



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

<sup>1</sup>Dr.Dipakkumar Dhansukhbhai Patel, <sup>2</sup>Dr. Subhashchandra Desai

<sup>1</sup>Ph.D Scholar, Department of Computer Science, The Sabarmati University, Ahmedabad, India.

<sup>2</sup>Department of Computer Science, The Sabarmati University, Ahmedabad, India

## 1. INTRODUCTION

Security of data text file on a computer can be done by utilizing encryption and decryption techniques. One technique is encryption and decryption of data encryption system text file with cryptography. Cryptography is the science or art to randomize and secure the message to avoid the manipulation of data by performing encryption and decryption on the data of the text file. Cryptographic algorithm consists of several types of one example is the algorithm Rivest Shamir Adleman (RSA). The purpose of this research is to design and implement RSA cryptography algorithm on text file data security. The results of this design is a desktop based software. This research is expected to provide benefits to the users, individually as well for other organizations. One of the principal challenges of resource sharing on data communication network is its security. This paper presents a design of data encryption and decryption in a network environment using RSA algorithm with a specific message block size. The algorithm allows a message sender to generate a public keys to encrypt the message and the receiver is sent a generated private key using a secured database. An incorrect private key will still decrypt the encrypted message but to a form different from the original message.

**Keywords:** Cryptography, Visual Cryptography, Steganography, Encryption, Decryption, Data Hiding, Stego Images, Security, RSA

## 2. BACKGROUND STUDY

RSA is founded in 1977 is a public key cryptosystem. RSA is an asymmetric cryptographic algorithm named after its founders Rivest, Shamir & Adelman. It is one of the best-known public key cryptosystems for key exchange or digital signatures or encryption of blocks of data. RSA uses a variable size encryption block and a variable size key. It is an asymmetric (public key) cryptosystem based on number theory, which is a block cipher system. It uses two prime numbers to generate the public and private keys size is 1024 to 4096 bits. These two different keys are used for encryption and decryption purpose. Sender encrypts the message using Receiver public key and when the message gets transmit to receiver, then receiver can decrypt it by using his

own private key. RSA operations can be decomposed in three broad steps; key generation, encryption and decryption. RSA have many flaws in its design therefore not preferred for the commercial use. When the small values of  $p$  &  $q$  are selected for the designing of key then the encryption process becomes too weak and one can be able to decrypt the data by using random probability theory and side channel attacks. The sequence of events followed by RSA algorithm for the encryption of multiple blocks. Decrypt blocks of data consisting of 64 bits by using a 64-bit key.

Elisa Bertino [1] explained the challenges, concept and approaches of database security. Many concepts regarding database security were provided and most significant techniques were discussed which were based on accessing control system. He defines the key access control models which were mandatory access control models and the role-based access control (RBAC) model. He also described security for advanced data management systems. The major drawback was that a new device was to be issued, when an individual user required to change the subscription.

Hua Li [2] et al. explained new compact dual-core architecture used in AES. The practice of using a new compact architecture started that consisted of two independent cores that practice encryption and decryption simultaneously. In order to provide round keys for encryption and decryption, proposed key generation unit with 32-bit data path was explored. The concept used to implement shift rows was the important design which helps to increase the encryption time. The major limitation was that in comparison to the other designs, this design also requires fewer more hardware resources.

H. C. Williams [3] modified the RSA public-key encryption algorithm. He suggested that if the encryption procedure was broken into a certain number of operations than remainder used as modulus could be factored after few more operations. This technique was in similar appearance to RSA so as produce digital signatures. The main limitation of this scheme was that very large prime numbers were used and generated mathematical errors were observed.

Adam J. Elbirt [4] et al. explained the AES block cipher algorithm using FPGA based kit. They proposed that for hardware implementations of encryption algorithms, reprogrammable devices were the best choice. The disadvantage was that when the implementation size was increased then the number of rounds unrolled also enhanced and this increase was partially offset by the packing of the round keys within the round structure.

Hung-Yu Chien [5] highlighted an efficient time bound hierarchical key assignment scheme. They proposed a tamper resistant device that has a new time bound key assignment scheme. It significantly improves the computational performance and reduces the implementation cost as well.

Taher Elgamal [6] proposed a signature scheme based on discrete logarithms and implemented Diffie-Hellman key distribution scheme that achieves a public key cryptosystem. The security of both systems depends on the difficulty of computing discrete logarithms over finite fields.

Martin E. Hellman [7] extended the Shannon theory approach to cryptography. He discussed about Shannon's random cipher model which was conservative than in such case when a randomly chosen cipher was considered, the security falls significantly. The concept of matching a cipher to a language and the trade-off between local and global uncertainty were also developed. The limitation of this approach is that it is not directly applicable to designing practical cryptographic systems.

Jason H. Li [8] et al. worked on scalable key management and clustering scheme for secure group communication in Adhoc and WSN. They describe scalable key management and clustering to achieve more secured system. The scalability problem was solved by partitioning communicating devices into subgroups with a leader in each subgroup. The Distributed Efficient Clustering Approach (DECA) provided robust clustering to form subgroups. Analytical and simulation results pinpoint the fact that DECA was energy efficient and resilient against node mobility. This scheme was not suitable for large cluster size.

Hung-Min Sun [9] et al. proposed dual RSA algorithm and also did the acute analysis of the security of the algorithm. Dual RSA was a variant of RSA which is helpful in some specific situations that require two instances of RSA with the advantage of reducing the storage requirements for the keys. The main drawback of using dual RSA was that the computational complexity of the key generation algorithms was also optimised.

Mao-Yin Wang [10] et al. configured single and multi-core AES architectures for flexible security. According to them the major building blocks for the architecture of AES was a group of AES processors. Each AES processor provides a block cipher scheme with a novel key expansion design approach for the original AES algorithm. In this multi core architecture the memory controller of each AES processor was designed for the maximum overlapping between data transfer and encryption and thus reducing interrupt handling load of the host processor.

### 3. PROPOSED METHODOLOGY

The main aim of the proposed algorithm is a secret message exchanges it between the sender and the receiver by using the RSA cryptosystem and they do not need to know the public key for each other. In the processes of encryptions and decryptions of the RSA cryptosystems, the encryption process done twice in a row by the sender and receiver of the message using the RSA algorithm, as well as the decryption process performed twice in succession by the receiver and sender of the message. The attributes used are text messages. It is processed through the encryptions and decryptions processes. There are three stages in the process of encryption and decryption of the message. In this combination process using a RSA algorithm to perform encryption and decryption of messages to be sent, while for the message delivery process using three pass algorithm protocol.

The RSA algorithm was first used to implement the concept of public key cryptography and has been widely used because it is easier to understand and implement than other public key algorithms. However, the RSA algorithm is computationally intensive with very large integer numbers. Strong primes are required for RSA security. Thus, additional cost is indispensable for generating strong primes in RSA.

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

For encrypting any message, the algorithm converts the given message into an integer number by using a suitable padding scheme. Then following formula is used to generate encrypted message C:  $C = M^E \text{ mod } (N)$

#### 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 RSA.

Step 5: Choose two prime numbers, p and q. From these numbers you can calculate the modulus,

$$n = pq$$

Step 6: Select a third number, e, that is relatively prime to (i.e. it does not divide evenly into) the product  $(p-1)(q-1)$ , the number e is the public exponent.

Step 7: Calculate an integer d from the quotient  $(ed-1)/(p-1)(q-1)$ . The number d is the private exponent.

Step 8: The public key is the number pair  $(n,e)$ . Although these values are publicly known, it is computationally infeasible to determine d from n and e if p and q are large enough.

Step 9: To encrypt a message, M, with the public key, creates the cipher-text, C, using the equation:

$$C = M^e \text{ Mod } n, \text{ C is cipher text.}$$

Step 10: Read cover image to hide cipher text.

Step 11: 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 12: 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 2X2 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 13: Save shares.(share1.png and share2.png)

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

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

Following formula is used to decrypt the encrypted message:

$$M = C \wedge D \text{ mod } (N)$$

#### **For Decryption:**

Step 1: Select Both Shares (VC share1 and VC share2) which gives you secret and stego image by process onwards Step 13.

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

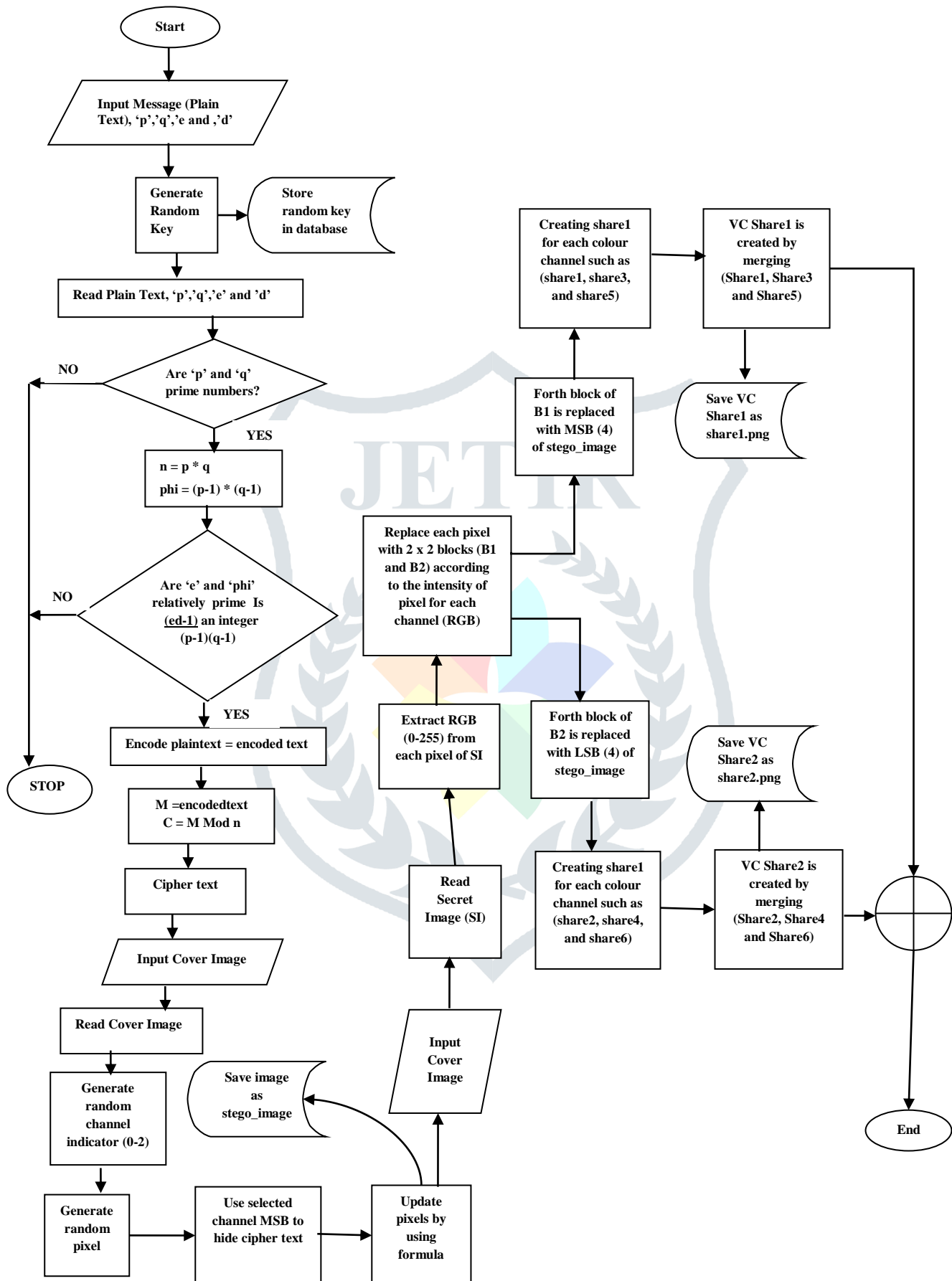
Step 3: Trace, extract and combine the values of fourth pixel of every 2X2 block of both shares to get stego image.

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.

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

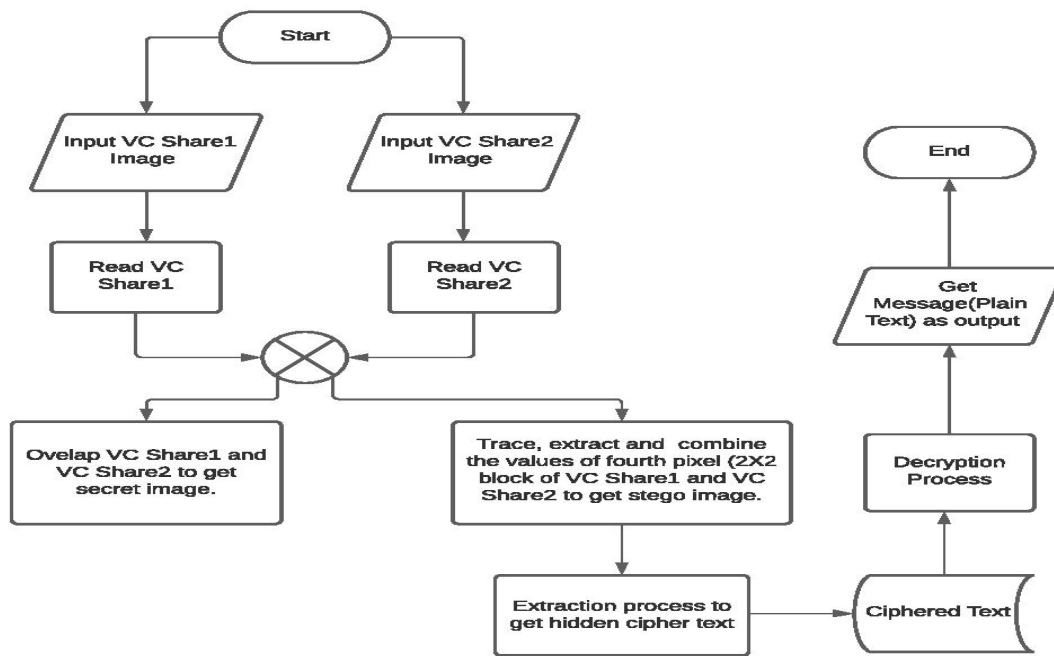
### 3.3 Flowchart for Embedding encryption



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



### 3.4 Flowchart for Extraction



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

## 4 RESULT ANALYSIS

r_id	algo_type	image_name	r_original_size	r_hidden_data	r_psnr	r_rms_e	key_id
1	RSA	in2.png	31	238	87.51	0.019	89
2	RSA	Das_ID.jpg	16	57	80.42	0.042	90
3	RSA	Pushkal_ID.jpg	11	68	81.6	0.037	91
4	RSA	Sanjay_ID.jpg	10	53	82.28	0.034	92
5	RSA	Rukmini_ID.jpg	12	69	81.65	0.037	93
6	RSA	Sanjay_ID.jpg	10	53	79.95	0.044	94
7	RSA	OmSaiRam.jpg	133	517	92.26	0.011	95
8	RSA	in4.png	169	1241	95.16	0.008	96
9	RSA	Sunflower1.jpg	84	135	88.34	0.017	97
10	RSA	images.jpg	10	84	81.07	0.039	98
11	RSA	Horse3.jpg	73	219	92.95	0.01	99
12	RSA	Shankh1.jpg	32	94	84.15	0.027	100
13	RSA	in2.png	31	238	87.09	0.02	101
14	RSA	in3.png	50	342	87.36	0.019	102
15	RSA	Neethu_ID.png	38	158	80.97	0.04	103
16	RSA	in5.png	80	299	87.17	0.019	104
17	RSA	in90.png	43	280	86.59	0.021	105
18	RSA	in4.png	169	1241	94.11	0.009	106
19	RSA	in5.png	80	299	89.33	0.015	107
20	RSA	in2.png	31	238	86.92	0.02	108

## Encryption process justification with example

<b>Input as plain text and Passkey</b>	<b>Omsaidip and 11</b>
<b>Output comes as ciphertext</b>	<b>Egykencz</b>

Here, input as the plain text with the passkey.

Plain Text = **Omsaidip**

Passkey = 11

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

For example,

'O' character ASCII CODE IS 079 and then binary equivalent is 01001111. And so on.

Table 7 shows the conversion of Plain text to ASCII code and then Binary Code

PLAIN TEXT	ASCII CODE	BINARY EQUIVALENT
O	079	01001111
m	109	01101101
s	115	01110011
a	097	01100001
i	105	01101001
d	100	01100100
i	105	01101001
p	112	01110000

The passkey is 11, so then it will after RSA operation execute in the binary number of plain text ASCII characters and then we received the ciphertext **Egykencz**.



As per the above example,

After performing RSA operations on 01001111 with Passkey logic then we get the new binary number 01000101.

Which is ASCII code is 069 and it is character code of E and so on we get all other character conversions after plain text to cipher text like **Egykencz**. The conversion table is shown below.

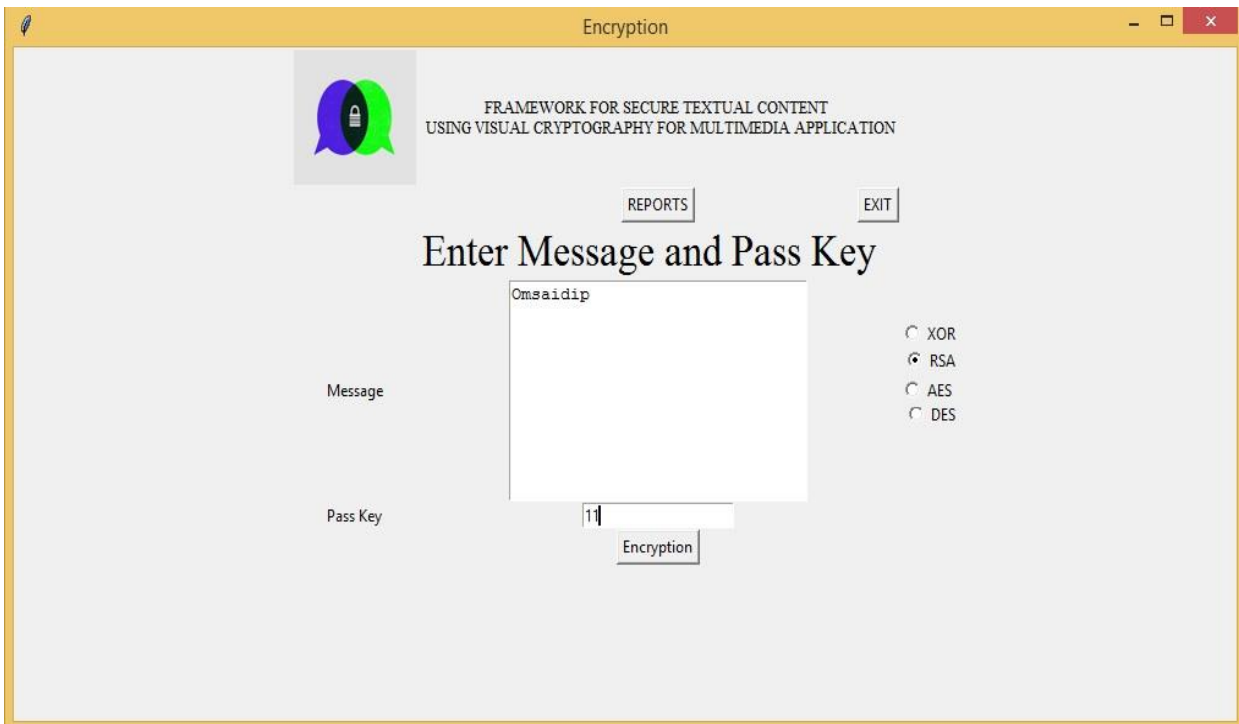
Table 8 shows the conversion of Binary code to ASCII code and Ciphertext

BINARY EQUIVALENT	ASCII CODE	CIPHERTEXT
01000101	069	E
01100111	103	g
01111001	121	y
01101011	107	k
01100011	099	c
01101110	110	n
01100011	099	c
01111010	122	z

As per the above table, the binary operation will convert the plain text to cipher text using a passkey and RSA operation.



The Plain text is securely encrypted into the ciphertext with a passkey and RSA operation. Now just see the below picture of tool encryption process execution with a passkey and RSA 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 XOR operation.

**4.2.3.2 Use of steganography and visual cryptography process justification with example**

<p><b>Input cipher text &amp; Cover Image</b></p>	<p><b>Egykencz &amp; Cover Image</b></p> 
<p><b>Output</b></p>	<p><b>Stego Image</b></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 21 Stego Creation - Result of phase1.

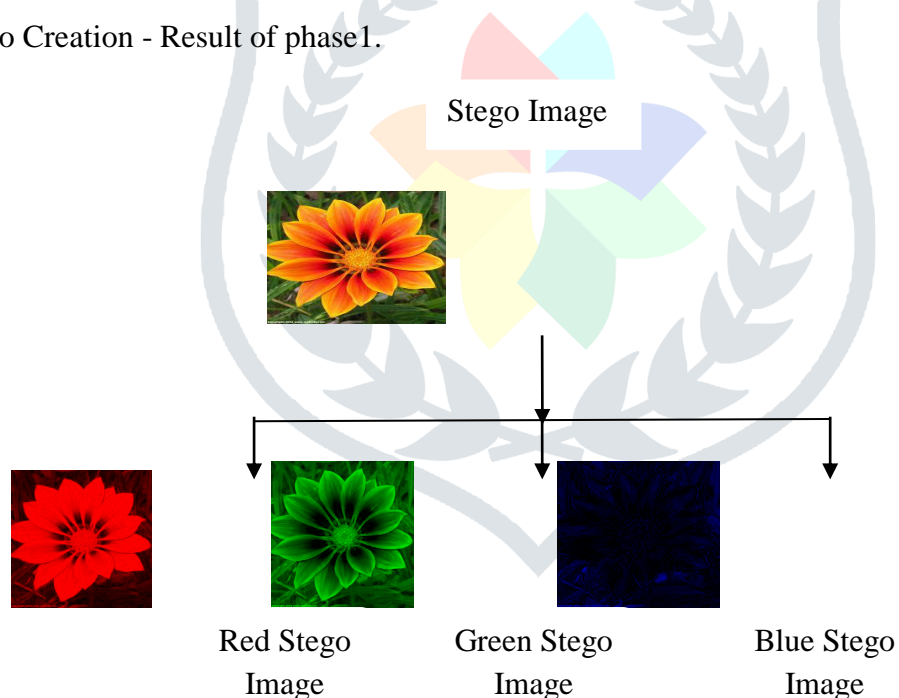
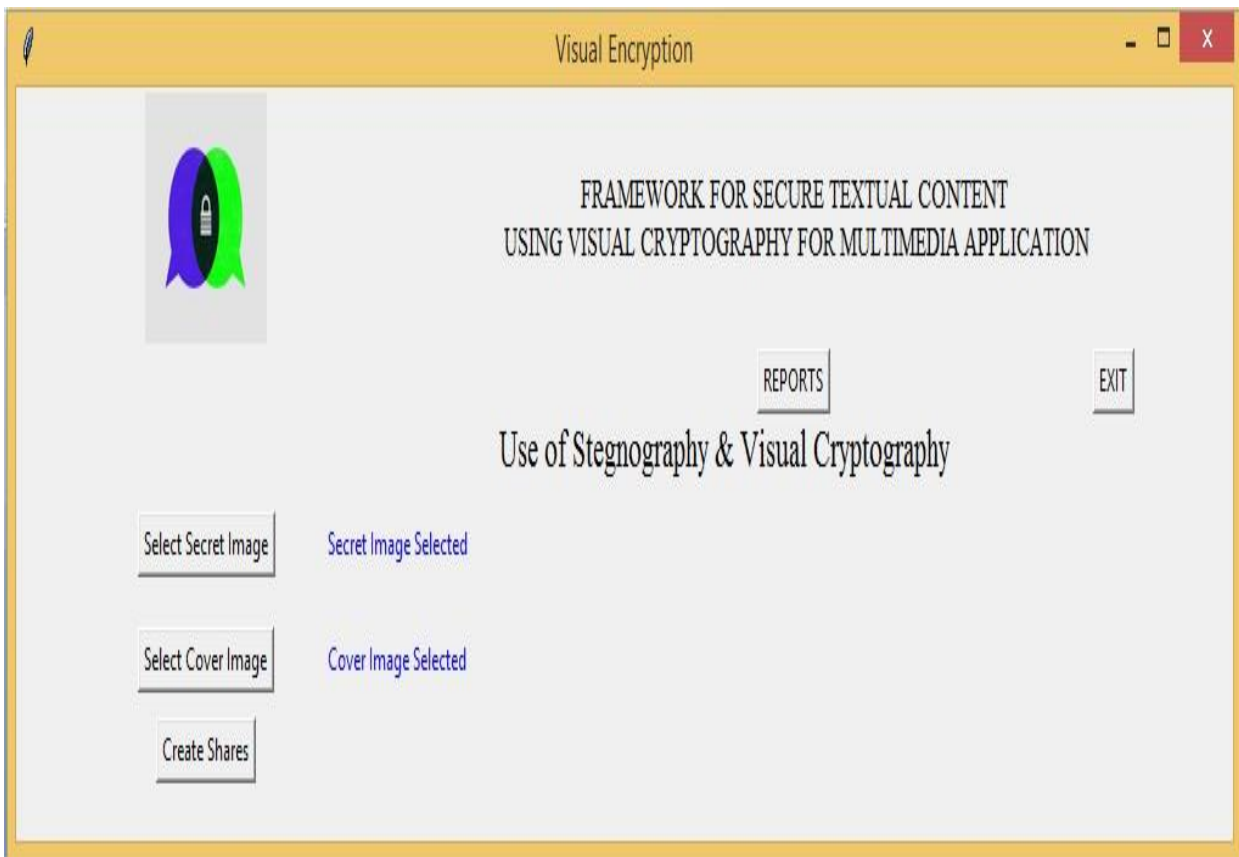


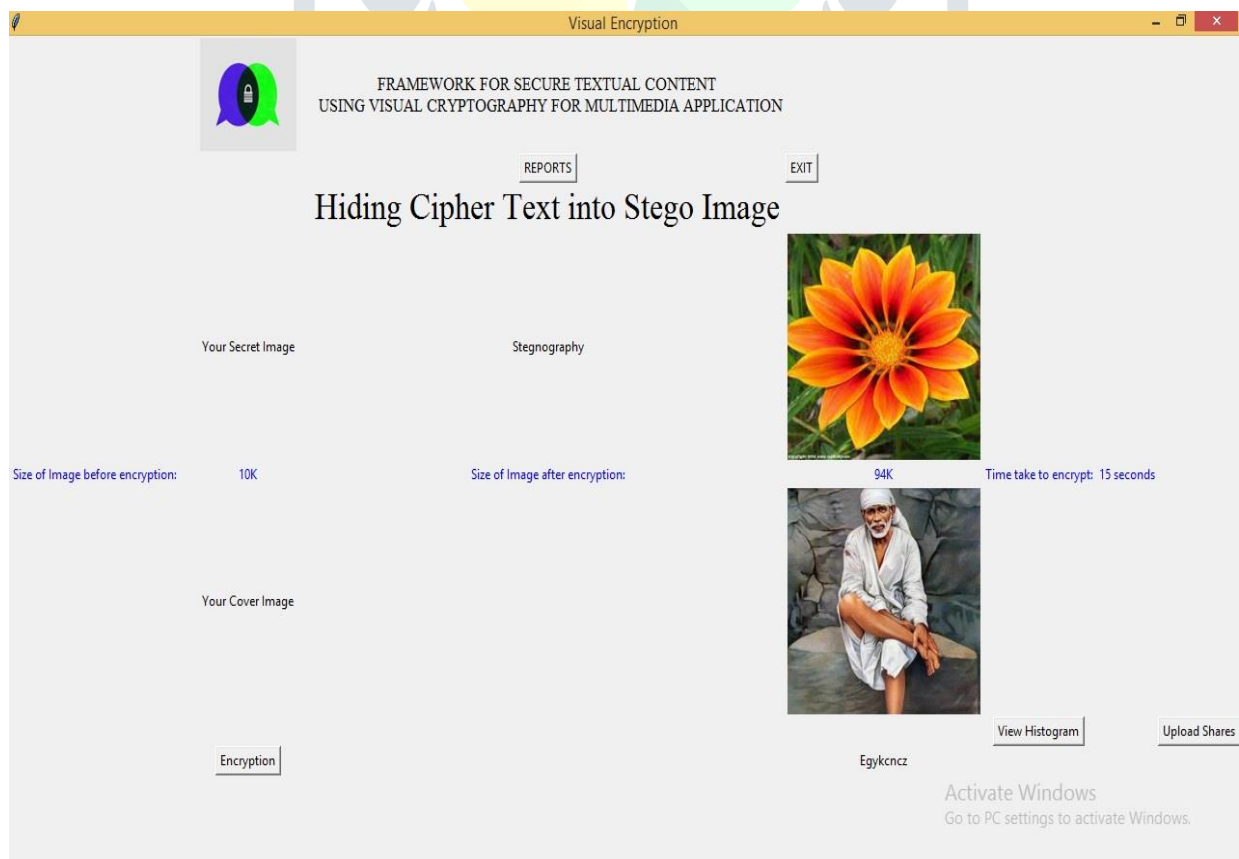
Figure 22 LSB substitution method stego image can divide in RGB color channel.

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




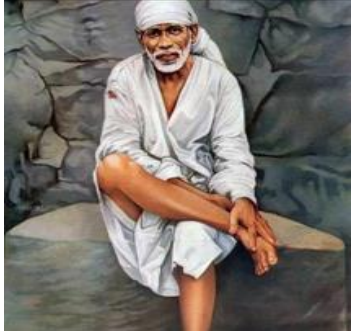
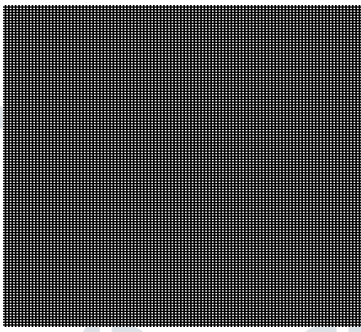
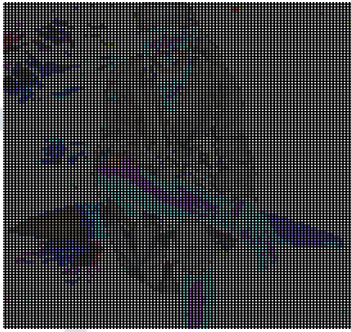
Illustrations = 2 Snapshots for select the stego image and secret image for creating the shares.

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**

<b>Input Stego Image and Secret image</b>	<p style="text-align: center;"><b>Stego Image</b></p> 	<p style="text-align: center;"><b>Secret Image</b></p> 
<b>Output Share 1 and Share 2</b>	<p style="text-align: center;"><b>Share 1</b></p> 	<p style="text-align: center;"><b>Share 2</b></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  $2 \times 2$  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.

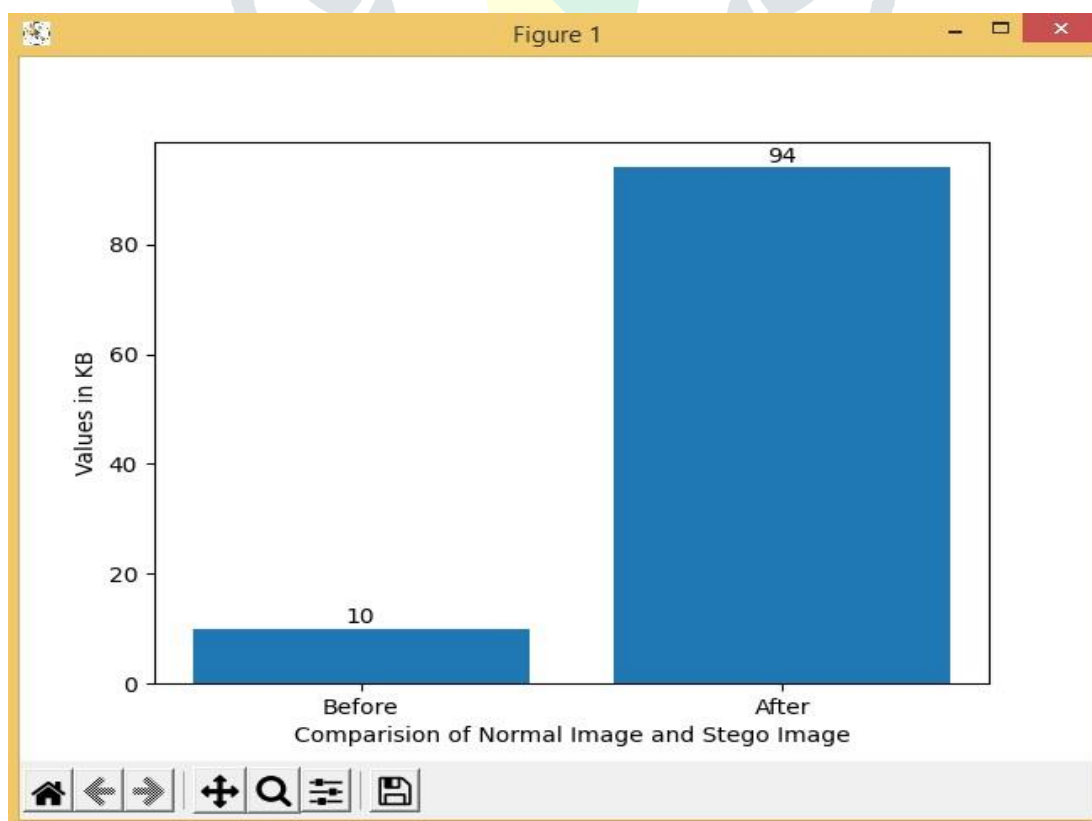


Secret Image



Figure 16 LSB substitution method Secret image can divide into RGB color channels.

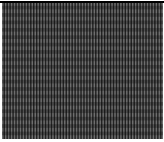
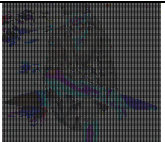
The snapshot of my tool for the after the visual cryptography the histogram will be check for the changes of image size as below.





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

#### 4.2.3.3 Decryption process justification with example

<b>Input</b>			<b>Egykcncz</b>
	(426 * 474) 1.41 KB	(1024 * 768) 34 KB	
	<b>Share 1</b>	<b>Share 2</b>	<b>Cipher Text</b>
<b>Output</b>	<b>Omsaidip</b>		
	<b>Original Text (Plain Text)</b>		

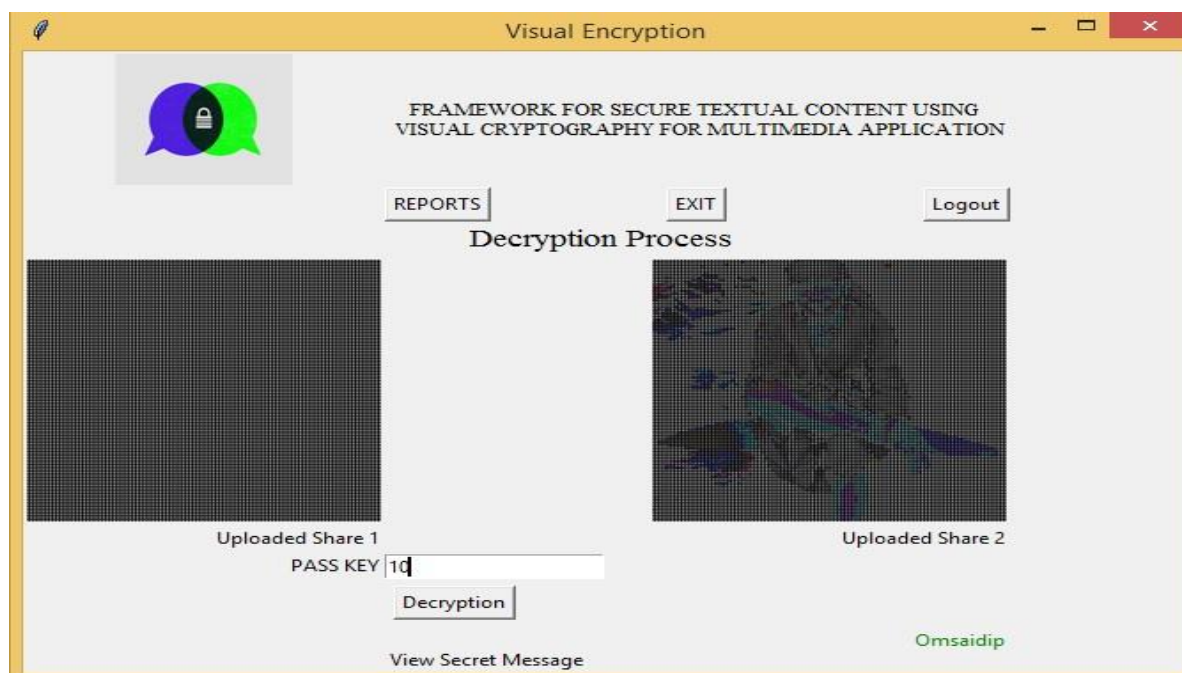
The procedure of decryption is straightforward. It is not necessary to restore the multimedia content if neither the stego picture nor the Cover Image2 is present. By overlapping the two shares, the Cover Image2 can be disclosed without the use of any mathematical processes. Using tracing, extraction, 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.


#### 4.2.3.4 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.

##### (1) 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 9 Calculation of Cover image hidden data, PSNR and RMSE

Cover Image	Hidden Data (KB)	PSNR (dB)	RMSE
 (213 * 237) 10.2 KB	84	81.07	0.039

## (2) 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 stego 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 Figure 8, 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.

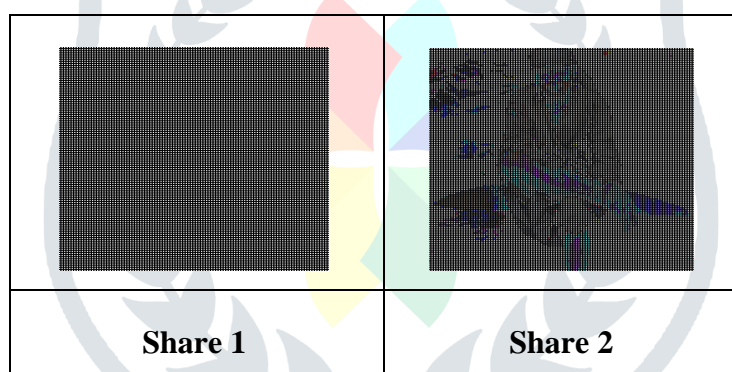


Figure 25 Shares created

## (3) 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.

#### (4) 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.

#### (5) 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.

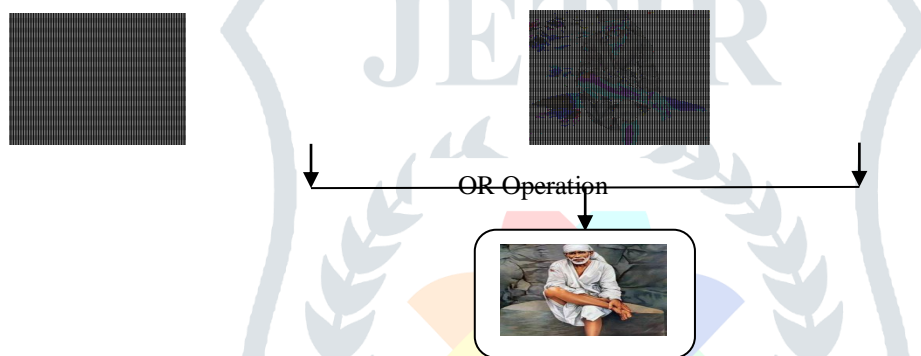


Figure 26 Result of Extraction

#### (6) 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<sub>f</sub> 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.

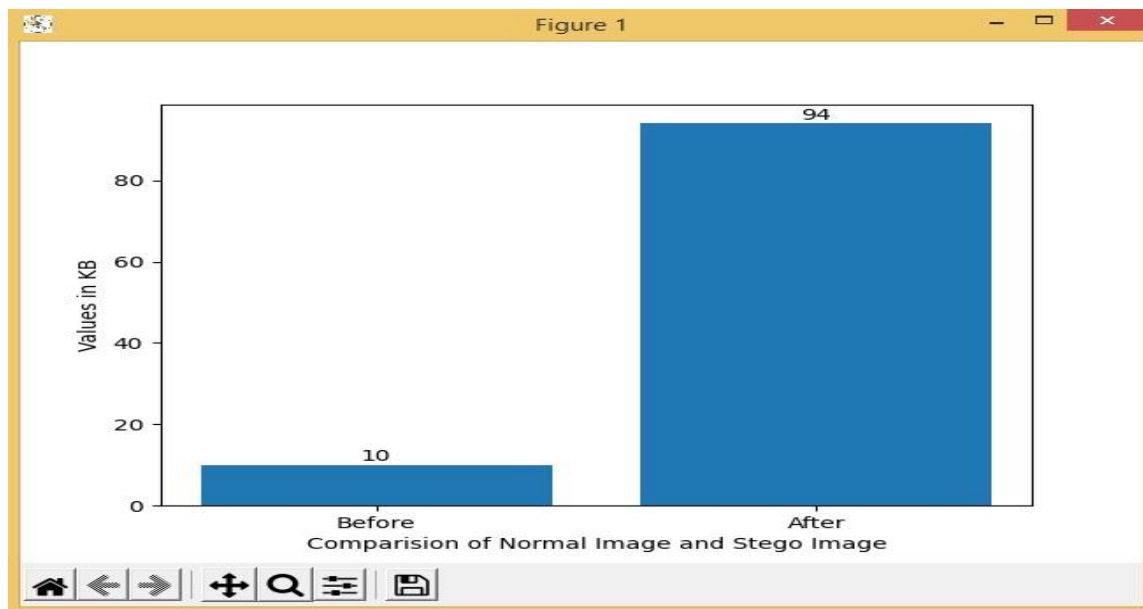
Table 10 Comparison of the Cover image with red, green & blue stego images

Cover Image	Red Stego Image	Green Stego Image	Blue Stego Image
			
(213 * 237) 10.2 KB	(213 * 237) 51.2 KB	(213 * 237) 53.7 KB	(213 * 237) 51 KB

## (7) 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.



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

### (8) 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.

### (9) 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.

## 5 SUMMARY

The RSA algorithm is a very interesting cryptographic algorithm, and it is definitely one of the best and most secure algorithms available as of today. It provides great encryption and is reliable in terms of security and performance. The encryption security relies on the fact that the prime numbers used during the key generation process must be large enough to be unbreakable, and this is quite interesting.

Even though the algorithm provides great encryption and it is reliable, the overall security really relies on the program developer or the algorithm user. If the user picks small prime numbers, it compromises a lot of the security that the RSA algorithm provides, and therefore is very vulnerable to cryptanalysis and brute-force attacks. So in other words, just using the RSA algorithm is not enough - it must be used properly in order to gain the encryption level it initially provides, as it must be used correct in terms of the key generation process and the initial preparation of the algorithm.



After critically analyzing RSA it is found that there are some flaws in it and to overcome these flaws a new algorithm has been proposed. The proposed algorithm increases the security of the system and also reduces the computation time. In future work can be carried out to decrease the complexity of the algorithm.

## 6 REFERENCES

- [1] E. Bertino, N. Shang and S. S. Wagstaff, "An Efficient TimeBound Hierarchical Key Management Scheme for Secure Broadcasting", IEEE Transactions on Dependable and Secure Computing, Vol. 5, No. 2, pp. 65-70, 2008.
- [2] Hua Li and J. Li, "A New Compact Dual-Core Architecture for AES Encryption and Decryption", IEEE Canadian Journal of Electrical and Computer Engineering, Vol. 33, No. 3, pp. 209- 213, 2008.
- [3] H. C. Williams, "A Modification of the RSA Public-Key Encryption Procedure", IEEE Transactions on Information Theory, Vol. 26, No. 6, pp. 726-729, 1980.
- [4] A.J. Elbirt, W. Yip, B. Chetwynd and C. Paar, "An FPGA Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 9, No. 4, pp. 545-557, 2001.
- [5] H. Chien, "Efficient Time-Bound Hierarchical Key Assignment Scheme", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 10, pp. 1301-1304, 2004.
- [6] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, Vol. 31, No. 4, pp. 469-472, 1985.
- [7] M. E. Hellman, "An Extension of the Shannon Theory Approach to Cryptography", IEEE Transactions on Information Theory, Vol. 23, No. 3, pp. 289-294, 1977.
- [8] Jason H. Li, B. Bhattacharjee, M. Yu and Levy, "A Scalable Key Management and Clustering Scheme for Wireless Adhoc and Sensor Networks", Journal of Future Generation Computer Systems, Elsevier Science Publishers, Vol. 24, pp. 860-869, 2008.
- [9] K. Bhatele, A. Sinhal and M. Pathak, "A Novel Approach to the Design of a New Hybrid Security Protocol Architecture", IEEE International Conference on Advanced Communication Control and Computing Technologies, pp.429-433, 2012.
- [10] M. Y. Wang, C. P. Su, C. L. Horng, C.W. Wu and C. T. Huang, "Single and Multi-core Configurable AES Architectures for Flexible Security", IEEE Transactions on Very Large Scale Integration Systems, Vol. 18, No. 4, pp. 541-552, 2010.