

“Securing textual information with an image in the image using a visual cryptography AES algorithm.”

Dr. Dipakkumar Dhansukhbhai Patel¹, Dr. Subhashchandra Desai²

¹Ph.D Scholar, Department of Computer Science, The Sabarmati University, Ahmedabad, India

²Department of Computer Science, The Sabarmati University, Ahmedabad, India

INTRODUCTION

Now a day's the uses of devices such as computer, mobile and many more other device for communication as well as for data storage and transmission has increases. As a result there is increase in no of user's also there is increase in no of unauthorized user's which are trying to access a data by unfair means. This arises the problem of data security. To solve this problem a data is stored or transmitted in the encrypted format. This encrypted data is unreadable to the unauthorized user. Cryptography is a science of information security which secures the data while the data is being transmitted and stored. There are two types of cryptographic mechanisms: symmetric key cryptography in which the same key is use for encryption and decryption. In case of asymmetric key cryptography two different keys are used for encryption and decryption. Symmetric key algorithm is much faster and easier to implement and required less processing power as compare to asymmetric key algorithm. The Advanced Encryption Standard (AES) was published by the National Institute of Standards and Technology (NIST) in 2001. This types of cryptography relies on two different keys for encryption and decryption. Finally, cryptographic hash function using no key instead key it is mixed the data.

BACKGROUND STUDY

In 1998, Joan Daemen and Vincent Rijmen developed the Advanced Encryption Standard (AES), a symmetric key block cipher. The AES method can be employed with any combination of data (128 bits) and key lengths of 128, 192, or 256 bits. The algorithm is referred known as AES-128, AES-192, or AES-256 depending on the key length. During the encryption-decryption process, the AES system goes through 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys to deliver the final cipher-text or retrieve the original plain text. The data length supported by AES is 128 bits, which can be split down into four basic working blocks. These blocks are organized as a 44-order matrix called the state, which can be thought of as a byte array. For both encryption and decryption, the cipher begins with the Round Key stage.

This output, on the other hand, goes through nine key phases before reaching the final step, each of which includes four transformations.

1- Subbytes, 2- Shift rows in a clockwise direction. 3- Mix columns, 4- Include a circular key.

In the last (10th) round, there is no Mix-column transformation. The full treatment was performed. Decryption uses Inverse Substitute Bytes, Inverse Shift Rows, and Inverse Mix Columns to reverse the encryption process. Each round of AES is governed by the transformations listed below. Byte transformation as a substitute AES data block is 128 bits long, therefore each data block contains 16 bytes. In sub-byte transformation, each byte (8-bit) of a data block is converted into another block using an 8-bit substitution box, also known as Rijndael X-box.

A triple-layered message security plan with an extremely high limit was proposed by the developers in S. Farrag, and its colleagues [1]. The first two layers make use of cryptography, and the third layer makes use of steganography to conceal information. In the primary layer, the mysterious message is encrypted using AES with a key length of 128 bits, which is the most secure encryption algorithm available. The yield from the first layer is transferred to the second layer, where it is scrambled and safeguarded with the help of clamorous strategic guidance. The 2D picture steganography approach is employed in the third and final layer of the plan, where the smallest component is hidden as a crisscross pattern in the RGB shade of the cover picture. This is the final layer of the plan.

According to the findings of this study A.M. Abdullah [2], the Advanced Encryption Standard (AES) algorithm is one of the most extensively utilized symmetric block cipher algorithms in use around the world. This approach is

used in hardware and software all over the world to encrypt and decrypt sensitive data, and it has a distinct structure that distinguishes it from other methods. When utilizing the AES method to encrypt data, hackers will have a difficult time decrypting the data once it has been encrypted. At this time, there is no evidence that this algorithm can be exploited. It is possible to encrypt with AES using three alternative key sizes: 128, 192, and 256 bits, with each of these ciphers requiring a 128-bit block size. This paper will provide an overview of the AES algorithm and explain some key parts of the method in detail, as well as demonstrate some prior research on it by comparing it to other algorithms such as DES, 3DES, Blowfish, and others.

To maintain secrecy, security, privacy, and confidentiality of sensitive data in this study Shafana A.R.F. [3], the authors have integrated the use of both processes, first encrypting the sensitive statistics and then concealing them in carrier media. The AES256 encryption algorithm is used to complete the encryption process, and Digital Images are used as service multimedia to deliver the service. At first, the AES256 algorithm is employed to encrypt sensitive data, which is then decrypted using a different technique. Through the use of the popular and impermeable Least Significant Bit (LSB) technique in Steganography, the encrypted messages are randomly embedded within a digital image so that they can be perceived as regularly recurring White noise. The use of both approaches has complicated the process of unintentional access since, even though the picture was suspected of containing any hidden messages, the cipher is still complicated. Therefore, this two-tier security device may be a low-cost and practical option for hiding secret messages on personal computers.

According to the findings of this study Al-Mamun A., [4], the secure and timely transmission of documents is a critical quality for every organization. Data confidentiality, authenticity, and dependability are constantly improving thanks to the use of strong encryption systems and algorithms. The Advanced Encryption Standard (AES), which is supported by the National Institute of Standards and Technology (NIST), is currently the most secure technique for maintaining data confidentiality. In summary, this research focuses on a comprehensive review of the security of the current AES algorithm, intending to increase the level of security provided by the method. By modifying the existing AES method by XORing an additional byte with the s-box value, we were able to significantly improve the Time Security and Strict Avalanche Criterion, as well as the overall security. The add insertion steganography method is the name given to the steganography method that was used in this paper G.C. Prasetyadi [5]. To avoid the annoyance of the message format, which is present in many common steganography methods, this steganography approach was chosen for its simplicity. To scramble the meaning of the concealed message, the AES-256 (Rijndael algorithm) encryption algorithm is used in conjunction with a secret passcode. Perception and validation of the original message are performed on a precise block of bytes to retrieve the message while maintaining its integrity.. As a result, the program, which is an implementation of the suggested algorithm, has been certified to be functional, but only for private use at this time due to the need for additional enhancements.

As a result, the attacker will either halt the transmission or conduct more thorough tests on the statistics from the sender to the receiver, which is the Cryptography problem addressed in this work M.E. Saleh [6]: the ciphertext appears useless. Using steganography has the disadvantage of making the message public as soon as the presence of secret data is printed or even inferred. Following the findings of this paper's research, a combined approach to information security has been developed, which combines Cryptography and Steganography techniques to improve information security. Beginning with an encrypted version of the Advanced Encryption Standard (AES) method, the secret message was transmitted over the network. Second, the approach was used to keep the encrypted message from being discovered. As a result, for the hybrid approach that has been proposed, two levels of safety have been established. According to Amal Joshy, and Fasila K.A. [7], we describe a technique for converting text into a picture using the RGB substitution algorithm, as well as a software application that encrypts the resulting photograph using the AES encryption algorithm. In this method, the secret key and the ciphertext are both sent in a single transmission, making for a very tidy package. To transform textual material into a photograph, the encryption and decryption system makes use of a mixed database on both the transmitter and receiver sides of the transaction. On the top of this encrypted image, one more pixel is added, and this pixel contains the cost of the combinational number that was previously used to convert the text into an image. The key that was previously used with the AES technique is now the same as the RGB resultant value, which is a significant improvement. Once this has been accomplished, both the resulting value as well as the snapshot that has been generated will be sent to the destination host. When it comes to function decryption, the receiver performs the decryption in reverse order.

It is proposed in Ghoradkar Sneha, and Shinde Aparna [8] that an Image Encryption and Decryption using AES (Advanced Encryption Standard) computation be used for image encryption. Because of the increasing use of photographs in a variety of industries, it is essential to safeguard classified photographic data from unauthorized access. When dealing with a square size of 128 pieces and a critical size of 256 pieces, an iterative approach is used in the design. When dealing with a critical size of 256 pieces, the number of rounds required is fourteen. The unpredictability of cryptography calculations, when used as a mystery key, increases the security of the system. According to this study, the picture is a contribution to AES Encryption to acquire the encoded picture, and the scrambled picture is a contribution to AES Decryption to obtain the initial picture.

According to Arun M., and Nivek T.N. [9], encryption and decryption are the most important procedures in any community security application, with the former being performed at the sender side and the latter being performed at the receiver side of the communication channel. Many encryption systems necessitate the use of a secret key, without which it is frequently impossible to recover the original statistics from the encrypted data in question. In this study, we propose a system that uses Modulo 256 logic to convert textual content into a pixel-based picture and then uses the AES algorithm to encrypt the received pixel-based picture after it has been decrypted. Because the key is sent along with the encrypted image, this technique is effective in resolving the AES key change problem.

The authors of this paper Jawad Ahmad, and Fwad Ahmad [10], thoroughly investigated the algorithms and provided a clear comparison between two encryption procedures, namely, Compression Friendly Encryption Scheme (CFES) and Advanced Encryption Standard (AES). The authors investigated and estimated the AES algorithm, as well as the CFES algorithm, for use in digital images, as well as their ability to protect against brute force and other attacks. The authors discovered that the weaknesses of this technique were associated with low entropy and flat association. As a result, it has been discovered that the algorithm with fewer correlation values provides greater security.

PROPOSED METHODOLOGY

AES is called AES-128, AES-192 and AES-256. This classification depends on the different key size used for cryptographic process. Those different key sizes are used to increase the security level. As, the key size increases the security level increases. Hence, key size is directly proportional to the security level. The input for AES process is a single block of 128 bits. The processing is carried out in several number of rounds where it depends on the key length: 16 byte key consists of 10 rounds, 24 byte key consists of 12 rounds, and 32 byte key consists of 14 rounds.

The first round of encryption process consists of four distinct transformation functions:

- Substitution Bytes
- ShiftRows
- MixColumns
- AddRoundKey

The final round consists of only three transformation ignoring MixColumns. The Decryption method is the reverse of encryption and it consists of four transformations.

- Inverse Substitution Bytes
- Inverse ShiftRows
- Inverse MixColumns
- AddRoundKey

AES – Encryption process

Substitution bytes: The 16 byte plain-text substitutes the corresponding value from substitution table S-box . It is a non-linear method which performs in the following way:

ShiftRows: In shiftrows transformation, the bytes in last 3 rows will be shifted cyclically over number of bytes present.

- The first row will remain same.
- The second row will get shifted to the left by one position.
- The third row will get shifted to the left by two positions.
- The fourth row will be shifted to the left by three positions.

MixColumns: MixColumns transformation performs by transforming each column of four bytes. It takes input as one column which is of 4 bytes and output as completely different 4 bytes by transforming the original column. The resultant matrix is same as the size of plain-text. MixColumn transformation will not be carried in the last round.

Add Round Key: The 16 bytes which is produced from MixColumns is equal to 128 bits which is XORed with the round key of 128 bits. The above process has been repeated until final round to produce the corresponding cipher text.

AES – DECRYPTION PROCESS

Inverse Substitution Bytes: Inverse Substitution Bytes is the inverse of the substitution byte transformation. This is performed through inverse S-box . This is obtained by applying inverse of substitution bytes and by computing multiplicative inverse of Galois Field - GF (2⁸).

Inverse ShiftRows: Inverse ShiftRows is the inverse of ShiftRows transformation. It carries out circular shifts in reverse direction for each last 3 rows and for the 2nd row, it performs one-byte circular shift to the right and it continues the process till (n-3)rd row.

Inverse MixColumns: Inverse MixColumns is the inverse of Mixcolumns transformation. It carries out operations on a matrix by column-wise. Resultant columns are in the form of polynomials.

AES Algorithm

3.3.1 Encryption Algorithm for text using steganography with cover image for stego image and using visual cryptography with the secret image.

For Encryption:

Step 1: Taking message (plain text) input by user.

Step 2: Generating random key in range.

Step 3: Storing random key in database.

Step 4: Converting plain text to cipher text by applying AES.

Step 5: AES system there are 10 rounds for 128-bit keys, 12 round for 192-bit keys, and 14 round for 256-bits in order to deliver final ciphertext or to retrieve the original plain-text. AES allows a 128 bit data length that can be divided into four basic operational blocks. These blocks are treated as array of bytes and organized as a matrix of the order of 4×4 that is called the state. For both encryption and decryption, the cipher begins with adding Round Key stage. Step 6: However, before reaching the final round, this output goes through nine main rounds, during each of those rounds four transformations are performed; 1- Subbytes, 2- Shift rows, 3- Mix columns 4 - Add roundkey. In the final (10th) round, there is no Mix column transformation.

Step 7: The AES permutation process has four stages of substitute bytes, shift rows, mix columns and add round key.

1) Substitution bytes – In this step, each byte (ai,j) of matrix is replaced with a sub byte (si,j), that is Rijndael S-Box. At the decryption end, the sub bytes are inverted to reach the original state.

2) Shift Rows - The shift rows operation, shift each rows with a certain constraint. That is first row of matrix is left same, the second, third and forth rows are shifted to one place left.

3) Mix Columns – In this step, the each column is multiplied with a fixed polynomial and the new value of the columns is placed.

4) Add Round Key – This sub key is derived from the main key and the sub key is added into this step by applying XOR to the matrix.

Step 8: Read cover image to hide cipher text.

Step 9: Hiding cipher text into cover image which gives us stego image.

(I) Generating Random Number between 0-2 for channel indicator. (0-Red, 1-Green, 2-Blue)

(II) Use MSB (3) of selected channel is used to hide cipher text according to table no. 2.

(III) Save image as stego_image.

Step 10: Hiding Stego Image in VC Shares

(I) Read Secret Image(SI).

(II) Extract RGB components from each pixel of SI.component which ranges from 0 – 255.

(III) According to the value of pixels in each channel (red,green and blue),each pixel is replaced with a 2×2 block(B1 and B2).

(IV) The fourth pixel of B1 and B2 is replaced with MSB(4) and LSB(4) of stego image.

(V) Create 2 shares for each color channel.(share1, share2, share3, share4, share5 and share6).

(IV) Shares 1, 3 and 5 are merged to form VC share1 and similarly Share2, Share4 and Share6 are merged to form VC share2.

Step 11: Save shares.(share1.png and share2.png)

Now Algorithm 1 – Embedding algorithm to hide the image and text encryption using steganography and visual cryptography

Decryption Algorithm for text using steganography with cover image for stego image and using visual cryptography with the secret image.

For Decryption:

Step 1: Select Both Shares (VC share1 and VC share2) which gives you secret and stego image by process onwards Step 12.

Step 2: Overlap VC share1 and VC share2 to get secret image.

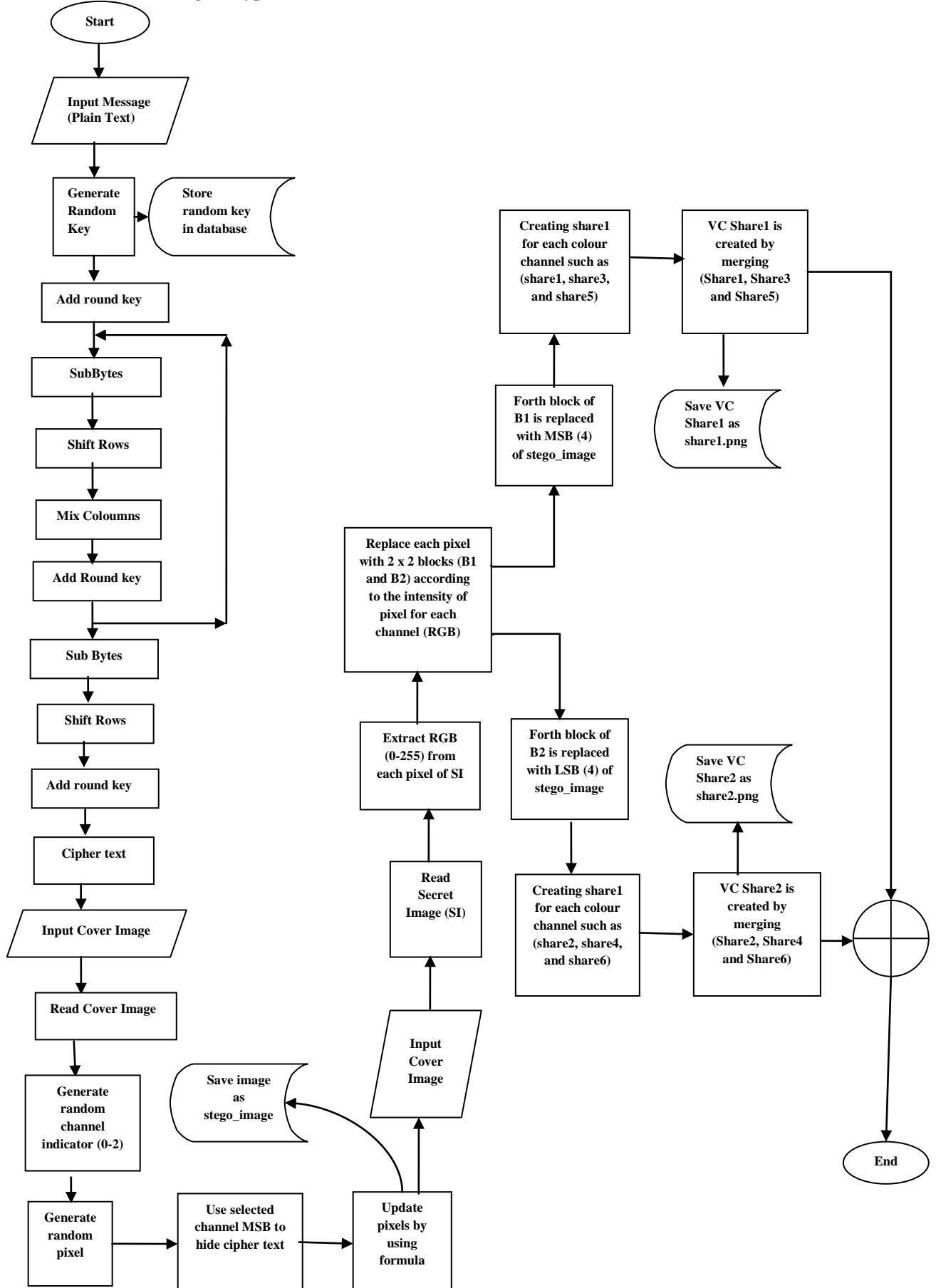
Step 3: Trace, extract and combine the values of fourth pixel of every 2×2 block of both shares to get stego image.

Step 4: From the recovered stego image the hidden cipher text is extracted by extraction process.

Step 5: The plain text is extracted from the cipher text by decryption.

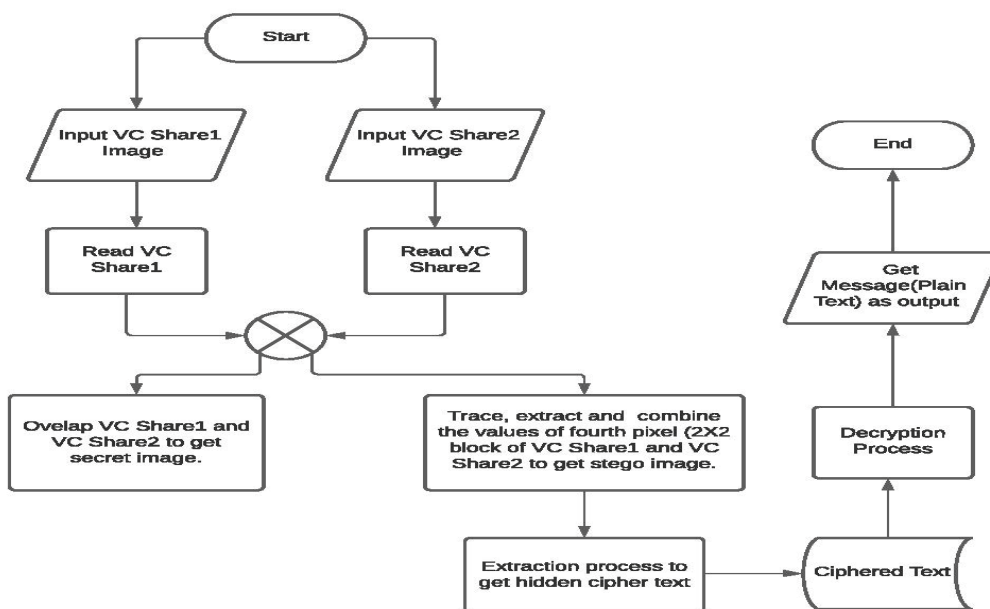
Now Algorithm 2 – Extracting algorithm to unhide image and text decryption using steganography and visual cryptography

Flowchart for Embedding encryption



[Flowchart 1: Embedding flowchart to hide image and text using encryption using steganography and visual cryptography]

Flowchart for Extraction



[Flowchart 2: Extracting flowchart to hide the image and text decryption using steganography and visual cryptography]

RESULT ANALYSIS

r_id	algo_type	image_name	r_original_size	r_hidden_data	r_psnr	r_rmse	keyid
1	AES	in2.png	31	238	86.65	0.021	13
2	AES	in2.png	31	238	88.06	0.017	14
3	AES	images (13).jpg	10	61	82.44	0.033	34
4	AES	Das_ID.jpg	16	57	83.11	0.031	35
5	AES	Lata_ID.jpg	10	55	80.29	0.043	36
6	AES	Jesica_ID.jpg	11	56	81.53	0.037	37
7	AES	Neethu_ID.jpg	7	33	79.49	0.047	38
8	AES	Sanjay_ID.jpg	10	53	78.87	0.05	39
9	AES	Philip_ID.jpg	10	61	85.45	0.024	40
10	AES	DylanRose_ID.png	113	342	89.18	0.015	42
11	AES	Eagle1.png	287	508	94.08	0.009	44
12	AES	DylanRose_ID.png	113	238	89.98	0.014	46
13	AES	img-113kb.png	113	345	85.19	0.024	47
14	AES	img-113kb.png	113	342	90.87	0.013	50
15	AES	Panda1.png	146	456	85.79	0.023	52
16	AES	in2.png	31	238	87.39	0.019	53
17	AES	img-113kb.png	113	340	87.49	0.019	54
18	AES	2cd43b_fcc6b8947ce4437da2ff2cdd600e137b_mv2.png	16	89	80.53	0.042	55
19	AES	in3.png	50	342	87.95	0.018	56
20	AES	in5.png	80	299	89.91	0.014	60

Encryption process justification with example

Input as plain text and Passkey	Krish and 12
Output comes as ciphertext	G~ed-

Here, input as the plain text with the passkey.

Plain Text = Krish

Passkey = 12

The plain text will convert the ASCII character value calculate and then it will convert it into the binary conversion. For example,

'K' character ASCII CODE IS 75 and then binary equivalent is 01010011. And so on.

Table 2 shows the conversion of Plain text to ASCII code and then Binary Code

PLAIN TEXT	ASCII CODE	BINARY EQUIVALENT
K	075	01001011
r	114	01110010
i	105	01101001
s	115	01110011
h	104	01101000

The passkey is 12, so then it will after AES operation execute in the binary number of plain text ASCII characters and then we received the ciphertext **G~ed-**.

As per the above example,

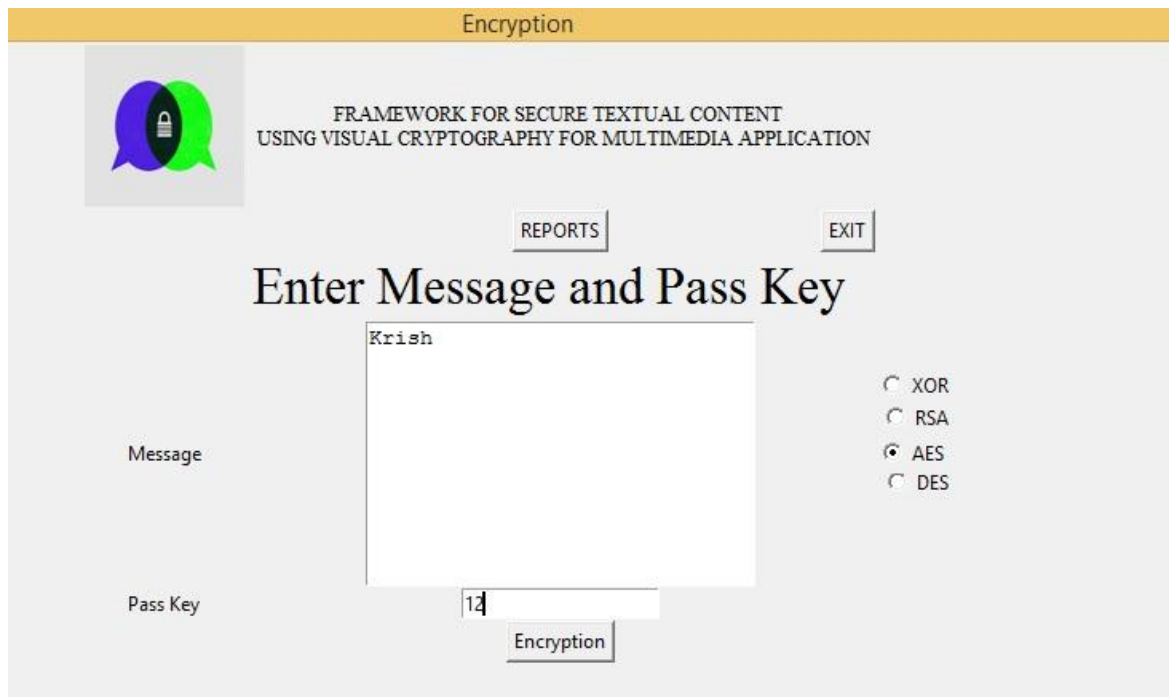
After performing AES operations on 01001011 with Passkey logic then we get the new binary number 01000111. Which is ASCII code is 071 and it is character code of G and so on we get all other character conversions after plain text to cipher text like G~ed-. The conversion table is shown below.

Table 3 shows the conversion of Binary code to ASCII code and Ciphertext

BINARY EQUIVALENT	ASCII CODE	CIPHERTEXT
01000111	071	G
01111110	126	~
01100101	101	e
01100100	100	d
00101101	045	-


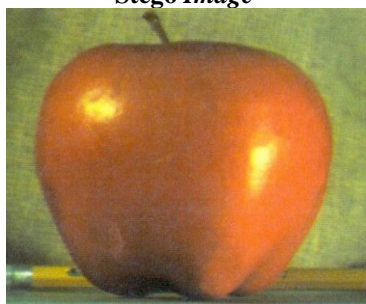
As per the above table, the binary operation will convert the plain text to cipher text using a passkey and AES operation. The Plain text is securely encrypted into the ciphertext with a passkey and AES operation. Now just see the below picture of tool encryption process execution with a passkey and AES operation.

The snapshot of my tool for the encryption process is as below.



Illustrations = 1 Snapshot for Encryption process using input plain text, passkey, and AES operation.

Use of steganography and visual cryptography process justification with example

<p align="center">Input cipher text & Cover Image</p>	<p align="center">Grid- & Cover Image</p> 
<p align="center">Output</p>	<p align="center">Stego Image</p> 

Phase 1: Stego Image Creation

After the encryption process in this phase, the secret message is embedded into random pixels of the Cover Image1 and the steps are described below.

- Step 1 = Read the secret message and convert them into bytes.
- Step 2 = Read the Cover Image1 and split it into RGB channels.
- Step 3 = Select one of the color channels using Pseudo-Random Number Generator (PRNG)
- Step 4 = Hide 4 bits of the secret message in a pixel-based on the Indicator value.

A random selection of a channel and an indicator to hide data are used in Steps 3 and 4, which are detailed in greater detail below. One of the color channels of each pixel will be randomly selected before secret data is hidden in each pixel. This will be illustrated in Table 5. Table 6 shows that after selecting the color channel at random, the three MSBs of the selected color channel are utilized as an indicator to determine whether or not to hide the data in the current pixel, as well as the number of bits to be hidden in each color channel. Similarly, if the three most significant bits of each pixel are the same, for example, 0 0 0 or 1 1 1, then no data will be buried in that particular pixel. The secret message bits will be substituted for one or two of the least significant bits of each component if this is not the case. Figure 21 depicts the results of a test using sample data, which shows how the method was tested.



Figure 4 Stego Creation - Result of phase1.



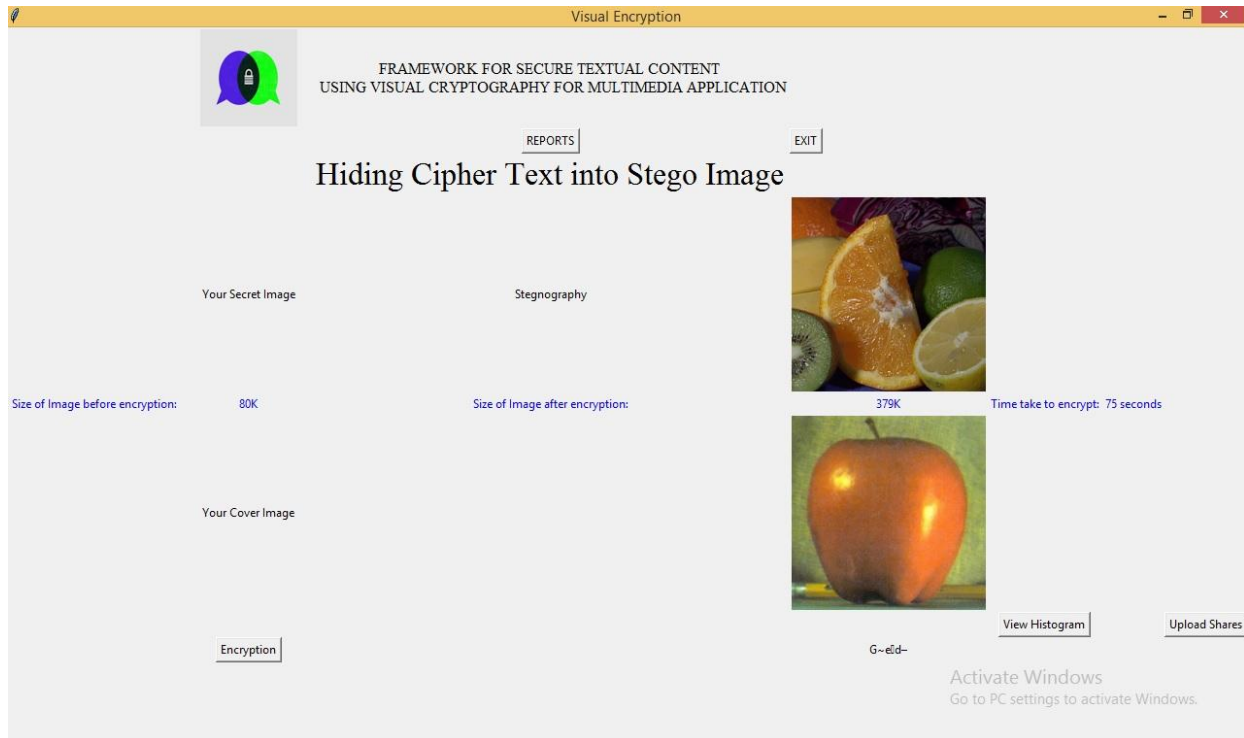
Figure 5: LSB substitution method stego image can divide in RGB color channel.

The snapshot of my tool for the select the cover image and the secret image for creating the shares are as below.







Illustrations = 2 Snapshots for select the stego image and secret image for creating the shares.

After the process of steganography and visual cryptography, the data will hide in the secret image and we will generate the histogram.



Illustrations = 3 Process of steganography and visual cryptography for hiding the secret image

Phase 2: Hiding stego image in VC shares

<p align="center">Input Stego Image and Secret image</p>	<p align="center">Stego Image</p> 	<p align="center">Secret Image</p> 
<p align="center">Output Share 1 and Share 2</p>	<p align="center">Share 1</p> 	<p align="center">Share 2</p> 

In this phase, the stego image created in phase 1 is embedded into the VC shares of Cover Image2 and the steps are described below.

- Step 1 = Read the stego image and read each pixel value
 - Step 2 = Separate the 8 bits of each color component into 2 nibbles
 - Step 3 = Read the Cover Image2 and create 2 shares for each color channel
 - Step 4 = Hide the first nibble (MSB) in share1 and second nibble (LSB) in share2
- Step 3 and Step 4 which involve the creation of VC shares and hiding of stego image simultaneously are explained below. The Cover Image2 is split up into 3 color channels (RGB) and two shares are created depending on the intensity of pixel values (whether it is greater than or less than 128) of each color channel.

It expands each pixel into two 2×2 blocks (B1 and B2) to which a color is assigned as shown in Fig. 2. This shows the blocks created for the Red channel. Similarly, blocks are created for Blue and Green channels. The fourth pixel of B1 is replaced with first nibble and B2 is replaced with the second nibble of the stego image. B1 of all pixels form share1 and B2 form share2.

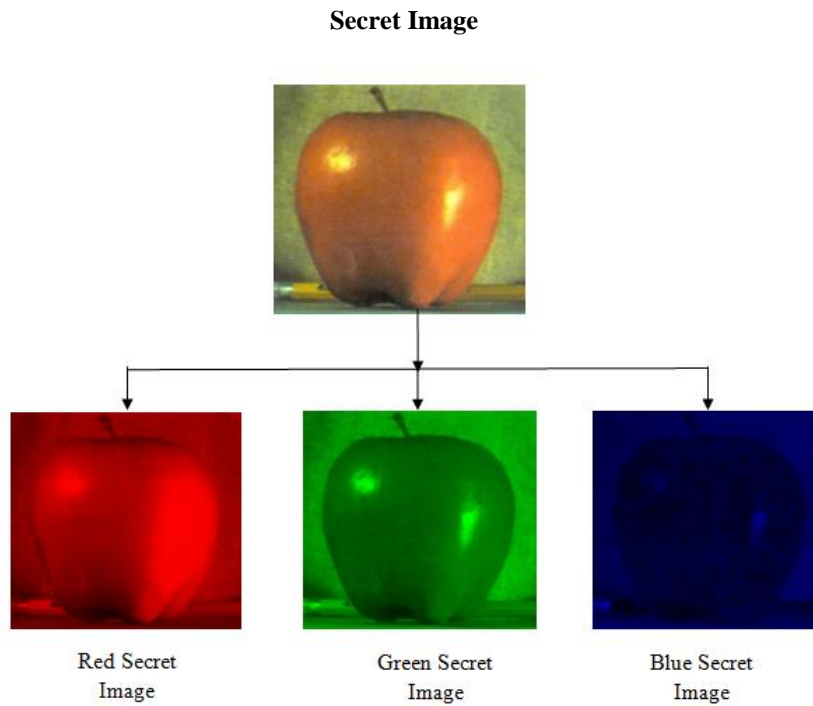
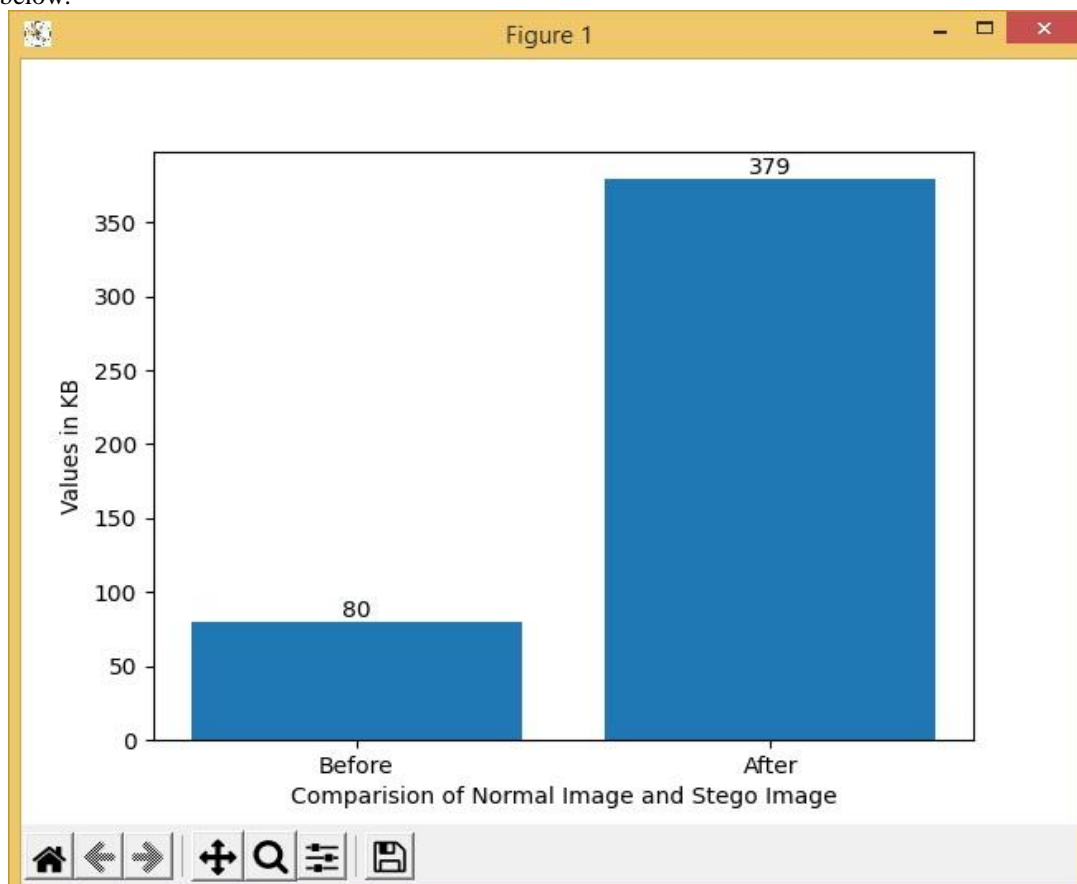


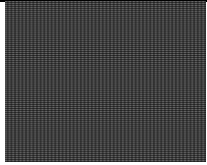

Figure 9 LSB substitution method Secret image can divide into RGB color channels.

The snapshot of my tool for the after the visual cryptography the histogram will be check for the changes of image size as below.



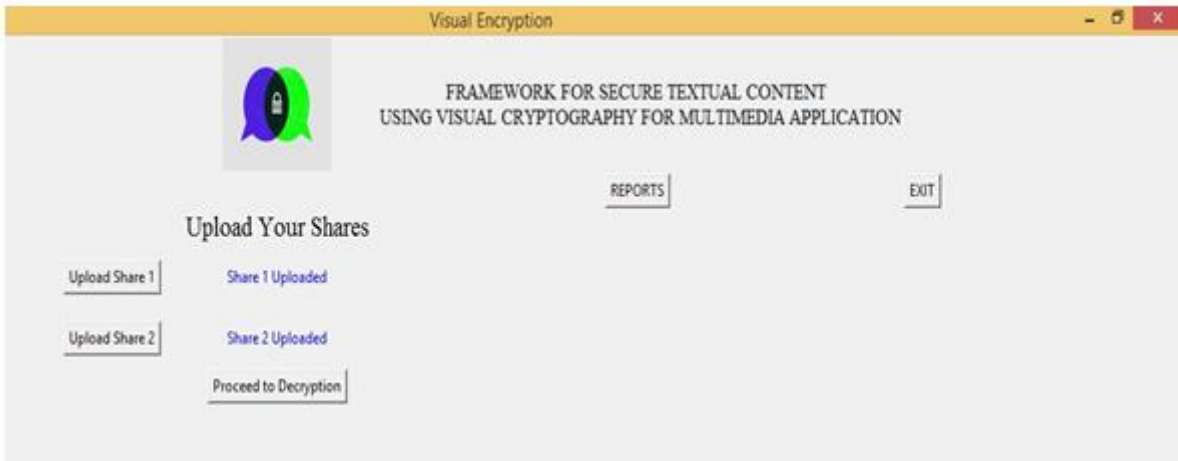
Illustrations = 4 Snapshot for histogram for stego image before and after operation of data hiding.

Decryption process justification with example

Input			G~ed-
	(1024 * 768) 5.41 KB Share 1	(1024 * 768) 149 KB Share 2	
Output	Krish		
	Original Text (Plain Text)		

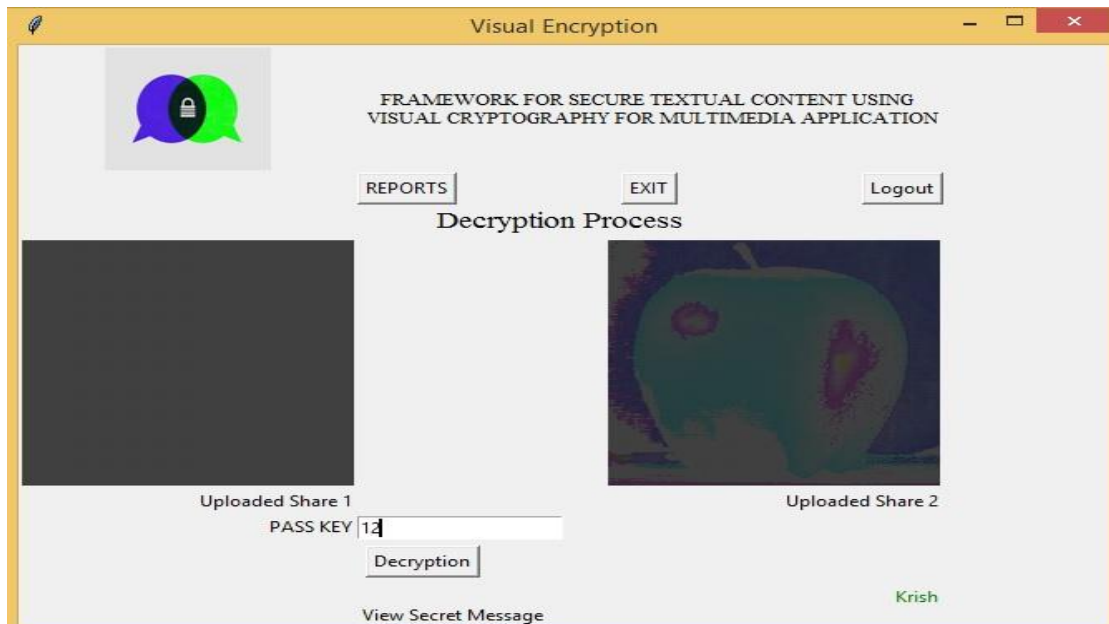
The procedure of decryption is straightforward. It is not necessary to restore the multimedia content if neither the stego picture nor the Cover Image2 is present. By overlapping the two shares, the Cover Image2 can be disclosed without the use of any mathematical processes. Using tracing, extraction, and combination of the values of the fourth pixel of every 2 x 2 blocks in each of the two shares, the stego image may be reconstructed. It is possible to extract a hidden message from the restored stego image. As a result, this multi-level stego-vc system aids in the secure transmission of communications, which is extremely difficult to crack.

The snapshot of my tool for the upload the shares for the decryption process.



Illustrations = 5 Snapshot for uploading the shares

The snapshot of my tool for the decryption process using with selecting the shares and putting the passkey for the same is as below.



Illustrations = 6 Snapshot for Decryption process using selecting the shares and putting the passkey.


Justification for result analysis

When it comes to concealing multimedia data, the suggested approach makes use of the advantages of VC and Steganography. The algorithm, which is built in the Python programming language, is tested using example data, and the results are depicted in Figure. Some of the most important aspects of this study are discussed and listed.

Imperceptibility

The second phase of the proposed solution is tested by concealing text files of varying sizes under a cover image. Calculating the RMSE and PSNR values is used to evaluate it, and the results are shown in Table 9. As a result, it is found that the PSNR value is 88.49 dB of the message, implying that there is no significant visual distortion even when hiding 38KB of the message. Table 3.PSNR and RMSE values Cover Image Hidden Data(KB) PSNR (dB) RMSE.

Table 6 Calculation of Cover image hidden data, PSNR and RMSE

Cover Image	Hidden Data (KB)	PSNR (dB)	RMSE
 (512 * 384) 80.4 KB	299	89.9	0.014

Resistance to Steganalysis

Although the changes made to the cover image as a result of data concealing are undetectable to HVS, a variety of steganalysis methods are available to discover the presence of a hidden message in the steg medium. Deduction via steganalysis can be avoided by employing the VC technique, which involves hiding the stego picture within the shares of a secret image that has been constructed. Even after suppressing the stego image, as illustrated in Figure8, the shares that are generated are always meaningless and worthless. This technique assures that hackers will not be able to deduce any information about the secret image from the shares that have been produced.

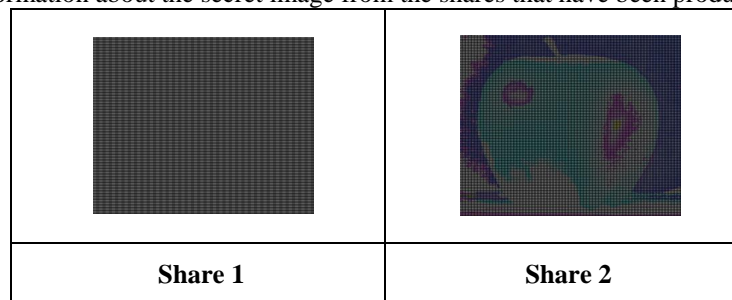


Figure 15 Shares created

Multilevel Security

- ✓ This system protects the information being communicated with four different layers of protection.
- ✓ Phase 1 involves encrypting the secret message.
- ✓ In phase 2, the secret message is hidden in an image using a dynamic and random algorithm.
- ✓ In phase 3, the stego picture will be embedded in VC shares.
- ✓ Shares of the hidden image made in step 3 are meaningless and dumb As a result, even if intruders are aware of the presence of a secret data stream, they will be unable to simply break into the system.

Multimedia security

This approach allows the user to hide various pieces of data in different formats, such as text and images, at the same time. Two secret text files, two cover images, and a secret picture are hidden in the shares of the secret image in this manner. From the received shares, the recipient can extract the hidden image, stego images, and secret messages. As a result, our technique enables the hidden transmission of multiple types of data in massive volumes.

Message Integrity

If the receiver can extract the precise message that was disguised and conveyed, the security technique is termed efficient. The Secret Message is hidden in the spatial domain of the image in this suggested approach, and no alterations are made, therefore the message obtained in the extraction phase is identical to the hidden message (Fig. 9). As a result, this strategy guarantees data integrity.

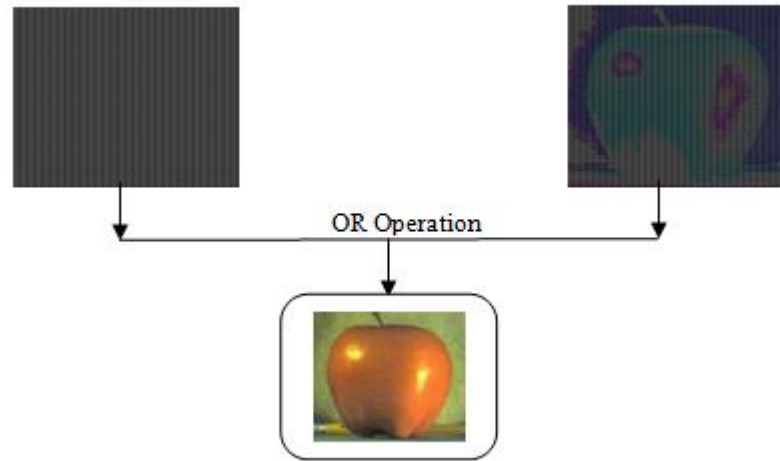


Figure 16 Result of Extraction

PSNR

The MSE represents the average of the squares of the "errors" between our actual image and our stego image. The error is the amount by which the values of the original image differ from the degraded image.

$$MSE = \frac{1}{mn} \sum_0^{m-1} \sum_0^{n-1} \|f(i,j) - g(i,j)\|^2$$

Where,

f represents the matrix data of our original image

g represents the matrix data of our stego image

m represents the numbers of rows of pixels of the images and i represents the index of that row




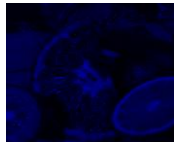
n represents the number of columns of pixels of the image and j represents the index of that column

Peak Signal to Noise Ratio (PSNR) The peak signal-to-noise ratio (PSNR) in decibels is computed between two images. This ratio is often used as a quality measurement between the original and the reconstructed image. The higher the PSNR better the quality of the reconstructed image.

$$PSNR = 20 \log_{10} \left(\frac{MAX_f}{\sqrt{MSE}} \right)$$

MAX_f is the maximum signal value that exists in the cover image. The PSNR of Cover and Stego image with different sizes of data hidden is shown in the figure below. Similarly, the PSNR of Cover Share and the Stego Share after hiding the stego image of size 512 X 384 are shown in fig.10. From the PSNR value, it is evident that the clarity of the Stego image is almost the same as the original image.

Table 7 Comparison of the Cover image with red, green & blue stego images

Cover Image	Red Stego Image	Green Stego Image	Blue Stego Image
 (512 * 480) 80.4 KB	 (512 * 480) 215 KB	 (512 * 480) 220 KB	 (512 * 480) 234 KB

Histogram

A histogram is a graphical representation of statistical information that uses rectangles to depict the frequency of data items in successive numerical intervals of equal size across time. Histograms are most commonly represented by the horizontal axis, with the independent variable drawn along the horizontal axis and the dependent variable plotted along the vertical axis.

The following histogram depicts the relationship between pixel value and the number of pixels in the image. As illustrated in Fig.11, the histograms created before and after hiding the stego pictures are shown in comparison. It indicates that by hiding the Stego picture in the VC shares, just a small number of pixel values are altered.

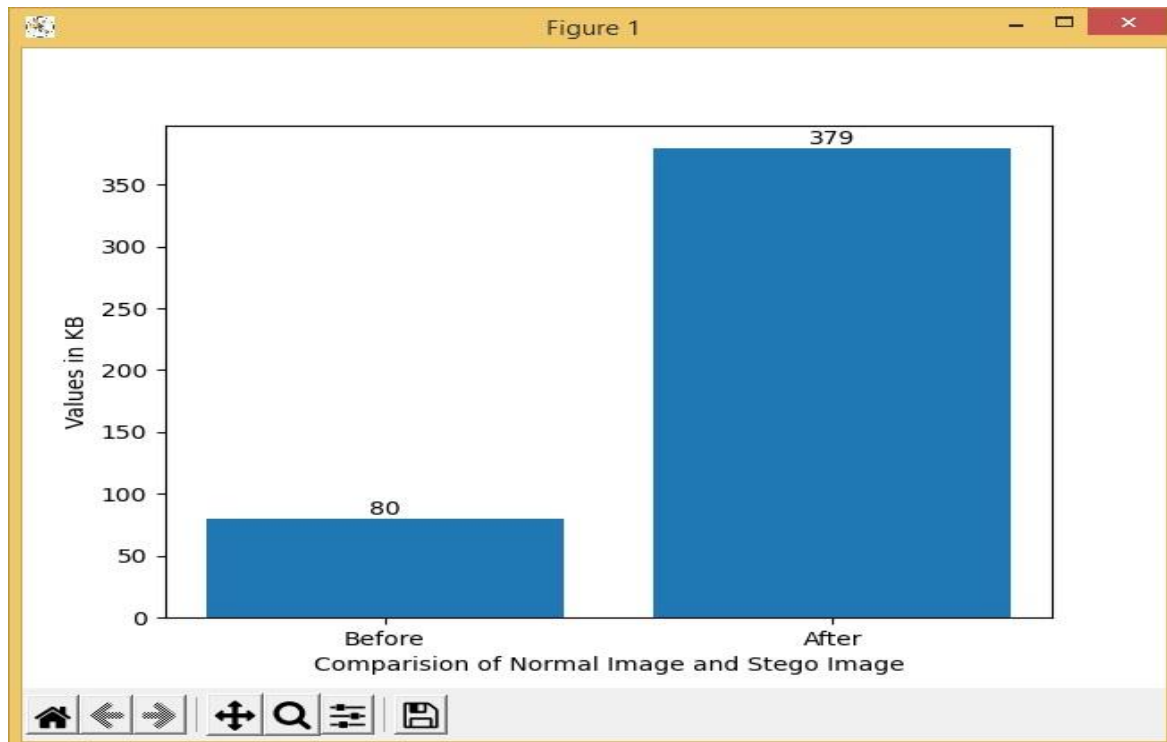


Figure 1 Before and after comparison of the normal image, and stego image

Robust and simple

This method is fairly easy because the data is hidden just by altering a small number of least significant bits, and it requires little computational power. Because we are using VC, there is no need for a complicated decryption algorithm. The algorithm's results corroborate the system's robustness, which is a good thing.

Capacity

In this system, the ability to conceal information is very high. Figure 10 shows a comparison between the size of the hidden message and the size of the cover image. The number of shares enhances the embedding capacity of the system because a greater number of stego pictures may be embedded in the shares as the number of shares increases.

SUMMARY OF CHAPTER

The proposed work makes use of AES algorithm to encrypt and decrypt the image and text. It makes use of 128 bit key for encryption. In our proposed system, encryption image doesn't remain the same. The encryption image is chosen in random. So, it is difficult for intruder to differentiate the encrypted image and the original image. So, AES algorithm is most suited for image encryption in real time applications. As a future work, we are planning for a different encryption keys in each round to perform encryption. Image Encryption and Decryption using AES algorithm is implemented to secure the image data from an unauthorized access. A Successful implementation of symmetric key AES algorithm is one of the best encryption and decryption standard available in market. With the help of PYTHON coding implementation of an AES algorithm is synthesized and simulated for Image Encryption and Decryption. The original images can also be completely reconstructed without any distortion. It has shown that the algorithms have extremely large security key space and can withstand most common attacks such as the brute force attack, cipher attacks and plaintext attacks.

REFERENCES

- [1]. S. Farrag, W. Alexan, and H. Hussein, "Triple-layer image security using a zigzag embedding pattern," in 2019 International Conference on Advanced Communication Technologies and Networking (CommNet'19), Morocco, Apr. 2019.
- [2]. A. M. Abdullah, Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data, Cyprus UK: Research Gate Departement Of Applied Mathematics & Computer Science, 2017.

- [3]. Shafana A.R.F. "TWO TIER SHIELD SYSTEM FOR HIDING SENSITIVE TEXTUAL DATA", Proceedings of 7th International Symposium, SEUSL, ISBN 978-955-627-120-1, pp. 97-103., 7th & 8th December 2017.
- [4]. Al-Mamun, A., Rahman, S., et al.: Security analysis of AES and enhancing its security by modifying S-box with an additional byte. *Int. J. Comput. Netw. Commun. (IJCNC)* 9(2) (2017).
- [5]. G. C. Prasetyadi, A. Benny Mutiara and R. Refianti, "File encryption and hiding application based on advanced encryption standard (AES) and append insertion steganography method," 2017 Second International Conference on Informatics and Computing (ICIC), Jayapura, 2017, pp. 1-5.
- [6]. M. E Saleh, A. A. Aly, and F. A. Omara, "Data Security Using Cryptography and Steganography Techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 6, pp. 390–397, 2016.
- [7]. Amal Joshy, Amitha Baby K X, Padma S, Fasila K A "Text to Image Encryption Technique using RGB Substitution and AES" IEEE International Conference on Electrical, Computer and Communication Technologies, Coimbatore, pp 19-21, February 2016.
- [8]. Ghoradkar, Sneha and Shinde, Aparna, "Review on Image Encryption and Decryption using AES Algorithm," *International Journal of Computer Applications (0975–8887)*, National Conference on Emerging Trends in Advanced Communication Technologies, (NCETACT-2015).
- [9]. Arun, M., Azarudeen S. Mohamed and Nivek, T.N. "AES based Text to Pixel Encryption using Color Code Conversion by Modulo Arithmetic". *International Journal of Recent Research in Science, Engineering, and Technology*. Vol. 1, No. 3 pp 37-42, June 2015.
- [10]. Jawad Ahmad and Fawad Ahmed —Efficiency Analysis and Security Evaluation of Image Encryption Schemes| *International Journal of Video & Image Processing and Network Security IJVIPNS-IJENS* Vol: 12 No: 04, 2012