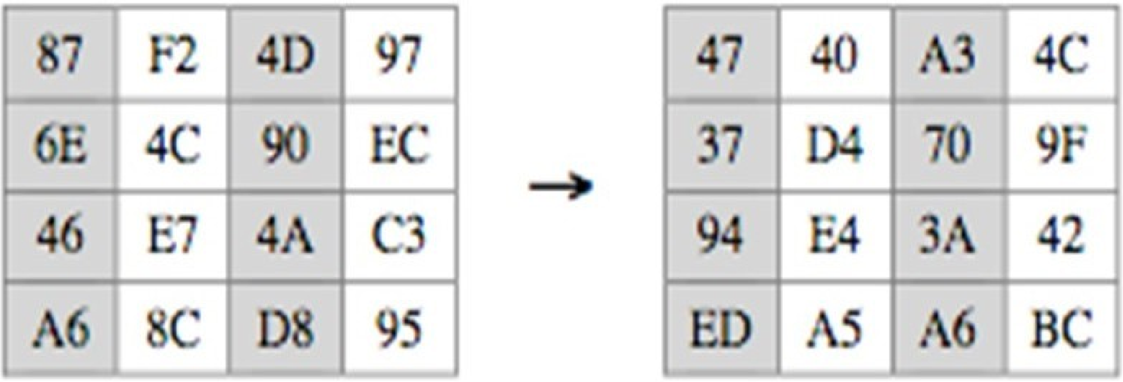


Figure 3.11: Mix column Transformation

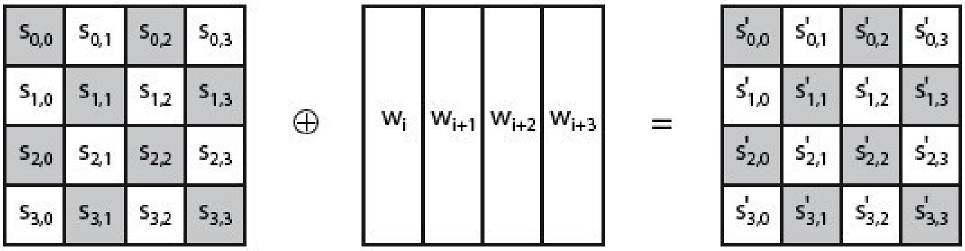


$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix} \rightarrow \begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix}$$

Figure 3.12: Example of Mix Column Transformation

3.16 Add Round Key Transformation

In the forward add round key transformation, called Add Round Key, the 128 bits of state are bitwise XORed with 128 bits of round key. The inverse add round key transformation is performs inverse operation of the mentioned.



$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} \oplus \begin{bmatrix} w_i & w_{i+1} & w_{i+2} & w_{i+3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Figure 3.13: Add Round Key

3.17 AES Expansion

The AES key expansion algorithm takes as input a 4-word (16byte) key and gives a linear array of words, providing a 4-word round key for the initial AddRoundKey stage and each of the 10 rounds of the cipher. It involves copying the key first in to the group of 4 words, and then constructing subsequent groups of 4 based on the values of the previous 4th words. The first word in each group of 4 gets "special treatment" with rotate + S-box + XOR constant on the previous word before XOR'ing the one from 4 back.

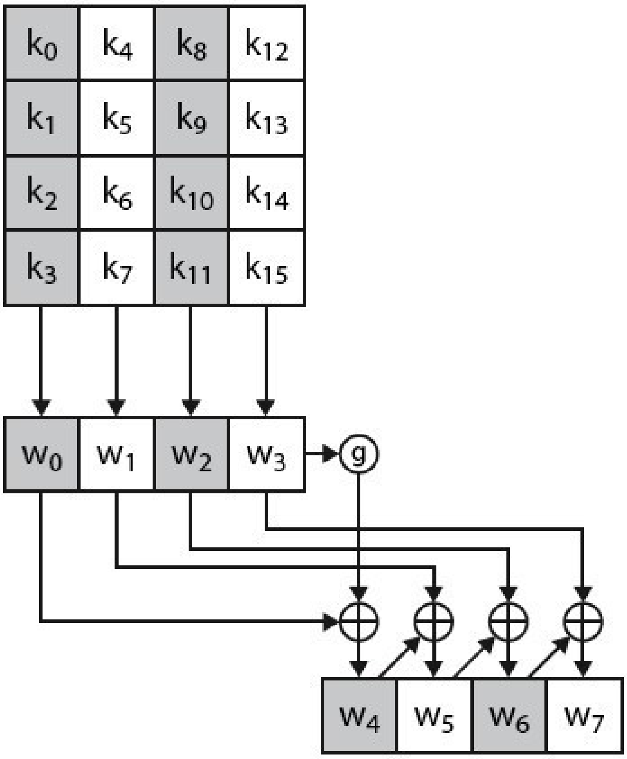


Figure 3.14: AES Key Expansion

For AES encryption in Fig 3.13, the plaintext i.e. the 16 byte value of the cover image (128-bit) arranged in 4x4 matrixes is xor-ed with the 128 bit key length. The resultant output of 4x4 matrixes is the cipher text which further undergoes through all the steps of AES. The cipher text is first given to substitute byte where each byte of the cipher text is substituted from the standard s-box. The resultant output is given to the shift rows, mix columns and the add round key. At the start the same key is used of one's choice and in the next subsequent round the key is expanded with the help of key expansion. The process continues till 9 rounds and at the 10 th round the mix column is excluded to get the encrypted form of image. In AES Decryption in Fig 3.14, the encryption process repeats in reverse order to get back the stego image.

3.18 Estimating the PSNR RATIO and MSE

The measurement of quality between the cover image and stego image of size m x n is given by PSNR as:

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

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

MAXI=Maximum value of pixel in Original image
m=No. of Row in cover image
n= No. of Column in cover image
Where I (i, j) and K (i, j) represents pixel value at position (i, j). PSNR is expressed as DB.

PSNR is the representation of quality of image i.e., higher the PSNR, lower is the difference between cover and stego image.

Chapter 4

Optimization

In computing,the word optimization is the process of altering a system to make some applications of it work more efficiently or use less resources. For such instance, a computer program may be optimized so that it runs faster, or to run with less memory requirements or other resources.

The optimization can have sense at different levels, from the lowest up to the highest levels of making of implementation, use or design of algorithms.

The optimization technique is generalized to leave until the end of the process of development, since the premature optimization can introduce new errors.

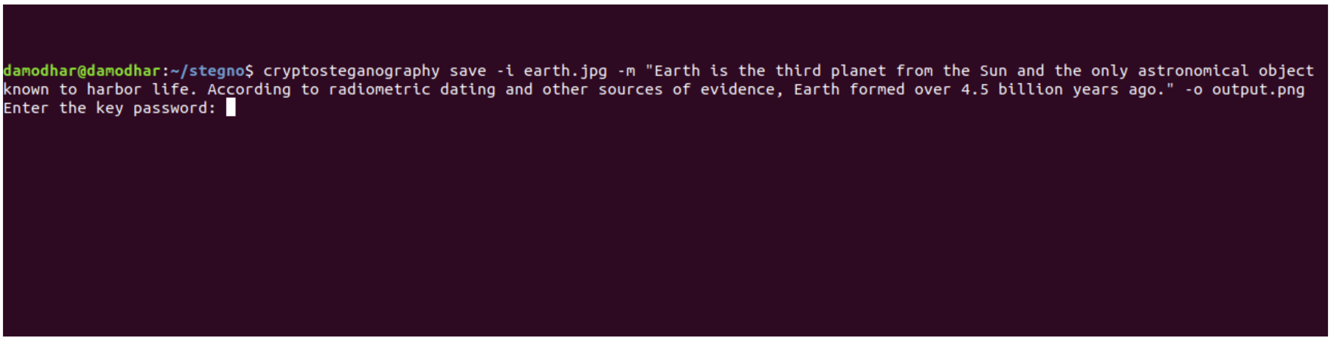
The optimized system may be a single computer program, a collection of computers or even an entire network such as the Internet.Thus the optimization techniques used in our project are:
1.List Comprehension
2.Dictionary Comprehension
3.Using imports properly
4.Using ‘try except’ Instead of ‘if else’
5.Modular Approach
6.Defining functions

Chapter 5

Results

5.1 Text Inside Image

For text steganography, each bits of the character are being substituted in the least bit of the pixel values of the image using LSB Technique. Here we consider two cases i.e.:
Case 1: Embedding the bits in column
Case 2: Embedding the bits in row



```
damodhar@damodhar:~/stegno$ cryptosteganography save -i earth.jpg -m "Earth is the third planet from the Sun and the only astronomical object
known to harbor life. According to radiometric dating and other sources of evidence, Earth formed over 4.5 billion years ago." -o output.png
Enter the key password: █
```

Figure 5.1: Terminal Window Before Hiding Text inside the image

1.The Command used for the encryption of the text in an image is: cryptosteganography save -i(cover image) -m(secret message) -o(stego image)
2.Further it asks for the key password for the secret message.

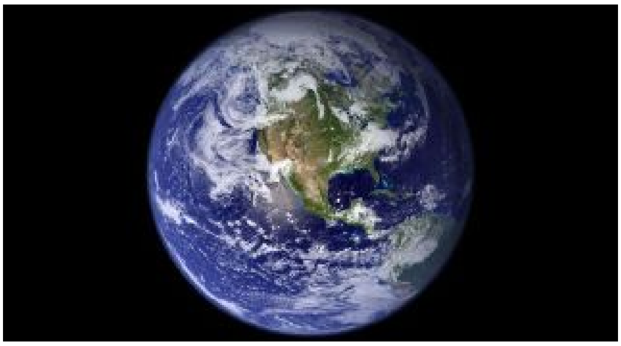
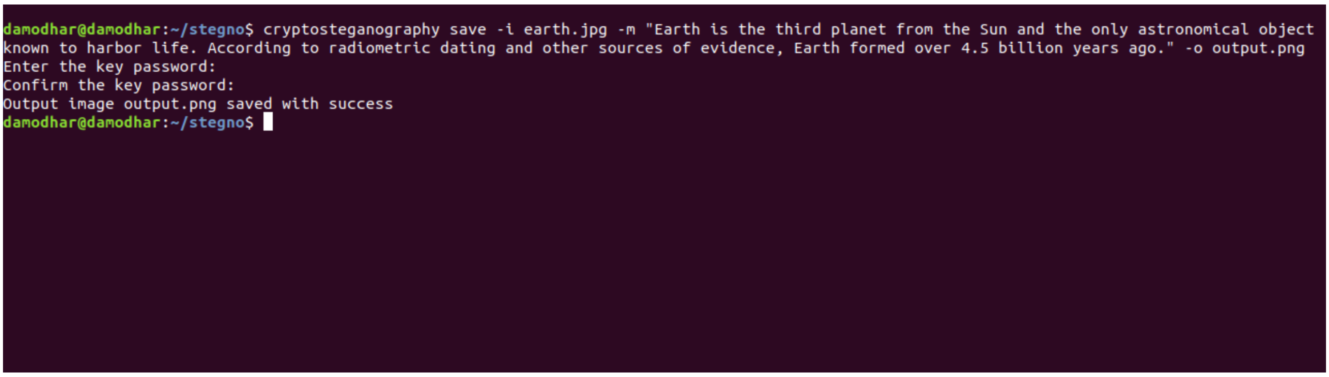


Figure 5.2: Cover Image

Original image of 320x175 is given as an input and the secret text is embedded into original image using LSB technique and in order to provide security for the hidden image AES encryption and decryption is applied and finally the secret text is retrieved back.



```
damodhar@damodhar:~/stegno$ cryptosteganography save -i earth.jpg -m "Earth is the third planet from the Sun and the only astronomical object
known to harbor life. According to radiometric dating and other sources of evidence, Earth formed over 4.5 billion years ago." -o output.png
Enter the key password:
Confirm the key password:
Output image output.png saved with success
damodhar@damodhar:~/stegno$ █
```

Figure 5.3: Terminal Window After Hiding Text inside the image

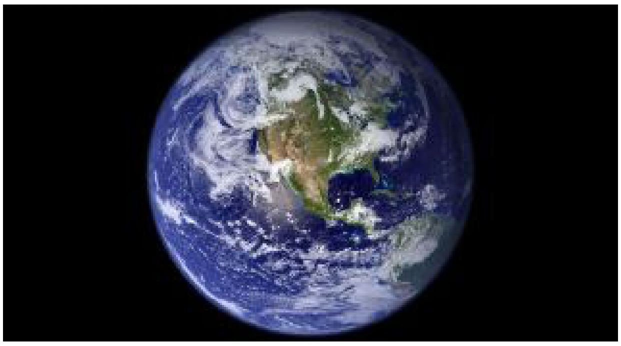
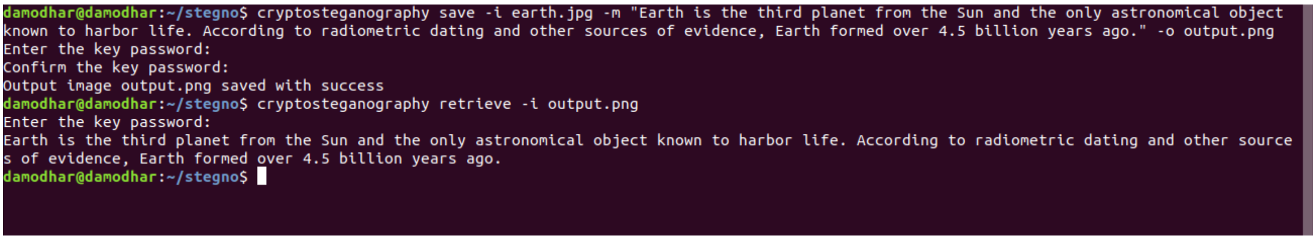


Figure 5.4: Stego Image



```
damodhar@damodhar:~/stegno$ cryptosteganography save -i earth.jpg -m "Earth is the third planet from the Sun and the only astronomical object
known to harbor life. According to radiometric dating and other sources of evidence, Earth formed over 4.5 billion years ago." -o output.png
Enter the key password:
Confirm the key password:
Output image output.png saved with success
damodhar@damodhar:~/stegno$ cryptosteganography retrieve -i output.png
Enter the key password:
Earth is the third planet from the Sun and the only astronomical object known to harbor life. According to radiometric dating and other source
s of evidence, Earth formed over 4.5 billion years ago.
damodhar@damodhar:~/stegno$ █
```

Figure 5.5: Decryption of the secret message from the stego image

1.The Command used for the decryption of the text in an image is: cryptosteganography retrieve -i(stego image)
2.Further it asks for the key password for the secret message.
3.If the entered key password is invalid then it displays the message that you have entered an invalid key password.Otherwise it will display the secret message.

5.1.1 Histogram Plot

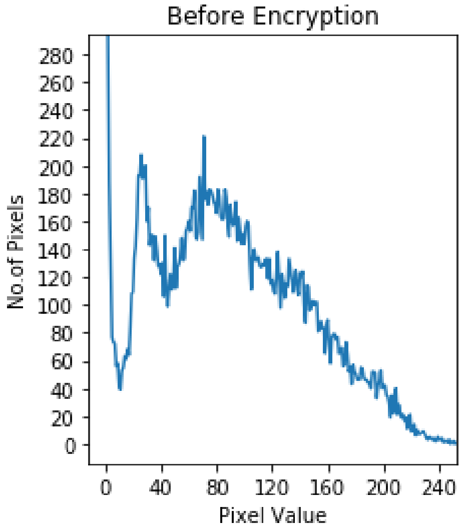


Figure 5.6: Histogram Plot of the Cover Image(4.2)

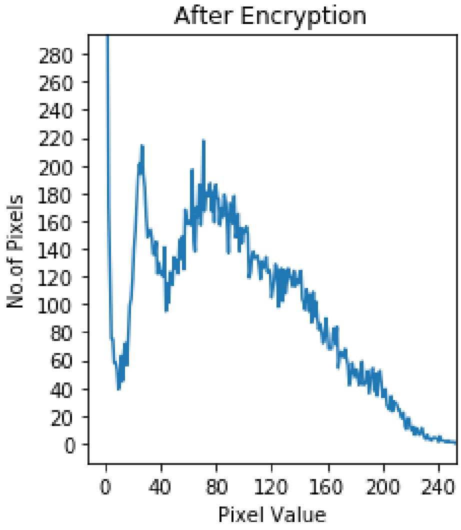


Figure 5.7: Histogram Plot of the Stego Image(4.4)

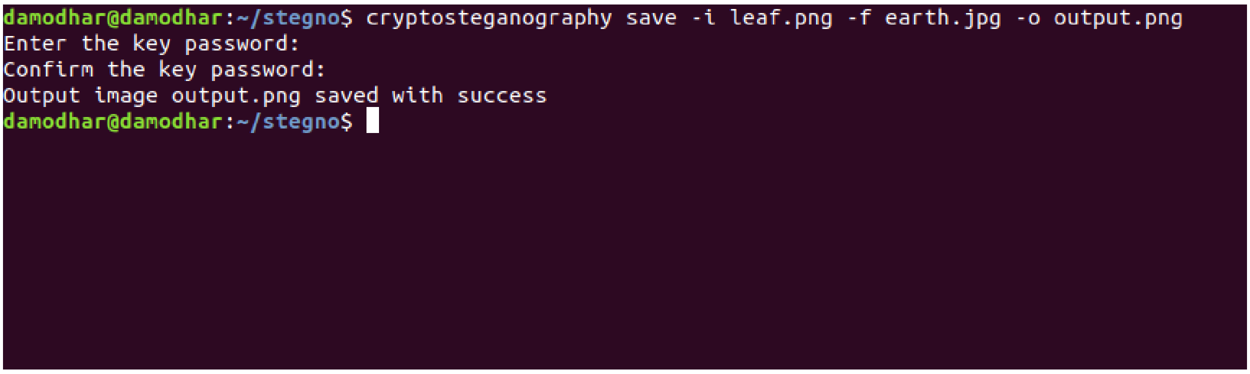
Histogram plot for both cover image and stego image is plotted. From the above fig 4.6 and 4.7 there is not much variation in the pixel value of image as LSB substitution is for 1-bit data.

5.1.2 PSNR and MSE ratio

The PSNR ratio calculated is 50.113990368159335dB and the MSE ratio is 0.6334047619047619.

5.2 Image Inside Image

Here the Cover Image is of length of size 360x360 the Secret Image to be embedded is of size 320x175. Thus the cover image is should be is of the size greater than the 2 times of the cover image.



```
damodhar@damodhar:~/stegno$ cryptosteganography save -i leaf.png -f earth.jpg -o output.png
Enter the key password:
Confirm the key password:
Output image output.png saved with success
damodhar@damodhar:~/stegno$ 
```

Figure 5.8: Terminal Window Before Hiding image inside the image

1.The Command used for the encryption of the image inside another image is: cryptosteganography save -i(cover image) -f(secret image) -o(stego image)
2.Further it asks for the key password for the secret message.
3.The stego image generated will be saved successfully.And hence it can be decrypted further.

Figure 5.9: Cover Image

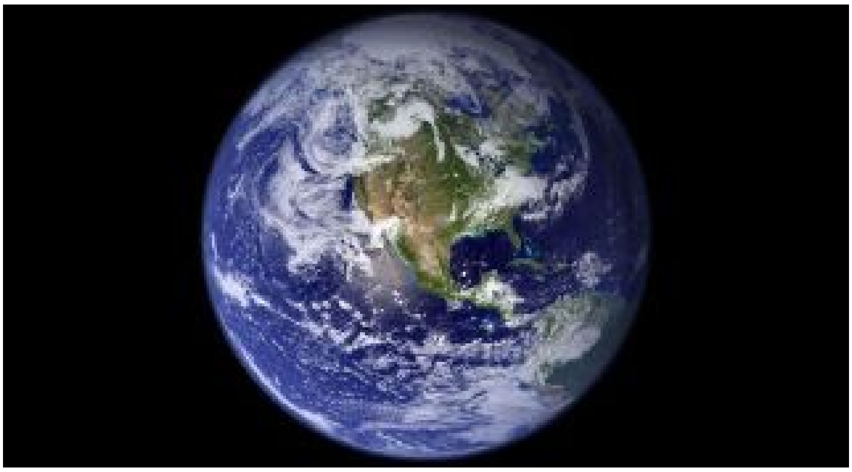


Figure 5.10: Secret Image

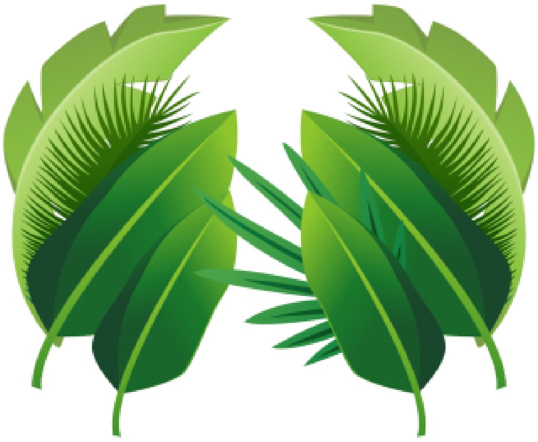


Figure 5.11: Stego Image

5.2.1 Histogram Plot

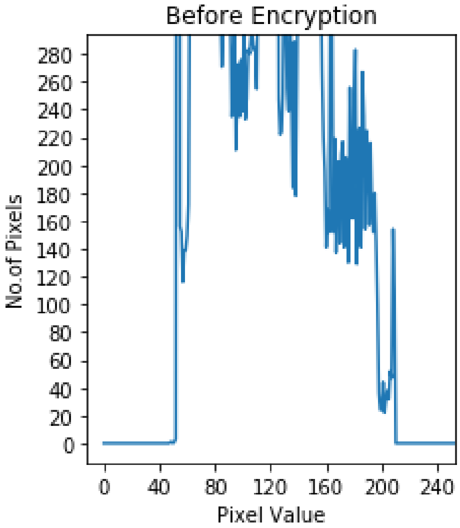


Figure 5.12: Histogram Plot of the Cover Image(4.9)

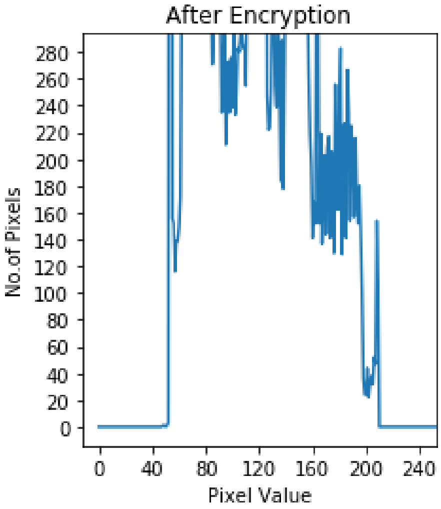


Figure 5.13: Histogram plot of the Stego Image(4.11)

For image inside image histogram plot in fig 4.12 and 4.13 we can infer that there is not much variation seen in the pixel value of cover and stego image.

5.2.2 PSNR and MSE ratio

The PSNR ratio calculated is 56.78113042449563dB and the MSE ratio is 0.13644804526748971.

Chapter 6

Conclusion And Future Work

In our approach, Least Significant Bit (LSB) technique is used to hide sensitive data i.e, text or image in the cover image which do not visually degrade the image to the point of being noticeable at the output. For the security, AES is used at both the ends i.e., sending and receiving ends. Both steganography and cryptography are combined together to achieve desired results. PSNR and MSE is calculated for the cover image and stego image and tabulated for different image formats.AES is preferred over DES due to simplicity and its speed.

6.1 Advantages

Used scheme provides good balance between embedding capacity and quality of stego image.
1. The picture quality of the cover image is hardly affected.
2. Increased embedding capacity.
3. Difficult to detect.
4. Protects message from attacks because of more security provided by AES.
5. No intruder can get any useful information from the original file during transmit.

6.2 Applications

Feature tagging elements can be embedded inside an image, such as names of the individuals or name of the location.
1. In Military applications to communicate and co-ordinate about attacks.
2. It does not advertise secret communication and therefore avoids scrutiny of sender and recipient.
3. Can be used in smart identity cards where the information of the person is secretly stored in image of the person itself.
4.To hide one-time password in images that are stored in mobile devices.

6.3 Future Work

1.The data can be embedded and analysis to audio and video files.
2.Image steganography with other techniques and algorithm can be implemented.

Chapter 7

Bibliography

1. Gurpreet Singh and Supriya proposed a paper "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for information security".
2. Qi Zang and Qunding proposed paper by name "Digital Image Encryption based on Advanced Encryption Standard [AES] algorithm",2015 fifth international conference on instrumentation and measurement, computer, communication and control.
3. Harpreet Kaur and Ajay Kakkar proposed a paper "Comparision of different image formats using LSB steganography", 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC).
4. S.M Masud Karim, Md.Saifur Rahman, Md.Ismail Hossain presented a paper "A new approach for LSB based image steganography using secrete key",14 th international conference on computer and information technology, December,2011.
5. Aman Arora, Manish Pratap Singh, Prateek Thakral, Naveen Jarwal proposed a paper by name "Image steganography using Enhanced LSB substitution technique",2016 fourth international conference on Parallel, Distributed and Grid Computing.
6. Priya Deshmukh presented a paper on "An image encryption and decryption using AES algorithm", International Journal of Scientific Engineering Research, Volume 7, Issue 2, February-2016.
7. TanmyBhaowmik, PramathaNathBasu, "On Embedding of text in Audio – A case of Steganography", International Conference on Recent Trends in Information, Telecommunication and computing, IEEE 2010.
8. Ashis Kumar Mandal, Mohammed Kaosar, Md. Olioul Islam and Md. DelowarHossain, "An Approach for Enhancing Message Security in Audio Steganography", IEEE 16th International Conference on Computer and Information Technology, 8-10 March, 2014.
9. JithuVimal and Ann Mary Alex, " Audio Steganography Using Dual Randomness LSB Method" , IEEE International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014.
10. Sumit Kumar Moudgil, Dr. U Ragavendran "Effective Use of Steganography on Audio Wave and Spy Analysis" International Journal of Electronics and

Communication Engineering Technology (IJECET), Volume 7, Issue 4, July-August 2016, pp. 32-39; ISSN Print: 0976-6464 and ISSN Online: 0976-6472; Journal Impact Factor (2016): 8.2691; InfoBase Index IBI Factor for the year 2015-16 is 3; Thomson Reuters Researcher ID: H-9822-2016.

11. Dr. K.B.Priya Iyer , Manisha R , Subhashree R ,Vedhavalli K "ANALYSIS OF DATA SECURITY IN CLOUD COMPUTING" International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics,2016 IEEE

12. Nentawe Y. Goshwe Arham Chopra ,"Data Encryption and Decryption Using RSA Algorithm in a Network Environment".

13. Po-Cheng Wu and Liang-Gee Chen "An Efficient Architecture for Two-Dimensional Discrete Wavelet Transform" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 11, NO. 4, APRIL 2001.