Data Hiding Tool Using Cryptography and Steganography

Capstone Project

Report *Submitted by*

SNo.	Register Number	Name
1	19BCY10026	Yogesh Kumar
2	19BCY10071	Suryakamal Jain
3	19BCY10083	Parth Chauhan
4	19BCY10085	Aditya Chouhan

in partial fulfillment of the requirements for the degree ofBachelor of Engineering and Technology

VIT BHOPAL UNIVERSITY
MADHYA PRADESH

Feb 2022

Contents

1.Introduction

We are developing a Data hiding tool using Cryptography and Steganography. Why have we decided to make it? Well, as we all know that data has become a particularly important part of most of our lives and everybody wants their data to be safe so that it cannot get misused by someone. So, let us assume that our important data is present on our computer and our system got compromised. In this situation, our data is present in a raw form inside our system, which can easily get misused by the intruder. So, keeping this problem in mind we are trying to make a tool that will keep our data safe with us in the hidden form. So, anyone unauthorized will not be able to access that data.

1.1 Cryptography

In computer science, the term "cryptography" refers to secure information and communication methods that use mathematical principles and a system of computations based on rules, or "algorithms," to change messages in ways that are challenging to read in such a way that only the message's originator and intended receiver can understand it. The word comes from the Greek word "kryptos," which means "hidden." These deterministic algorithms are employed in the creation of cryptographic keys, digital signatures, online browsing on the internet, and private communications like credit card transactions and email. The encrypted message is known as cipher text and the message to be encrypted is known as plain text.

.

1.2 Steganography

Steganography is a technique for hiding secret information by enclosing it in a conventional, non-secret file or communication; the information is subsequently retrieved at the intended location.

Steganography, which translates to "covered writing" or "hidden writing," is derived from the Greek terms "stegos," which means it can be used to hide almost any type of digital content, including text, images, videos, and audio. The content concealed by steganography is known as hidden text.

Prior to learning about steganography, it is critical to comprehend what pixels and color models are. The smallest unit of a picture is a pixel, and (assuming RGB) all pixel's colors are a function of the ratios of red, green, and blue. Therefore, a pixel with the values 0, 0, and 1 would represent 0 parts of red, 0 parts of green, and 1 part of blue, making it a blue pixel. The biggest number that can be represented in 8 digits in an 8-bit system is 11111111, which would be 255, and the smallest number can be represented in 8 digits is 00000000, which would be 0. A pixel can hold up to 8 digits (zeroes or ones) in an 8-bit system. Therefore, in an 8-bit situation, any pixel might accept any value for each of the colors that falls between 0 and 255. Let us imagine there are three pixels in a random 8-bit grid, and each pixel has the R, G, and B values listed below.

	The proportion of Red (R)	The proportion of Green (G)	The proportion of Blue (B)
Pixel 1	00101101	00011100	11011100
Pixel 2	10100110	11000100	00001100
Pixel 3	11010010	10101101	01100011

And if we wish to store the binary value of the number 200, which is 11001000, we can. and substitute each digit of that number with the pixel grid's least significant digit, which is shown in strong red letters. This digit is typically the last digit. The updated color palette would look like this:

	The proportion of Red (R)	The proportion of Green (G)	The proportion of Blue (B)
Pixel 1	00101101	00011101	11011100
Pixel 2	10100110	11000101	00001100
Pixel 3	11010010	10101100	01100011

As a result, the notified image would be indistinguishable from the original image in terms of color changes in the three channels for the three pixels of the original image.

1.3 The Combination

First, we will enhance our security by combining cryptography with steganography. As cryptography encrypts the data into an unreadable form hence, anyone who is not authorized by the owner of the data will not able to determine the true contents of the message. Now with steganography, we hide that data inside some other non-suspicious data.

Now, let us see **Why we are using the combination of cryptography and steganography and not using them individually?**

Limitations of using cryptography individually:

- The risk of the information being compromised if the key is taken is a noteworthy problem with the various encryption techniques. As a result, protecting the distribution of keys becomes an issue. The only thing that happens when cryptography is used in security systems is that the issue of secure communication is changed into one of key management.

- It can be easily detectable as anyone can decode that something is wrong with the data because of how it looks and then he can use some computer application and try to decrypt it.

Limitations of using steganography individually:

- It cannot modify the data's general structure. So, when an attacker reveals that steganography has been used, it gets broken and once hidden information is decoded, the data can be used by anyone.

1.4 AES ALGORITHM

The AES Encryption algorithm, commonly known as the Rijndael algorithm, which has a block/chunk size of 128 bits is chosen by the government of the US. These distinct blocks are converted using keys that are 128, 192, and 256 bits long. It then connects these blocks to create the ciphertext after encrypting each one separately.

To encrypt sensitive data, AES is used in hardware and software around the globe. It is crucial for government computer security, cybersecurity, and the protection of electronic data. It is structured on a substitution-permutation network, sometimes referred to as a substitution-permutation network. It comprises several interconnected processes, some of which involve bit shuffling (permutations). be buried inside any other sort of digital material, and it can be used to hide almost.

	AES	DES	3DES	Blowfish	TwoFish
Key length	128,192 or 256 bits	56 bits	112 and 168 bits (internally)	32 to 448 bits	128 bits 192 bits 256 bits
Block Size	128,192 or 256 bits	64 bits	64 bits	64 bits	128 bits
Speed	Extremely fast	Slow	Faster than DES	Fast	Moderate
Security	Excellent Security	Not secure enough	More secure than DES	Secure enough	Moderate
Cipher Type	Symmetric block cipher	Symmetric block cipher	Symmetric block cipher	Symmetric block cipher	128 bits
Possible Keys	2^{128}, 2^{192}, or 2^{256}	2^{256}	2^{112} or 2^{168}	2^{32} or 2^{448}	256
Attacks Prone to	—	Differential and linear cryptanalysis	Differential, brute force attack	Differential, brute force attack	Highly secure with still no cryptanalysis found
Developed in	2000 by NIST	1975 by IBM	1970 by IBM	1993 by Bruce Schneer	1998 by David Wagner, John Kelsey et al

Fig.1: Differences between some of the Common Encryption Algorithms

AES is effective in both software and hardware and is based on a design concept called a substitution-permutation network. AES does not make use of a Feistel network, unlike its predecessor DES. AES is a Rijndael variation with a key size of 128, 192, or 256 bits and a fixed block size of 128 bits. While the block and key sizes for Rijndael per se must be a multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits, they are defined to be any multiple of 32 bits. A specific finite field is used for most AES computations.

AES uses a 16-byte, 4x4 column-major order array with the b_0, b_1,, b_{15} known as the state to work.

$$\begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}$$

An AES cipher's key size determines how many rounds of transformation are required to turn the

input, known as the plaintext, into the desired output, known as the ciphertext.

There will be many rounds as:

- 10 rounds for 128-bit keys.
- 12 rounds for 192-bit keys.
- 14 rounds for 256-bit keys.

There are numerous processing steps in each cycle, one of which is dependent on the encryption key itself. Using the same encryption key, a series of reverse rounds are used to convert the ciphertext back into the original plaintext.

The algorithm's high-level description:

I. **Key Expansion –** The AES key schedule is used to create round keys from the cypher key. For each round plus one, AES needs a distinct 128-bit round key block.

II. **Initial round key addition:**

   a. **Add Round Key –** Bitwise XOR is used to combine each byte of the state with one byte of the round key.

III. **9, 11 or 13 rounds:**

   a. Sub Bytes – A non-linear substitution phase in which every bite is changed for an alternative based on a lookup table.

   b. Shift Rows – A transposition phase in which the final three rows of the state are cycled through a set number of times.

   c. Mix Columns – A linear mixing operation that combines each column's four bytes using the state's columns as its input.

   d. Add Round Key

IV. Final round (making 10, 12 or 14 rounds in total):

- **a.** Sub Bytes
- **b.** Shift Rows
- **c.** Add Round Key

The Sub Bytes step:

In the Sub Bytes stage, an 8-bit substitution box is used to replace each byte $a_{i,j}$ in the state array with a Sub Byte $S(a_{i,j})$.Remember that the state array is just plaintext/input before round 0. This process provides non-linearity in the cipher. The multiplicative inverse over GF (2^8), which is renowned for having good non-linearity qualities, is how the S-box was created. The S-box is built by fusing the inverse function with an invertible affine transformation, preventing assaults based on elementary algebraic features. The Inv Sub Bytes step, or the inverse of Sub Bytes is employed during decryption and entails taking the inverse of the affine transformation first before determining the multiplicative inverse.

The Shift Rows step:

The Shift Rows step cycles through each row of the state, shifting the bytes in each row by a specific offset. The first row is left unaltered for AES. The second row's bytes are shifted to the left. The third and fourth rows are also displaced by two and three offsets, respectively. [note 6] Bytes from each column of the input state are then used to create each column of the output state of the Shift Rows step. This step is crucial because if the columns were encrypted separately, AES would break down into four separate block ciphers.

The Mix Columns step:

The four bytes of each state column are mixed using an invertible linear transformation in the Mix Columns step. Each input byte has an impact on all four output bytes when using the Mix Columns function, which accepts four bytes as input and outputs four bytes. Diffusion in the cipher is provided by Mix Columns in conjunction with Shift Rows.

$$\begin{bmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_{0,j} \\ a_{1,j} \\ a_{2,j} \\ a_{3,j} \end{bmatrix} \qquad 0 \leq j \leq 3$$

Multiplication and addition of entries make up matrix multiplication. Entries are handled as polynomial coefficients of order x^7 bytes. Simply put, addition is XOR. The multiplicative polynomial display style x8+x4+x3+x+1 is modulo irreducible. If bits-by-bit processing is used, shifting should be followed by a conditional XOR with $1B_{16}$ if the shifted value is greater than FF_{16} (overflow must be corrected by subtraction of generating polynomial). These are exceptional situations involving the standard multiplication in GF (2^8). In a broader sense, each column is treated as a polynomial over GF (2^8) and is multiplied modulo $01_{16}\, z^4 + 01_{16}$ with a fixed polynomial c (z) = $03_{16}\ z^3 + \ 01_{16}\ z^2 + 01_{16}\ z + 02_{16}$ The coefficients are shown in their hexadecimal equivalent of the bit polynomials' binary representation from GF (2)[x]. The Mix Columns step can also be considered a multiplication by the specific MDS matrix displayed in the finite field GF (28).

The Add Round Key:

The subkey and state are joined in the Add RoundKey step. Rijndael's key schedule is used to create a subkey from the main key for each round; each subkey has the same size as the state. By utilizing bitwise XOR to combine each byte of the state with its matching byte from the subkey, the subkey is added.

2. Literature Review

S. No.	Reference	Year	Proposed Work	Conclusion
1	Domenico B. et al. **[6]**	2007	Utilized pictures as both cover objects and cryptographic keys.	**The system's performance is analyzed with that of the Vernam cipher and F5 algorithm.**
2	Dhawal S. et al. **[1]**	2010	Used cryptography and steganography for storing data securely.	**This paper concludes that for data to be stored securely, using both cryptography and steganography together is better.**
3	Dipti K. et al. **[7]**	2010	Integrated asymmetric cryptography, freq. domain image steganography, and a new security module	**Provided a highly secured system as it has used a total of 4 keys used for encrypting and embedding**
4	Dr.R. Sridevi et al. **[4]**	2010	First used image to hide the data and then used cryptographic technique on that image.	**Least Significant Bit (LSB) is used for steganography and AES is used for cryptography. As these methods are more secure than other.**
5	Phad Vitthal S. et al. **[16]**	2012	A high security model by combining cryptographic and Steganographic security	**A secret message is encrypted before hiding it in the cover image which gives high security to secret data**
6	Md. Khalid Imam et al. **[14]**	2014	A new algorithm has been proposed to ensure security and meet the requirements of steganography.	**Layer 1 implements authorization, layer 2 authentication, integrity and non-repudiation, layer 3 confidentiality and partial security, layer 4 robustness.**
7	Md. Khalid Imam et al. **[11]**	2015	A comparison of steganography and cryptography and how they work together to improve security.	**Secure combination combines steganography and cryptography to improve security, capability, and robustness of secure combination.**
8	Z. V. Patel et al. **[8]**	2015	Defined three types of combined crypto-steganography namely Pure, Secret Key and Public Key.	**Provided a brief analysis of which steganographic techniques are more appropriate for particular purposes.**
9	Mehndiratta et al. **[18]**	2015	A state-of-the-art investigation work around cryptography and steganography.	**Concluded that both cryptography and steganography techniques provide security for secrete information.**
10	R. Mishra and P. Bhanodiya **[15]**	2015	The paper introduces steganography and cryptography techniques and reviews their types, attacks, and previous work.	**Cryptography and steganography both have features to protect data over networks, but data compression measures should be taken. LSB is the most widely used technique.**
11	Zawa, Zin May et al. **[17]**	2015	A combined technique of using a block-based transformation and blowfish encryption.	**Cryptography makes decryption process difficult for attackers and steganography hides data completely not to be found by anyone, providing more security.**
12	Marwa E. et al. **[5]**	2016	Used modified AES for encryption i.e. (AES_MPK) and merged it with steganography for more security.	**It highlighted the need for a novel approach that integrates both techniques into one.**

13	Pranali R. et al. **[10]**	2016	Explored steganography and cryptography, and their respective methodology and limitations	**LZW provides a lossless compression method to extract text without changes in the message.**
14	Ako M. et al. **[2]**	2016	Provided GUI based system to hide data inside image and transmit that image to another user.	**Used Hash Least Significant bit (H-LSB) steganography technique to provide more security to user data.**
15	Sultan A. et al. **[3]**	2017	Provided comparative study of using cryptography and steganography together and which technique to use first.	**Showed that performing cryptography before steganography is more secure than performing steganography before cryptography.**
16	Ahmed AL-Shaaby et al. **[12]**	2017	Encrypted messages are encrypted using AES and SHA-2, and LSB is used to embed them in images, video, or audio.	**This paper combines steganography and cryptography to hide text in color images with the help of c# language.**
17	Francis, Neetha **[20]**	2018	Provided a two-step methodology for applying public steganography based on the matching method in various areas of a picture.	**It is better to combine steganography using LSB with cryptography to achieve a greater degree of security.**
18	Amit Singh et al. **[13]**	2018	Using cryptography to make data complex and then LZW compression increases the hiding capacity of an image by 50%, allowing double data hiding.	**LZW provides a lossless compression method to extract text without changes in the message.**
19	AksharaSree et al. **[19]**	2018	A survey of modern techniques which combined steganography and cryptography	**There is explosive growth in secure communication and information hiding which maintains information integrity.**
20	Mustafa S. et al. **[9]**	2019	Explored several ways to create a hybrid system by integrating steganography and cryptography.	**It highlighted the need for a novel approach that integrates both techniques into one.**

3. Proposed Work

3.1 Types of System

The types of systems that can be derived by combining the two types of cryptography i.e., Secret Key and Public Key Cryptography with image steganography are the following:

1. **Pure Steganography:**

This technique simply uses the steganography approach only without combining other methods. It is working on hiding information within the cover image or carrier.

2. **Secret Key steganography:**

The secret key steganography uses the combination of the secret key cryptography technique and the steganography approach. The idea of this type is to encrypt the secret message or data by a secret key approach and to hide the encrypted data within the cover carrier.

3. **Public Key Steganography:**

The last type of steganography is to combine the public key cryptography approach and the steganography approach. The idea of this type is to encrypt the secret data using the public key approach and then hide the encrypted data within the cover carrier.

Since secret key encryption quickly and efficiently encrypts vast amounts of data, we have used the combination of the secret key cryptography with the steganography approach. Also, the Symmetric algorithms are also simple to implement at the hardware level.

3.2 Combining Secret Key

First using cryptography on the data and after that performing steganography on the encrypted data: -

In this approach we first encrypt the data into unreadable form after that we hide that encrypted data inside a cover image provided by the user, using steganographic techniques.

We reverse the process at the time of extraction. We first extract the encrypted data from the image and after that perform decryption on that encrypted data to gain the original data.

First using steganography on the data and after that performing cryptography on the image: -

In this approach, we first take the data from the user and hide that data inside an image. Then we use cryptographic techniques to encrypt the image. At the time of getting the original data. We first decrypt the data after that we perform an extraction process to get the original data that is hidden inside the image.

We have gone through many research papers and found that performing cryptography before steganography is the most fashionable way for combining both these techniques. This way has higher security as compared to using steganography before cryptography and less risk exposing and performing steganography before cryptography gives cipher text as our final result. Cipher text can easily catch hackers' attention as some vital information is present behind that. But it is hard in the case of images as it is less suspicious.

.

3.3 Flow-Chart

The below flow chart shows the working of the tool.

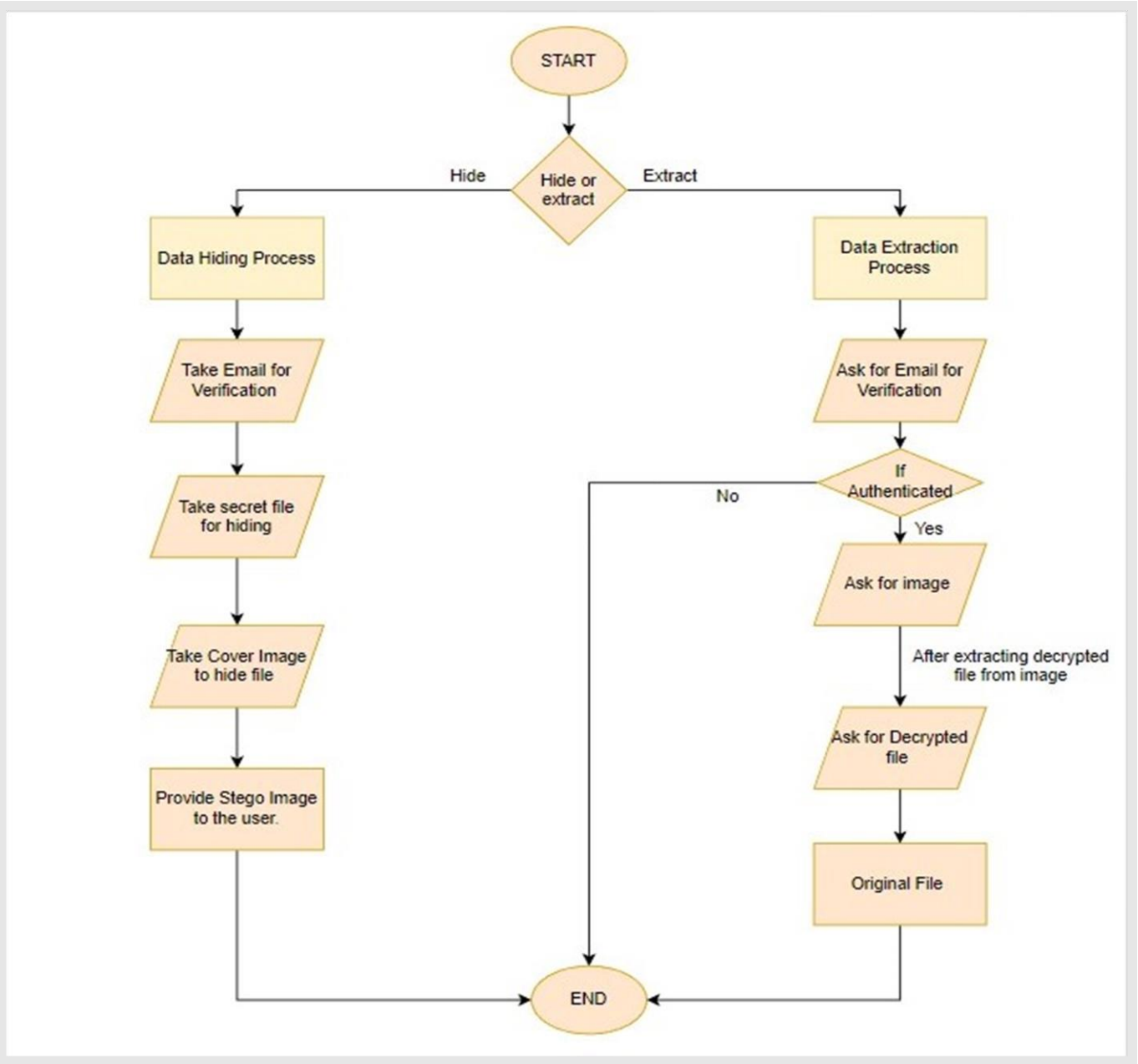

Fig.2

3.4 THE ALGORITHM

Encrypting and Embedding Process

a) After the user selects encrypts as an option, we ask them to provide the email so that we can verify the user's authenticity at the time of the extraction process.

b) Then we ask for the file that the user wants to hide.

c) After getting the file from the user we start our encryption process. We first encrypt the provided file.

d) Then we ask for a cover image from the user in which we will be hiding that file.

e) After that, we hide the file behind the image and provide the final image to the user which they need to keep with them for further use and extraction of data from it.

f) After getting the final stego image, the user can delete all the instances of that data which are hidden inside our stego image.

Decryption and Extraction Process

a) Now as we are done with the encryption or embedding process, we can now look at the process of extracting.

b) When the user selects the decryption option, the first step that occurs is email verification

c) After validating the user through email verification, we can move to our extraction process

d) For extraction, we first send a secret key to the user's authentic email address.

e) After getting the key from the user we extract the data which is hidden behind the image.

f) Then we perform a decryption process to get the original data from the image for use

3.4 System Architecture Diagram

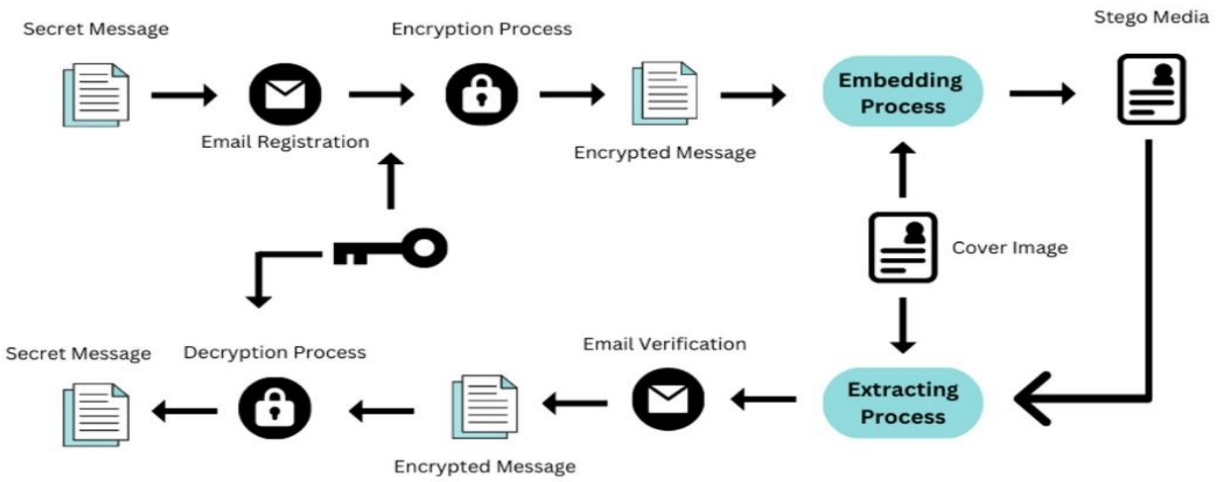

Fig.3

1. So, firstly, the user will be authenticated using their email id. After this, they must provide the data in zip file format to the tool they want to secure.

2. After this, it will first encrypt the data using the cryptography module of python then it will hide that encrypted data inside an image by using steganography (specifically image steganography).

3. Now, if a user wants to get back their data, they must prove their legitimacy by entering the same email id they entered at the time of embedding.

4. If it matches the tool will share a unique key to that email id and if the user enters the same key only then they can get back their original data.

Working of Cryptography

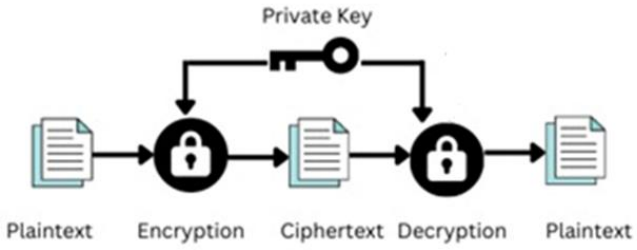

Fig.4: Encryption and Decryption Process

In the encryption process, we must give the data that is called plaintext as an input which we want to encrypt to the encryption procedure. After that, it will encrypt that data using the private key and provide you with the ciphertext, also known as encrypted text as output.

In the decryption process, the only difference is that We give the ciphertext as an input that we want to encrypt to the decryption procedure. After that, it will decrypt that using the same private key which we used at the time of encryption, and it will provide you with the original data or the plaintext as output.

Working of Steganography

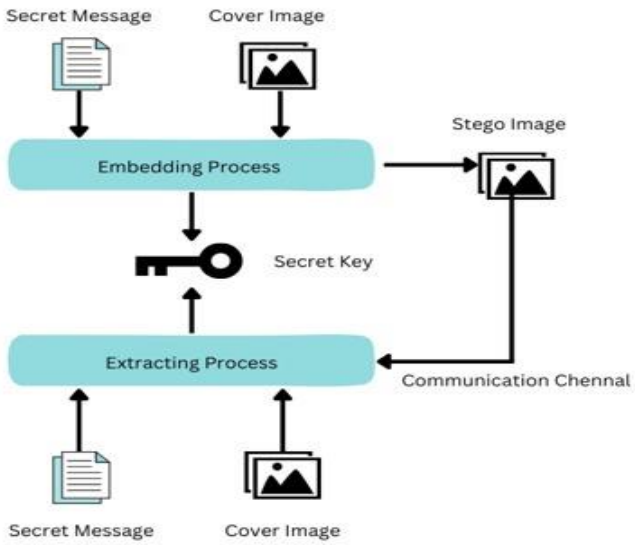

Fig.5: Image Steganography Process

For the embedding procedure in image steganography, it will first ask for the information you want to conceal within the image, which will be called cover image. Following that, an image in which we want to hide that information will be requested as input. Now, the steganography procedure will conceal that data inside of that image. This final picture is known as a stego picture.

During the extraction step, the embedding procedure will be reversed. It will first request the stego image of the tool that contains the information as input. After that, when the cover image and hidden information are separated, you will be provided with the original information and cover image individually.

4. Result Analysis

We have analyzed the difference in the size and dimensions of original image i.e. the image before embedding any data inside it and the stego image i.e. the image we got after embedding our data with the help of this tool and the findings are mentioned below:

From the figures 6 to 10 given below we can see that we have taken 2 scenarios:

1. In the first scenario, we took an image of 86.5 kb (0.0865 mb) as the cover image and have embedded data of size 8.33 mb inside it. After embedding we got the final image with size of 8.42 mb.
2. In the second scenario we took an image of 30.2 kb (0.0302 mb) as the cover image and have embedded data of size 22.9 mb inside it. After embedding we got the final image with size of 23.3 mb.

   In both scenarios the dimensions i.e., height and the width of the cover and stego image remains same.

   The size got increased because we are using LSB as mentioned earlier as our steganography technique which modifies the pixels of the cover image by hiding the secret data within the pixels itself, the same we can see in our results that only the size of the stego image is increasing without any change in dimensions and any distortion.

   So, we can say that our tool is generating the desired results and is working efficiently and there is no time delay while executing the code.

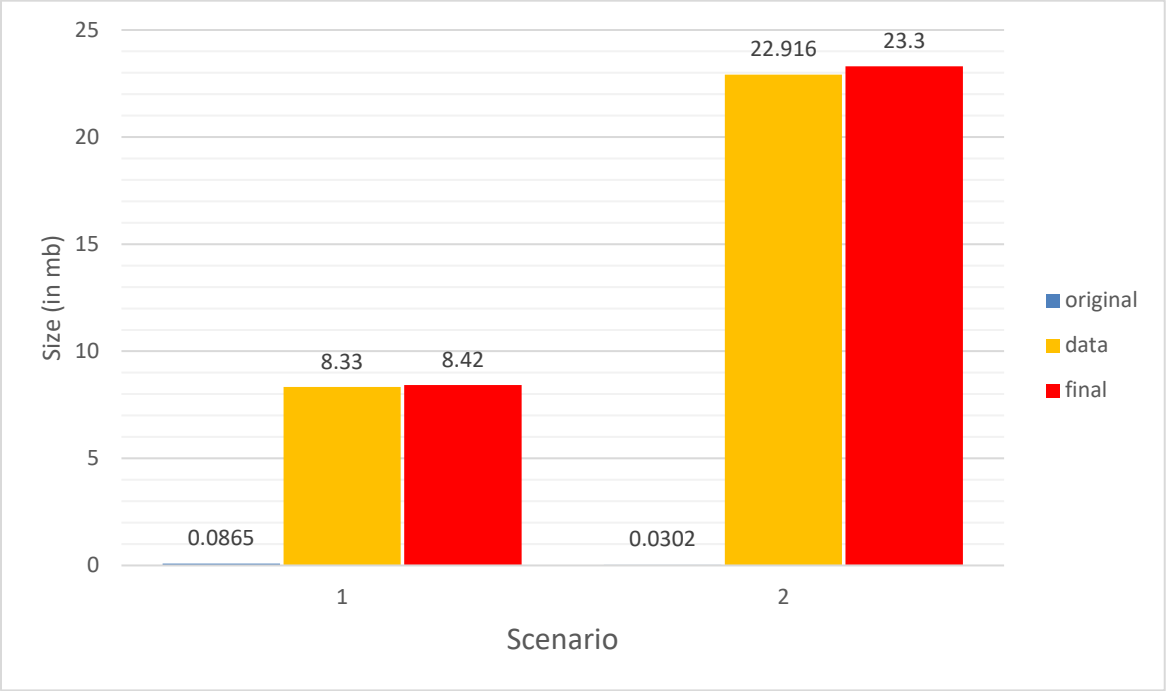
25
20
15
10
5
0
Size (in mb)
22.916
23.3
8.33
8.42
0.0865
0.0302
1
2
Scenario
original
data
final

Fig.6

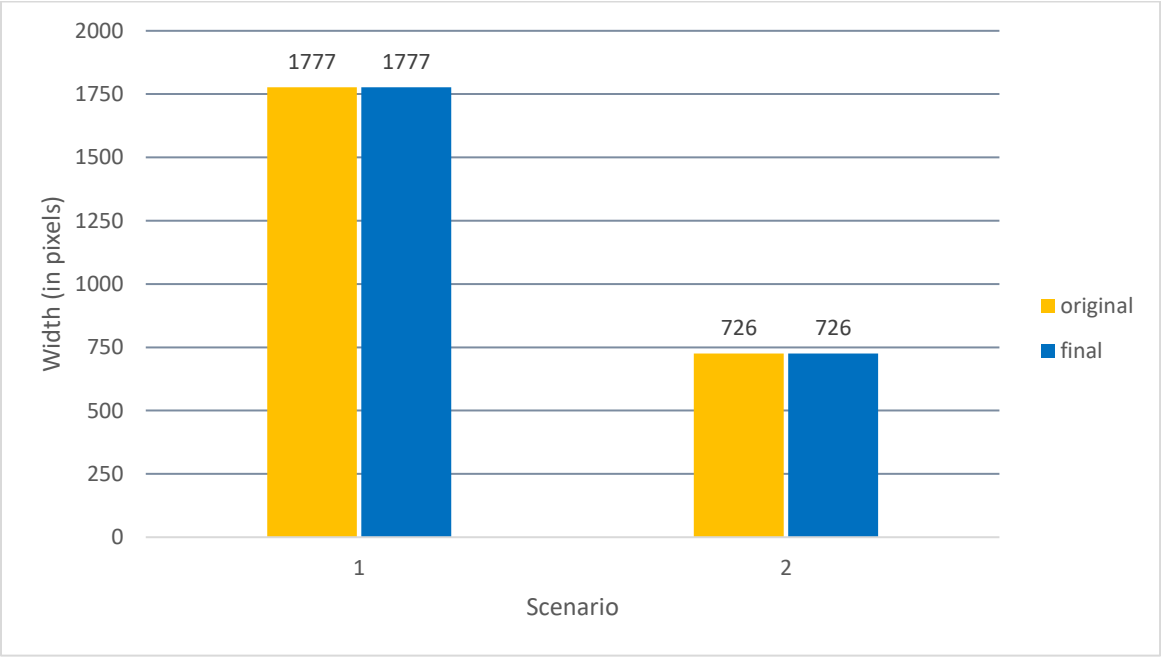
2000
1750
1500
1250
1000
750
500
250
0
Width (in pixels)
1777
1777
726
726
1
2
Scenario
original
final

Fig.7

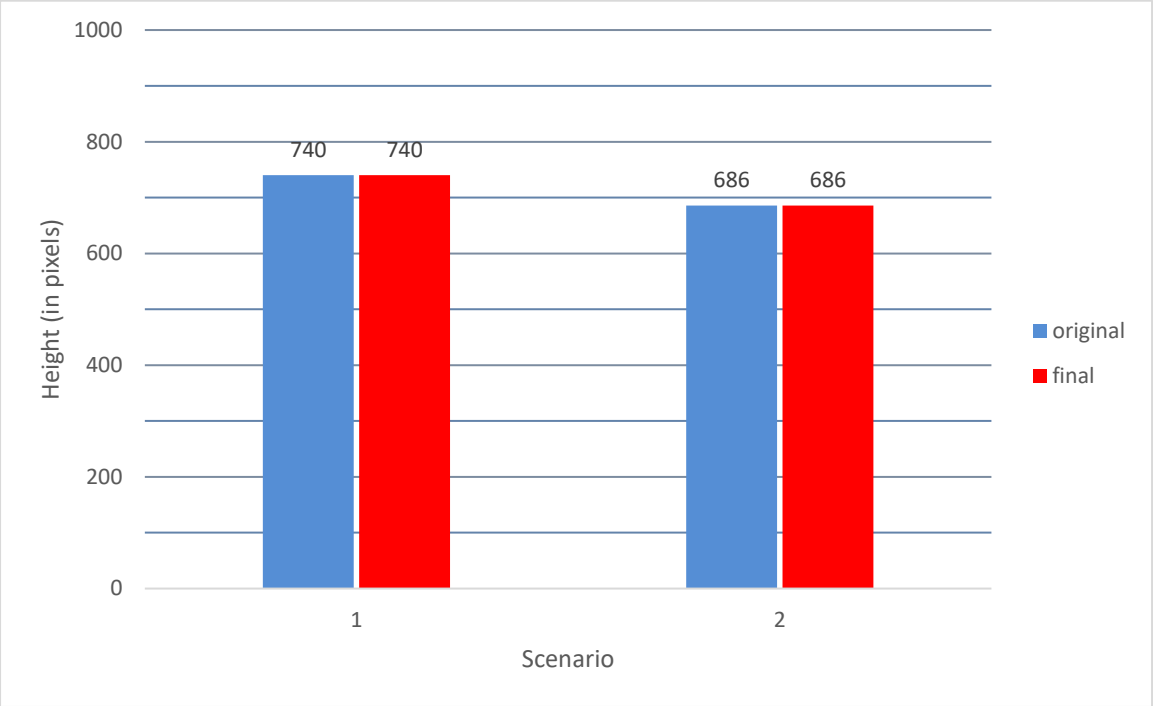
1000
800
600
400
200
0
Height (in pixels)
740
740
686
686
1
2
Scenario
original
final

Fig. 8

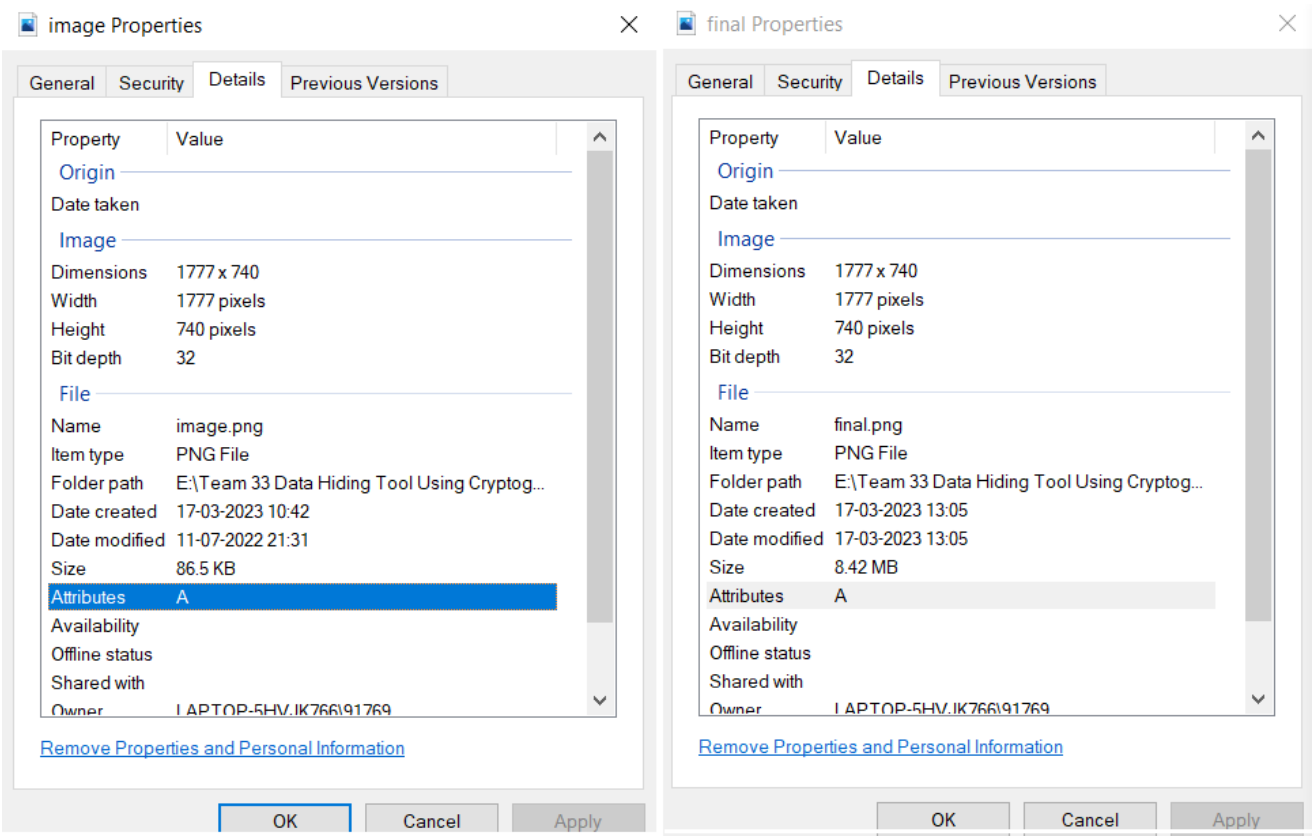
image Properties
General
Security
Details
Previous Versions
Property
Value
Origin
Date taken
Image
Dimensions 1777 x 740
Width 1777 pixels
Height 740 pixels
Bit depth 32
File
Name image.png
Item type PNG File
Folder path E:\Team 33 Data Hiding Tool Using Cryptog...
Date created 17-03-2023 10:42
Date modified 11-07-2022 21:31
Size 86.5 KB
Attributes A
Availability
Offline status
Shared with
Owner LAPTOP-5HVJK766\91769
Remove Properties and Personal Information
OK
Cancel
Apply
final Properties
General
Security
Details
Previous Versions
Property
Value
Origin
Date taken
Image
Dimensions 1777 x 740
Width 1777 pixels
Height 740 pixels
Bit depth 32
File
Name final.png
Item type PNG File
Folder path E:\Team 33 Data Hiding Tool Using Cryptog...
Date created 17-03-2023 13:05
Date modified 17-03-2023 13:05
Size 8.42 MB
Attributes A
Availability
Offline status
Shared with
Owner LAPTOP-5HVJK766\91769
Remove Properties and Personal Information
OK
Cancel
Apply

Fig.9

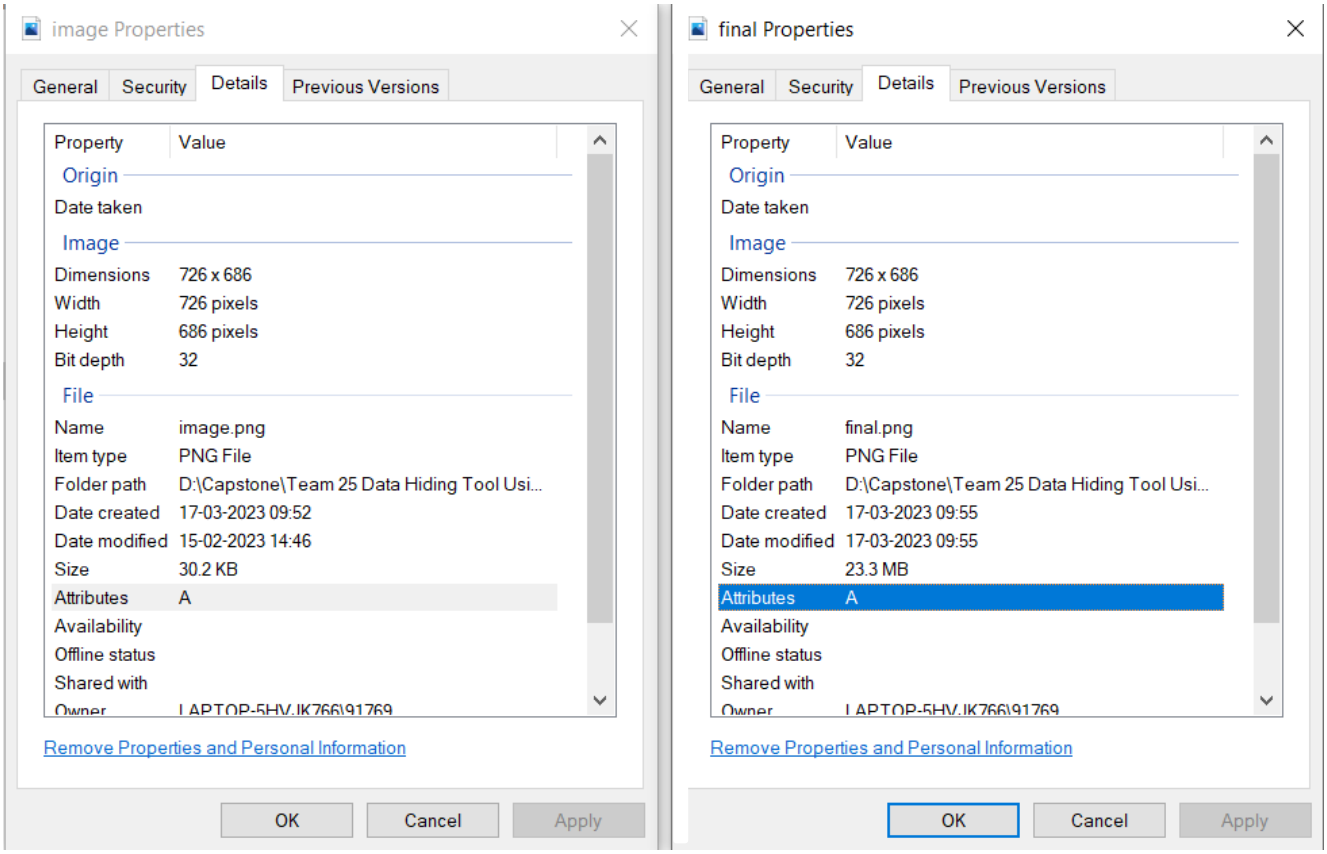
image Properties
General
Security
Details
Previous Versions
Property
Value
Origin
Date taken
Image
Dimensions 726 x 686
Width 726 pixels
Height 686 pixels
Bit depth 32
File
Name image.png
Item type PNG File
Folder path D:\Capstone\Team 25 Data Hiding Tool Usi...
Date created 17-03-2023 09:52
Date modified 15-02-2023 14:46
Size 30.2 KB
Attributes A
Availability
Offline status
Shared with
Owner LAPTOP-5HVJK766\91769
Remove Properties and Personal Information
OK
Cancel
Apply
final Properties
General
Security
Details
Previous Versions
Property
Value
Origin
Date taken
Image
Dimensions 726 x 686
Width 726 pixels
Height 686 pixels
Bit depth 32
File
Name final.png
Item type PNG File
Folder path D:\Capstone\Team 25 Data Hiding Tool Usi...
Date created 17-03-2023 09:55
Date modified 17-03-2023 09:55
Size 23.3 MB
Attributes A
Availability
Offline status
Shared with
Owner LAPTOP-5HVJK766\91769
Remove Properties and Personal Information
OK
Cancel
Apply

Fig. 10

5.Conclusion

After reading the above sections, we are sure that now you have a very brief understanding of what we are developing, how we are going to do it, and the techniques and algorithms we are using to create our project's functionality. So, in this section, we will summarize everything and then discuss the benefits of using our project and what we hope to achieve with it. We are creating a data hiding tool so that people using our project can more safely and discreetly hide their data. The job of our tool is to encrypt and embed any file provided by the user, as well as extract and decrypt the file when the user wants it. In this project, we combined two separate cybersecurity techniques (cryptography and steganography) to provide the highest level of security and abstraction for user data.

The algorithms we are implementing in our project help solve many of the problems that we and others typically face when trying to use these types of software. We want our users to have a friendly experience while using the software, users only need to click a button to access any function. Our project provides the users with the possibility to upload any file format, be it an image, document, or video, etc. In our project, we also ensure that our users are not limited to uploading small files but also provide them with the possibility to upload large files. Since we are using steganography, the final output for our users is an image containing encrypted user files. This image must be kept by the user so that if he wants to retrieve his data, he must provide it to our software so it can be extracted and that the user will get their original files. Our project will also include an about page so users can understand how we protect their data, how they can use the tool and provide the users with complete transparency and security.

6. Bibliography

1. Dhawal Seth, L. Ramanathan, Abhishek Pandey, "Security Enhancement: Combining Cryptography and Steganography".

2. Ako Muhammad Abdullah, Roza Hikmat Hama Aziz, "New Approaches to Encrypt and Decrypt Data in Image using Cryptography and Steganography Algorithm".

3. Sultan Almuhammadi and Ahmed Al-Shaaby, "A survey on recent approaches combining cryptography and steganography".

4. Dr.R. Sridevi, Vijaya Lakshmi Paruchuri, K.S. SadaSiva Rao, "Image Steganography combined with Cryptography".

5. Marwa E. Saleh, Abdelmgeid A. Aly, Fatma A. Omara, "Data Security Using Cryptography and Steganography Techniques".

6. Domenico, B. and Luca L., "Image-Based Steganography and Cryptography".

7. Dipti Kapoor Sarmah, Neha Bajpai, "Proposed System for data hiding using Cryptography and Steganography", International Journal of Computer Applications.

8. Z. V. Patel, S. A. Gadhiya, "A Survey Paper on Steganography and Cryptography ", International Multidisciplinary Research Journal (www.rhimrj.com), ISSN: 2349-7637, Volume-2 Issue-5, May-2015.

9. Mustafa S. Taha, Mohd Shafry Mohd Rahim, Sameer A. lafta, Mohammed M. Hashim, H. M. Alzuabidi,"Combination of Steganography and Cryptography: A short Survey ".

10. Pranali R. Ekatpure, Rutuja N Benkar, "A Comparative Study of Steganography and Cryptography", International Journal of Science and Research (https://www.ijsr.net/).

11. Md. Khalid Imam Rahmani, Mr. Amit Kumar Goyal, Manisha Mudgal "Study of Cryptography and Steganography System" International Journal of Engineering and Computer Science ISSN: 2319-7242 Volume 4 Issue 8 Aug 2015, Page No. 13685-13687.

12. Ahmed AL-Shaaby, Talal AlKharobi “Cryptography and Steganography: New Approach" Society for Science and Education United Kingdom VOLUME 5, NO. 6 ISSN: 2054 –7420.

13. Amit Singh, Nayan Solanki, Prof. Foram Shah “Hybrid Approach For Securing Data" IOSR Journal of Engineering (IOSRJEN) ISSN (e): 2250-3021, ISSN (p): 2278-8719 Volume 9.

14. Md. Khalid Imam Rahmani, Kamiya Arora, Naina Pal "A Crypto-Steganography: A Survey" International Journal of Advanced Computer Science and Applications, Vol. 5, No. 7, 2014.

15. R. Mishra and P. Bhanodiya, "A review on steganography and cryptography," 2015 International Conference on Advances in Computer Engineering and Applications, Ghaziabad, India, 2015, pp. 119-122, doi: 10.1109/ICACEA.2015.7164679.

16. Phad Vitthal S., Bhosale Rajkumar S., Panhalkar Archana R. "A Novel Security Scheme for Secret Data using Cryptography and Steganography" I. J. Computer Network and Information Security, 2012, 2, 36-42 Published Online March 2012 in MECS (http://www.mecs-press.org/)

17. Zawa, Zin May, and Su Wai Phyob. “Security Enhancement System Based on the Integration of Cryptography and Steganography.” (2015).

18. Mehndiratta, Aarti. “Data Hiding System Using Cryptography & Steganography: A Comprehensive Modern Investigation.” (2015).

19. Akshara Sree Challa et al. “Role of cryptography and steganography in securing digital information: a review.” (2018).

20. Francis, Neetha. “Information Security using Cryptography and Steganography.” International journal of engineering research and technology 3 (2018).