APPLICATIONS IN BANKING USING VISUAL CRYPTOGRAPHY

Submitted in partial fulfillment of the requirements of the degree of

Bachelor of Engineering

Submitted by

PARAG SALGAONKAR

Guided By

Prof. Poonam Shrivastava

Department of

ELECTRONICS & TELECOMMUNICATION ENGINEERING

WATUMULL INSTITUTE OF ELECTRONICS ENGINEERING AND COMPUTER TECHNOLOGY

Year 2015-2016

WATUMULL INSTITUTE OF ELECTRONICS ENGINEERING AND COMPUTER TECHNOLOGY, WORLI, MUMBAI - 400018

Year 2015-2016

Vision

To become one of the leaders in the field of Electronic and Telecommunication education by promoting the aptitude of inquiry and study among the budding engineers for customizing and popularizing technology for the masses.

Mission

To Produce conductive teaching and learning environment for the stakeholders, thereby creating suitable workforce for the service sectors.

Watumull Institute Of Electronics Engineering And Computer Technology

IN AFFILIATE OF THE UNIVERSITY OF MUMBAI

Project Report Approval for B. E.

This Dissertation entitled "**Application In Banking Using Visual Cryptography**" by **Parag Salgaonkar** is approved for the degree of Bachelor of Engineering in Electronics and Telecommunication.

Supervisors /Guide's

Prof. Poonam Shrivastava	
-----------------------------	----------
(Name)	(Signature)

Head of Department

Dr. Sunita Sharma	
-----------------------------	-----------
(Name)	(Signature)

Principal

Dr. Sunita Sharma	
-----------------------------	------------
(Name)	(Signature)

Examiners

1.	
-----------------------------	------------
(Name)	(Signature)
2.	
-----------------------------	-------------
(Name)	(Signature)

Date:
Place: Mumbai

Thesis Approval for Bachelors of Engineering

This project report entitled “Application In Banking Using Visual Cryptography” by Parag Salgaonkar is approved for the degree of Electronics and Telecommunication Engineering.

Examiners

1. ………………………………..
2. ………………………………..

Dr. Sunita Sharma

(HOD & Principal)

Date:

Place: Mumbai

ACKNOWLEDGEMENT

Among the wide panorama of people who provided us help and motivation to complete our project, we are grateful in presenting to you the rare shades of technology by documenting project "Applications in Banking using Visual Cryptography"

We wish to express our deep sense of gratitude to our **HOD & Principal Dr. Sunita Sharma** for providing us the facilities to bring this project a success. Her encouragement proved to be a boon in the path of our achievement.

We are thankful to our guide, Prof. Poonam Shrivastava for giving us valuable inputs for the development of our project.

Last but not the least we would also like to thank our non-technical/non-teaching staff.

PARAG SALGAONKAR

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Parag Salgaonkar	
--------------------------	-------------------
(Name of Student)	(Signature)

Date:

Place: Mumbai

ABSTRACT

Core banking is a set of services provided by a group of networked bank branches. Bank customers may access their funds and perform other simple transactions from any of the member branch offices. The major issue in core banking is the authenticity of the customer, due to unavoidable hacking of the databases on the internet. To solve this problem of authentication, we are proposing an algorithm based on image processing, i.e. visual cryptography using time constraints. Visual Cryptography is a special encryption technique to hide information in images in such a way that it can be decrypted by the human visual system. But the encryption technique needs cryptographic computation to embed the system time into the encrypted image. The customer has to enter a validity period during encryption. This validity is the time available to the receiver to decrypt the image to obtain secret data. While decoding, the systems checks whether the system time is less than the validity image, thus destroying the image after validity period.

Keywords – Visual Cryptography, secret sharing, time constraint.

Contents

APPLICATION IN BANKING USING VISUAL CRYPTOGRAPHY........................i

ACKNOWLEDGEMENT........................v

ABSTRACT........................1

1. INTRODUCTION........................4

1.1 Motivation........................5

1.2 Problem Statement........................5

1.3 What is Cryptography used for?........................6

1.4 Cryptography Tools........................6

1.5 Classification of steganographic techniques:........................9

1.6 Cryptography and Security........................11

1.7 Protecting Against Malicious Cryptography........................11

1.8 Points to remember........................13

1.9 Requirement Analysis........................13

2. REVIEW OF LITERATURE AND PROBLEM FORMULATION........................14

2.1 Covert communication techniques........................15

2.2 ALGORITHMS USED........................16

2.2.1 LSB algorithm........................16

2.2.2 DISCRETE WAVELET TRANSFORM (DWT) Technique........................17

2.2.3 DCT algorithm........................18

2.3 Detecting Cryptography........................19

2.4 Applications of cryptography........................22

2.5 Limitations of algorithms used above:........................22

3. VC USING LEAST BIT SUBSTITUTION TECHNIQUE........................23

3.1 Encryption Scheme........................23

3.2 Decryption Scheme........................25

3.3 Technology Used........................27

4. PROJECT IMPLEMENTATION........................29

4.1 Encryption Interface........................29

4.2 Decryption Interface........................32

SUMMARY AND CONCLUSION........................36

Summary........................36

Conclusion........................36

Future work 36
REFERENCES 37
APPENDIX I 38
Source Code 38
Encryption 38
Decryption 43

List of Figures

FIGURE 1.1 JPEG FILE WITHOUT EMBEDDED TEXT 7
FIGURE 1.2 TIFF FILE WITH EMBEDDED TEXT 7
FIGURE 1.3 CLASSIFICATION OF CRYPTOGRAPHY TECHNIQUES 9
FIGURE 2.1 GENERAL BLOCK DIAGRAM FOR CRYPTOGRAPHY 16
FIGURE 2.2 WAVELET DECOMPOSITION INTO ONE LEVEL 17
FIGURE 2.3 SPATIAL DOMAIN TO FREQUENCY DOMAIN TRANSFORMATION 18
FIGURE 2.4 WAVELET DECOMPOSITION INTO THREE LEVELS 18
FIGURE 3.1 ENCRYTION SCHEME 23
FIGURE 3.2 EXAMPLE OF THE ENCRYPTION SCHEME 24
FIGURE 3.3 DECRYPTION SCHEME 25
FIGURE 3.4 EXAMPLE OF THE DECRYPTION SCHEME 26
FIGURE 3.5 GUI FIGURE 27
FIGURE 3.6 PROPERTY INSPECTOR 28
FIGURE 4.1 ENCRYPTION INTERFACE 29
FIGURE 4.2 ENCRYPTION EXAMPLE 30
FIGURE 4.3 ENCRYPTION OUTPUT 31
FIGURE 4.4 DECRYPTION INTERFACE 32
FIGURE 4.5 DECRYPTION EXAMPLE 33
FIGURE 4.6 DECRYPTION OUTPUT 34
FIGURE 4.7 DECRYPTION OUTPUT(SESSION TIMED OUT) 35

List of Tables

TABLE 2.1 COMPARISON OF COVERT COMMUNICATION TECHNIQUES 15

1. INTRODUCTION

Today, most applications are only as secure as their underlying system. Since the design and technology of middleware has improved steadily, their detection is a difficult problem. As a result, it is nearly impossible to be sure whether a computer that is connected to the internet can be considered trustworthy and secure or not. The question is how to handle applications that require a high level of security, such as core banking and internet banking.

In a core banking system, there is a chance of encountering forged signature for transaction. And in the net banking system, the password of customer may be hacked and misused. Thus security is still a challenge in these applications. Here, we propose a technique to secure the customer information and to prevent the possible forgery of password hacking. The concept of image processing, an visual cryptography is used. Image processing is a technique of processing an input image and to get the output as either improved form of the same image and/or characteristics of the input image.

Visual Cryptography (VC) is a method of encrypting a secret image into a secondary shadow image. Basically, Visual Cryptography Scheme is an encryption method that uses combinatorial techniques to encode secret written materials. The idea is to convert the written material into an image and encode this image into n shadow images.

1.1 Motivation

Historically, ancient Greeks covered tablets with wax and used them to write on. The tablets were composed of wooden slabs. A layer of melted wax was poured over the wood and allowed to harden as it dried. Hidden messages could be carved into the wood prior to covering the slab. When the melted wax was poured over the slab, the now concealed message was later revealed by the recipient when they re-melted the wax and poured it from the tablet.

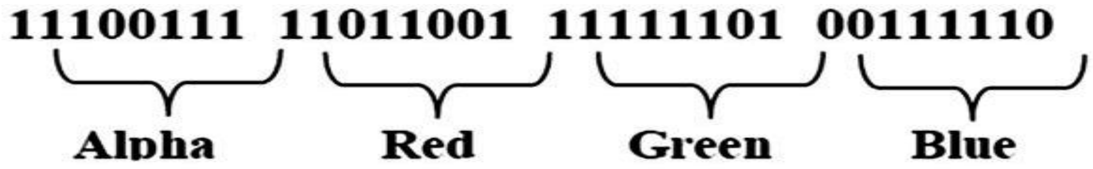
From the 1stcenturythrough World War II invisible inks were often used to conceal hidden messages. At first, the inks were organic substances that oxidized when heated. The heat reaction revealed the hidden message. As time passed, compounds and substances were chosen based on desirable chemical reactions. When the recipient mixed the compounds used to write the invisible message with a reactive agent, the resulting chemical reaction revealed the hidden data. Today, some commonly used compounds are visible when placed under an ultraviolet light.

One of the group members was interning at a Hardware networking firm, one of the engineers was working on the network of an online trading company and mentioned how vulnerable the network was to a cyber attack. This lack of security was our main motivation to develop our final year project to improve the network security in places which has trading important documents and exchange of large sums of money.

1.2 Problem Statement

The main objective is to develop an algorithm to embed a secret image into a cover (host) image without causing significant perceptual distortion in the stego -image. Further the objective is to extract the embedded image from the stego-image without loss of information.

Visual cryptography is a cryptographic technique which allows visual information (Image, text, etc) to be encrypted in such a way that the decryption can be performed by the human visual system without the aid of computers. Image is a multimedia component sensed by human. The smallest element of a digital image is pixel. In a 32 bit digital image each pixel consists of 32 bits, which is divided into four parts, namely Alpha, Red, Green and Blue; each with 8 bits. Alpha part represents degree of transparency. If all bits of Alpha part are ‗0', then the image is fully transparent. This is represented in the following figure.



1.3 What is Visual Cryptography (VC) used for?

Like many security tools, VC can be used for a variety of reasons, some good, some not so good. Legitimate purposes can include applications like watermarking images for reasons such as copyright protection. Digital watermarks (also known as fingerprinting, significant especially in copyrighting material) are similar to cryptography in that they are overlaid in files, which appear to be part of the original file and are thus not easily detectable by the average person. VC can also be used as a way to make a substitute for a one-way hash value (where you take a variable length input and create a static length output string to verify that no changes have been made to the original variable length input). Further, VC can be used to tag notes to online images (like post-it notes attached to paper files). Finally, VC can be used to maintain the confidentiality of valuable information, to protect the data from possible sabotage, theft, or unauthorized viewing.
Unfortunately, VC can also be used for illegitimate reasons. For instance, if someone was trying to steal data, they could conceal it in another file or files and send it out in an innocent looking email or file transfer. Furthermore, a person with a hobby of saving illegitimate data to their hard drive may choose to hide the evidence through the use of VC. And, as was pointed out in the concern for terrorist purposes, it can be used as a means of covert communication. Of course, this can be both a legitimate and an illegitimate application.

How is it different from steganography?
While Cryptography and Steganography are related, there is a difference between the two. Cryptography is used to scramble messages so that they cannot be understood. It does not hide the fact that the message exists. Steganography, on the other hand, conceals the fact that the message exists by hiding the actual message in another.

1.4 Visual Cryptography Tools

There are a vast number of tools that are available for VC. An important distinction that should be made among the tools available today is the difference between tools that do VC, and tools that do cryptoanalysis, which is the method of detecting cryptography and destroying the original message. Cryptoanalysis focuses on this aspect, as opposed to simply discovering and decrypting the message, because this can be difficult to do unless the encryption keys are known.
A comprehensive discussion of vc tools is beyond the scope of this article. However, there are many good places to find vc tools on the Net. The plethora of tools available also tends to span the spectrum of operating systems.

Given below is an example of text embedded in an image. As you can see by the *before* and *after* pictures below, it is very hard to tell them apart, embedded message or not.

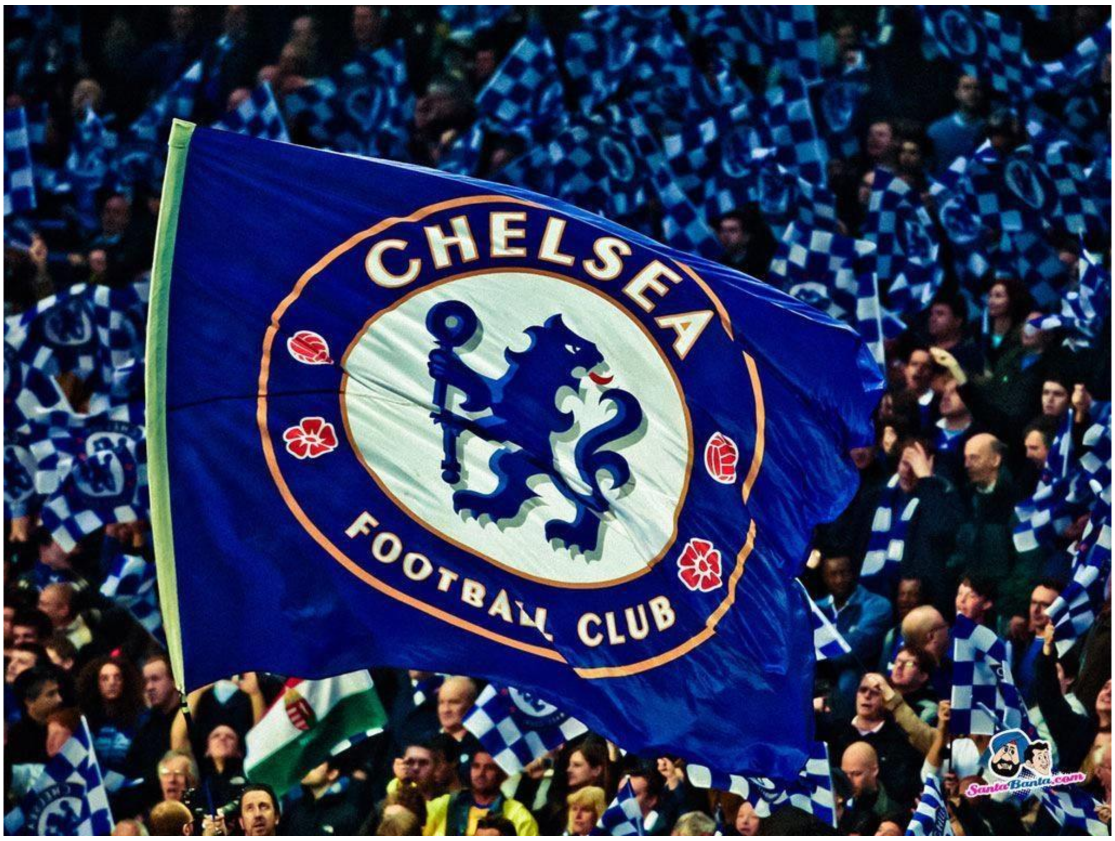


Figure 1.1JPEG file without embedded text

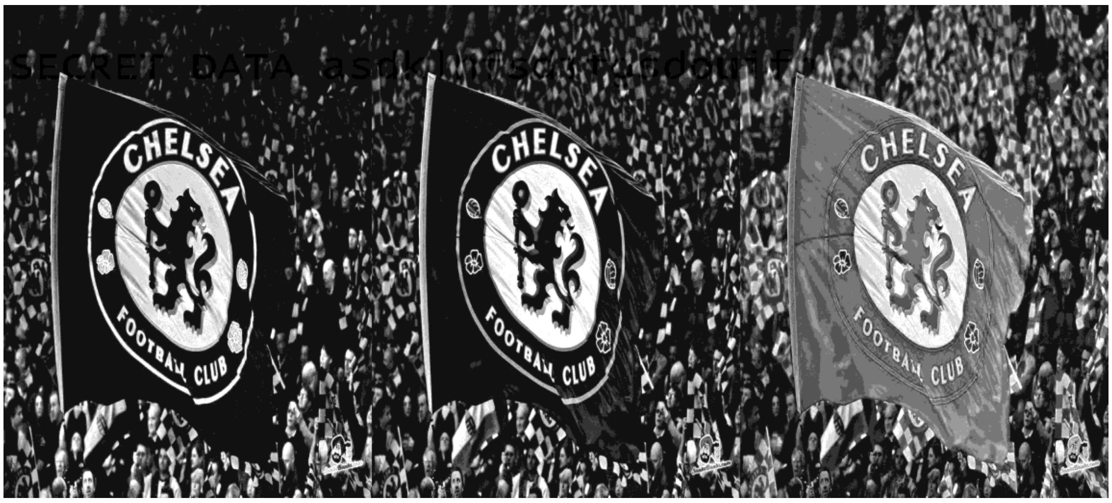


Figure 1.2 TIFF file with embedded text

It includes one line of text: "Secret data asdklhfsdifgsdouifh"

Common methods of concealing data in digital images include:
Least significant bit insertion: This is a very popular method because of its simplicity. In this method, the LSB of each byte in the image is used to store the secret data. The resulting changes are too small to be recognized by the human eye. The disadvantage of this technique is that since it uses each pixel in an image, a lossless compression format like .bmp or .tif has to be used for the image. If lossy compression is used, some of the hidden information might be lost.
Masking and filtering: These methods hide information in a manner similar to paper watermarks. This can be done, for example, by modifying the luminance of parts of the image. It does change the visible properties of an image, but if done with care the distortion is barely discernable. This method is much more robust than LSB modification with respect to compression since the information is hidden in the visible parts of the image.
Transformations: This is a more complex way of hiding information in an image. Various algorithms and transformations are applied on the image to make hide information in it. DCT (Direct cosine transformation) is one such method. DCT is used by the JPEG compression algorithm to transform successive 8 x 8 pixel blocks of the image, into 64 DCT coefficients each. VC tools can use the LSB of the quantized DCT coefficient can be used to hide information. In addition to DCT, images can be processed with fast Fourier transformation and wavelet transformation. Other image properties such as luminance can also be manipulated.
VC hides the covert message but not the fact that two parties are communicating with each other. The VC process generally involves placing a hidden message in some transport medium, called the carrier. The secret message is embedded in the carrier to form the VC medium. The use of a VC key may be employed for encryption of the hidden message and/or for randomization in the cryptography scheme. In summary:
VC_medium = hidden_message + carrier (within time constraints).

1.5 Classification of steganographic techniques:

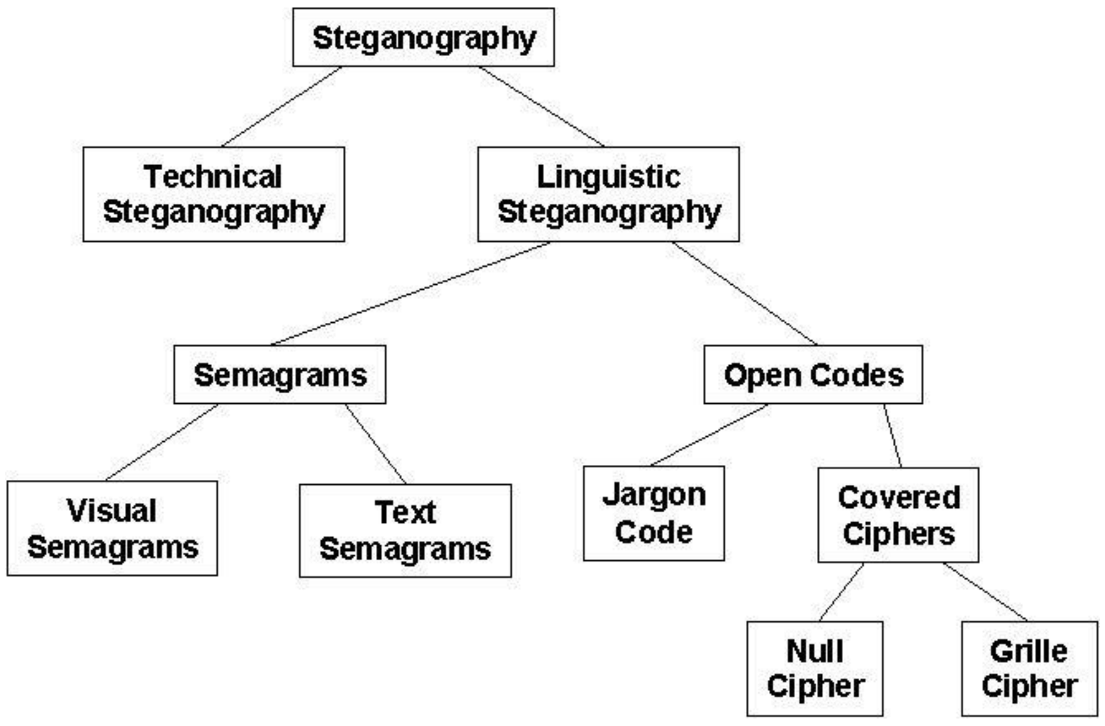


Figure 1.3Classification of Cryptography Techniques

Figure 1.3 shows a common taxonomy of steganographic techniques. (1)
Technical cryptography uses scientific methods to hide a message, such as the use of invisible ink or microdots and other size-reduction methods.
Linguistic cryptography hides the message in the carrier in some non-obvious ways and is further categorized as semagrams or open codes.
Semagrams hide information by the use of symbols or signs. A visual semagram uses innocent-looking or everyday physical objects to convey a message, such as doodles or the positioning of items on a desk or Website. A text semagram hides a message by modifying the appearance of the carrier text, such as changes in font size or type, adding extra spaces, or different flourishes in letters or handwritten text.
Open codes hide a message in a legitimate carrier message in ways that are not obvious to an unsuspecting observer. The carrier message is sometimes called the overt communication whereas the hidden message is the covert communication. This category is subdivided into jargon codes and covered ciphers.
Jargon code, as the name suggests, uses language that is understood by a group of people but is meaningless to others. Jargon codes include *war chalking* (symbols used to indicate the presence and type of wireless network signal, underground terminology, or an innocent conversation that conveys special meaning because of

facts known only to the speakers. A subset of jargon codes is cue codes, where certain prearranged phrases convey meaning.

Covered or concealment ciphers hide a message openly in the carrier medium so that it can be recovered by anyone who knows the secret for how it was concealed. A grille cipher employs a template that is used to cover the carrier message. The words that appear in the openings of the template are the hidden message. A null cipher hides the message according to some prearranged set of rules, such as "read every fifth word" or "look at the third character in every word."

As an increasing amount of data is stored on computers and transmitted over networks, it is not surprising that cryptography has entered the digital age. On computers and networks, cryptography applications allow for someone to hide any type of binary file in any other binary file, although image and audio files are today's most common carriers.

Cryptography provides some very useful and commercially important functions in the digital world, most notably digital watermarking. In this application, an author can embed a hidden message in a file so that ownership of intellectual property can later be asserted and/or to ensure the integrity of the content. An artist, for example, could post original artwork on a Website. If someone else steals the file and claims the work as his or her own, the artist can later prove ownership because only he/she can recover the watermark. Although conceptually similar to cryptography, digital watermarking usually has different technical goals. Generally only a small amount of repetitive information is inserted into the carrier, it is not necessary to hide the watermarking information, and it is useful for the watermark to be able to be removed while maintaining the integrity of the carrier.

Cryptography has a number of nefarious applications; most notably hiding records of illegal activity, financial fraud, industrial espionage, and communication among members of criminal or terrorist organizations.

Null Ciphers

Historically, null ciphers are a way to hide a message in another without the use of a complicated algorithm.

Future Usage:

Cryptography is a powerful tool that enables people to communicate without eavesdroppers even being aware that the communication is taking place. It can even be combined with cryptography so that the message is not just hidden but also scrambled so that even on being discovered it cannot be read. In the future, the most important use of cryptography techniques will probably be in the field of digital watermarking to track the copyright and ownership of electronic media. There has also been growing concern in law enforcement agencies about the use of cryptography for exchange of illegal material via web page images, audio and other files. Steganalysis is one such branch of digital crime detection and prevention.

1.6 Visual Cryptography and Security

As mentioned previously, VC is an effective means of hiding data, thereby protecting the data from unauthorized or unwanted viewing. But crypto is simply one of many ways to protect the confidentiality of data. It is probably best used in conjunction with another data-hiding method. When used in combination, these methods can all be a part of a layered security approach. Some good complementary methods include:
Encryption - Encryption is the process of passing data or plaintext through a series of mathematical operations that generate an alternate form of the original data known as cipher text. The encrypted data can only be read by parties who have been given the necessary key to decrypt the cipher text back into its original plaintext form. Encryption doesn't hide data, but it does make it hard to read!
Covert channels - Some tools can be used to transmit valuable data in seemingly normal network traffic. One such tool is *Loki*. *Loki* is a tool that hides data in ICMP traffic (like ping).

1.7 Protecting Against Malicious cryptography

Unfortunately, all of the methods mentioned above can also be used to hide illicit, unauthorized or unwanted activity. What can you do to prevent or detect issues with stego? There is no easy answer. If someone has decided to hide their data, they will probably be able to do so fairly easily. The only way to detect cryptography is to be actively looking for in specific files, or to get very lucky. Sometimes an actively enforced security policy can provide the answer: this would require the implementation of company-wide acceptable user policies that restrict the installation of unauthorized programs on company computers.

Using the tools that you already have to detect movement and behavior of traffic on your network may also be helpful. Network intrusion detection systems can help administrators to gain an understanding of normal traffic in and around your network and can thus assist in detecting any type of anomaly, especially with any changes in the behavior of increased movement of large images around your network. If the administrator is aware of this sort of anomalous activity, it may warrant further investigation. Host-based intrusion detection systems deployed on computers may also help to identify anomalous storage of image and/or video files.

They are the visual attack (actually seeing the differences in the files that are encoded) and the statistical attack: "The idea of the statistical attack is to compare the frequency distribution of the colors of a potential crypto file with the theoretically expected frequency distribution for a crypto file." It might not be the quickest method of protection, but if you suspect this type of activity, it might be the most effective. For JPEG files specifically, a tool called Crypdetect, which looks for signs of cryptography in JPEG files, can be employed. cryptobreak, a companion tool to crypdetect, works to decrypt possible messages encoded in a suspected cryptographic file, should that be the path you wish to take once the crypto has been detected.

Cryptography (from the Greek for "covered writing") is the secret transmission of a message. It is distinct from encryption, because the goal of encryption is to make a message difficult to read while the goal of cryptography is to make a message altogether invisible. A VC message may also be an encrypted as an extra barrier to interception, but need not be. Cryptography has the advantage that even a talented code-cracker cannot decipher a message without knowing it is there.

Cryptography has been used since ancient times; Greek historian Herodotus records how one plotter of a revolt communicated secretly with another by shaving a slave's head, writing on his scalp, letting his hair grow back, and sending the slave as an apparently unencumbered messenger. The number of ways in which a cryptoographic message might be sent is limited only by human ingenuity. A photograph of a large group of people, for example, might contain a Morse-code message in the expressions of the people in the photograph (e.g., smiling for dot, blank for dash) or in the directions they are looking (e.g., slightly to the left for dot, straight at the camera for dash). Writing in invisible ink or miniaturizing a message, as on microfilm, are also forms of cryptography. Probably the commonest form of cryptography involves the embedding of messages in apparently innocent texts, with the letters of message indicated either by subtle graphic emphasis (e.g., heavier ink, lighter ink, a small defect) or by special positioning. For instance, reading first word of every sentence in what appears to be an ordinary letter might yield a cryptographic message.

Like most other forms of cryptography and secret writing, cryptography has thrived in the digital era. Most digital documents contain useless or insignificant areas of data, or involve enough redundancy that some of their information can be altered without obvious effect. For instance, one might conceal a message bit stream inside a digital audio file by replacing the least-significant bit of every waveform sample (or every n^{th} waveform sample) with a message bit; the only effect on the file, if played back as audio, would be a slight decrease in the sound quality (probably imperceptible). Although cryptographic messages can be hidden in any kind of digital files, image files, because they contain so much data to begin with, are usually used for digital cryptography. Today a number of commercial or shareware programs exist for encoding text into cryptographic images ("crypto-images"), and are used by millions of people worldwide who wish to evade surveillance, especially by governments. This includes people who have reason to fear punishment for expressing their political ideas, as well as terrorists.

1.8 Points to remember

When embedding data, one must remember the following important points and restrictions:

- The cover data should not be significantly degraded by the embedded data.
- The embedded data should be directly encoded into the media, rather than into a header or wrapper, to maintain data consistency across formats.
- The embedded data should be as immune as possible to modifications from intelligent attacks or anticipated manipulations such as filtering.

1.9 Requirement Analysis

Software Requirement: MATLAB R2015a

Resources required for the LSB technique:

- For text insertion:

✓ No. of base image pixels = $(N_c * 8) + 1$

N_c is the no. of bits in the string

- For image insertion:

✓ No. of base image pixels = $[(N_c * C_w * 8)] + 32$

N_c = No. of pixels in the image to be watermarked

C_w = No. of components of the image *(e.g. 3 in case of RGB)*

In next chapter we will review the literature related to visual cryptography and formulate our problem statement. In the following chapters we will see the implementation of the project. Finally we will summarize the results obtained and conclude.

2. REVIEW OF LITERATURE AND PROBLEM FORMULATION

With the rapid advancement of network technology, multimedia information is transmitted over the Internet conveniently. Various confidential data such as military maps and commercial identifications are transmitted over the Internet. While using secret images, security issues should be taken into consideration because hackers may utilize weak link over communication network to steal information that they want. To deal with the security problems of secret images, various image secret sharing schemes have been developed. It is now common to transfer multimedia data via the Internet. With the coming era of electronic commerce, there is an urgent need to solve the problem of ensuring information safety in today‘s increasingly open network environment.

A. Black And White VC Schemes

Naor and Shamir‘s proposed encoding scheme to share a binary image into two shares Share1 and Share2. If pixel is white one of the above two rows of Fig. 2.1 is chosen to generate Share1 and Share2. Similarly if pixel is black one of the below two rows of Fig.2.1 is chosen to generate Share1 and Share2. Here each share pixel p is encoded into two white and two black pixels each share alone gives no clue about the pixel p whether it is white or black. Secret image is shown only when both shares are superimposed. Stacking shares represents OR operation to human visual system. OR operation is lossy recovery. If XOR operation is applied instead of OR then we can get lossless restore of the original image. But, XOR operation requires computation. The physical stacking process can only simulate the OR operation. The drawbacks of this scheme are:

- It is for black and white images.
- Need more storage capacity as shares are four times the original image.
- It is time consuming as single pixel encoding at each run.

Many advanced schemes were introduced when a colored image is encrypted. A multi-pixel non-expanded scheme for color images introduced which can encode more than one pixel for each run resulting same size of shares as secret image. The scheme achieves high efficiency for encoding and this works well for general access structure for chromatic images without pixel expansion but it generates meaningless shares.

The disadvantage of the above schemes is that only one set of confidential messages can be embedded, so to share large amounts of confidential messages several shares have to be generated.

B. Color Visual Cryptography Schemes

Sharing a secret color image and also to generate the meaningful share to transmit secret color image Chang and Tsai anticipated color visual cryptography scheme. For a secret color image two significant color images are selected as cover images which are the same size as the secret color image. Then according to a predefined Color Index Table, the secret color image will be hidden into two camouflage images. One disadvantage of this scheme is that extra space is required to accumulate the Color Index Table. In this scheme also number of sub-pixels is in proportional to the number of colors in the secret image as in Verheul and Van Tilborg Yang and Laih schemes. When more colors are there in the secret image the larger the size of shares will become.

To overcome this limitation Chin-Chen Chang et al developed a secret color image sharing scheme based on modified visual cryptography. This scheme provides a more efficient way to hide a gray image in different shares. In this scheme size of the shares is fixed; it does not vary when the number of colors appearing in the secret image differs. Scheme does not require any predefined Color Index Table. Though pixel expansion is a fixed in this scheme is not suitable for true color secret image. To share true-color image Lukac and Plataniotis introduced bit-level based scheme by operating directly on S-bit planes of a secret image.

2.1 Covert communication techniques

Cryptography	**Steganography**
Protection Of message	Message Hiding Cover unimportant
Secure Communication	Hidden communication
Data encrypted, Not embedded	Possibly large Messages Embedded
Goal: Unbreakable	Goal: Undetectability

Table 2.1Comparison of covert communication techniques

The general block diagram for cryptography can be as shown below:

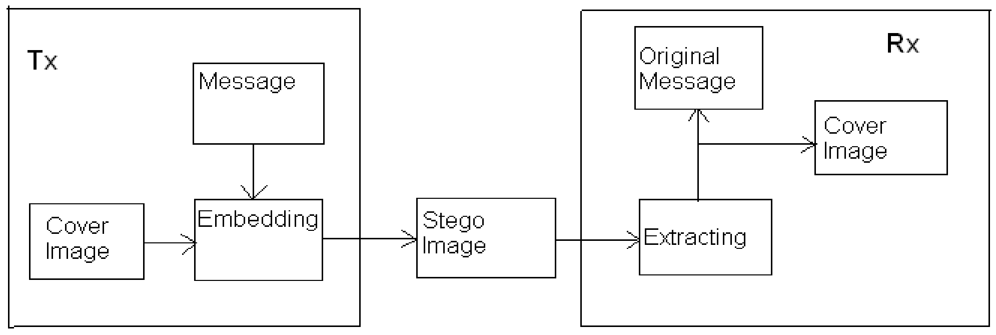


Figure 2.1General block diagram for cryptography

2.2 ALGORITHMS USED

Following algorithms have been used to implement cryptography
1. LSB algorithm
2. DWT based algorithm
3. DCT based algorithm

2.2.1 LSB algorithm

In LSB algorithm, data to be embedded is in the LSB of the image pixel.

For example: to insert 'A' character in following pixels (Each pixel is of 8 bits)

Before Embedding:

```
00100111   11101001   11001000   00100111
11101001   11001000   00100111   11101001
```

After Embedding:

```
00100111   11101000   11001000   00100110
11101000   11001000   00100111   11101001
```

Since information is embedded in LSB, not much change occurs in the image.

2.2.2 DISCRETE WAVELET TRANSFORM (DWT) Technique

DWT is being increasingly used for cryptography. Decomposition and reconstruction of images is achieved through DWT. In DWT, image is iteratively decomposed into high and low frequency bands implying sub-band coding. Since DWT gives excellent spatial and frequency localization properties it is easy to identify the image areas where a disturbance can be easily concealed.

A DWT based non-oblivious image embedding method is proposed where a binary image is embedded into high frequency sub-band of a gray scale image. The HH band has a negligible percentage of the total image energy, so changing the coefficients of HH band results in a stego image with excellent perceptual quality.

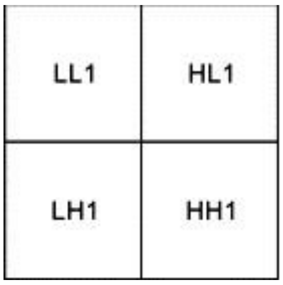


Figure 2.2Wavelet Decomposition into one level

The forward wavelet-based transform uses a 1-D sub-band decomposition process where a 1-D set of samples is converted into the low-pass sub-band (Li) and high-pass sub-band (Hi).The low-pass sub-band represents a down sampled low-resolution version of the original image. The high-pass sub-band represents residual information of the original image, needed for the perfect reconstruction of the original image from the low-pass sub-band. The 2-D sub-band decomposition is just an extension of 1-D sub-band decomposition. The entire process is carried out by executing 1-D sub-band decomposition twice, first in one direction (horizontal), then in the orthogonal (vertical) direction. For example, the low-pass sub-bands (Li) resulting from the horizontal direction is further decomposed in the vertical direction, leading to LLi and LHi sub-bands. Similarly, the high pass sub-band (Hi) is further decomposed into HLi and HHi. After one level of transform, the image can be further decomposed by applying the 2-D sub-band decomposition to the existing Lli sub-band. This iterative process results in multiple "transform levels". For example, in Fig. 2.5(a), the first level of transform results in LH1, HL1, and HH1, in addition to LL1, which is further decomposed into LH2, HL2, HH2, LL2 at the second level, and the information of LL2 is used for the third level transform. We refer to the sub-band LLi as a low-resolution sub-band and high-pass sub-bands LHi, HLi, HHi as horizontal, vertical, and diagonal sub-band respectively since they represent the horizontal, vertical, and diagonal residual information of the original image. An example of three-level decomposition into sub-bands of the image shown below.

Figure 2.3 Wavelet decomposition into three levels

Advantage: The DWT based cryptographic technique exhibits high perceptual transparency of the stego-image. It is also robust to common image processing attacks like intensity adjustment and image cropping.

2.2.3 DCT algorithm

Concept: The discrete cosine transform separates an image into spectral sub-bands of differing importance with respect to the image visual quality. DCT, like DFT transforms a signal from spatial domain to the frequency domain.

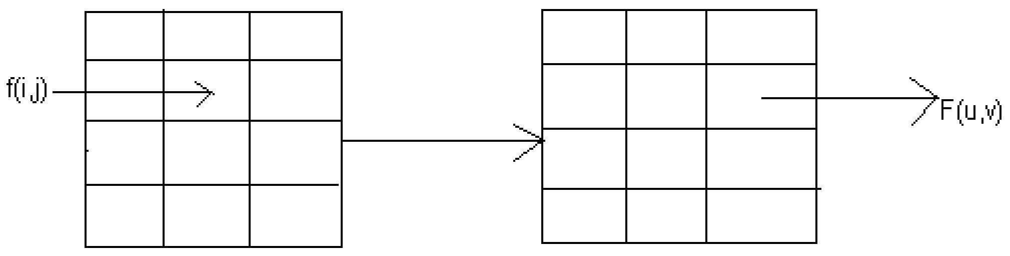


Figure 2.4Spatial domain to frequency domain transformation

Each DCT co-efficient F(u,v) of an 8x8 block of image pixels f(x,y) is given by:

$F(u,v)=C(u)C(v)[\Sigma\Sigma f(x,y)^{*} \cos\underline{^{(2x+1)}u\pi\cos^{(2y+1)}v\pi)}]$

1616

Selection of coefficients for embedding data:
Since many image processing techniques affect the perceptually insignificant regions of the image, data should not be embedded in these regions. For example, the information placed in the high frequency spectrum of an image can be easily eliminated with little degradation to the image by any process that performs low pass filtering.

We can formulate the whole process of hiding and retrieving data in steps as shown below

1. Read data to be hidden
2. Read the image in which data is to be hidden
3. Apply digital transformation methods on the image
4. Hide the data in the image
5. Retrieve the original data

Here the hiding and retrieving of images are the two main processes to be done.
These two processes can be described as

Hiding process

Input to the hiding process contains data image to be hidden and a cover image, which hides the data image. Set up these images for hiding. The input cover image is tested to know whether the given image can be used to hide data or not. If the image is invalid then the process exits. If the image is valid then hiding process takes place. This results in stego image containing hidden information.

Seeking process

Input to seeking process is stego image containing hidden information obtained during hiding process. Setup these files for seeking. Now seek the steno image to retrieve data image hidden in it. As a result we get back the original data image that was hidden.

2.3 Detecting Cryptography

Cryptanalysis, the detection of cryptography by a third party, is a relatively young research discipline with few articles appearing before the late-1990s. The art and science of cryptanalysis is intended to detect or estimate hidden information based on observing some data transfer and making no assumptions about the cryptography algorithm. Detection of hidden data may not be sufficient. The cryptanalyst may also want to extract the hidden message, disable the hidden message so that the recipient cannot extract it, and/or alter the hidden message to send misinformation to the

recipient. Cryptography detection and extraction is generally sufficient if the purpose is evidence gathering related to a past crime, although destruction and/or alteration of the hidden information might also be legitimate law enforcement goals during an on-going investigation of criminal or terrorist groups.
Cryptanalysis techniques can be classified in a similar way as cryptanalysis methods, largely based on how much prior information is known.

- Cryptography-only attack: The cryptography medium is the only item available for analysis.
- Known-carrier attack: The carrier and media both are available for analysis.
- Known-message attack: The hidden message is known.
- Chosen-cryptography attack: The cryptography medium and algorithm are both known.
- Chosen-message attack: A known message and cryptography algorithm are used to create cryptography media for future analysis and comparison.
- Known-cryptography attack: The carrier and cryptography medium, as well as the cryptography algorithm, are known.

Cryptography methods for digital media can be broadly classified as operating in the image domain or transform domain. Image domain tools hide the message in the carrier by some sort of bit-by-bit manipulation, such as least significant bit insertion. Transform domain tools manipulate the cryptography algorithm and the actual transformations employed in hiding the information, such as the discrete cosine transform coefficients in JPEG images.
It follows, then, that cryptanalysis broadly follows the way in which the cryptography algorithm works. One simple approach is to visually inspect the carrier and cryptography media. Many simple cryptography tools work in the image domain and choose message bits in the carrier independently of the content of the carrier. Although it is easier to hide the message in the area of brighter color or louder sound, the program may not seek those areas out. Thus, visual inspection may be sufficient to cast suspicion on a cryptography medium.
A second approach is to look for structural oddities that suggest manipulation. Least significant bit insertion in a palette-based image often causes a large number of duplicate colors, where identical (or nearly identical) colors appear twice in the palette and differ only in the least significant bit. Cryptography programs hide information merely by manipulating the order of colors in the palette cause structural changes, as well. The structural changes often create a signature of the cryptography algorithm that was employed.
Cryptographic techniques generally alter the statistics of the carrier and, obviously, longer hidden messages will alter the carrier more than shorter ones.
Statistical analysis is commonly employed to detect hidden messages, particularly when the analyst is working in the blind there is a large body of work in the area of statistical cryptanalysis.

Statistical cryptanalysis is made harder because some cryptography algorithms take pains to preserve the carrier file's first-order statistics to avoid just this type of detection. Encrypting the hidden message also makes detection harder because encrypted data generally has a high degree of randomness, and ones and zeros appear with equal likelihood.
Recovery of the hidden message adds another layer of complexity compared to merely detecting the presence of a hidden message. Recovering the message requires knowledge or an estimate of the message length and, possibly, an encryption key and knowledge of the crypto algorithm.

Most cryptanalysis today is signature-based, similar to antivirus and intrusion detection systems. Anomaly-based cryptanalysis systems are just beginning to emerge. Although the former systems are accurate and robust, the latter will be more flexible and better able to quickly respond to new cryptography techniques. One form of so-called "blind cryptography detection" distinguishes between clean and cryptography images using statistics based on wavelet decomposition.
Steganalysis helps find embedded cryptography but also shows writers of new cryptography algorithms how to avoid detection.

Cryptanalysis
Cryptanalysis is the art and science of breaking the security of cryptographic systems. David Kahn has defined security as techniques of protecting information. Intelligence is defined as methods of retrieving hidden information. As the goal of cryptography is to conceal the existence of a secret message, a successful attack on cryptographic system consists of detecting that a certain file contains embedded data and render it useless. In fact cryptanalysis is as much use as is cryptography.
Cryptanalysis involves attacking cryptographic systems by detection, destruction, extraction or modification of embedded data. To design and develop a superior and more robust system it is necessary to understand the means by which an attacker can defeat a crypto system.

2.4 Applications of visual cryptography

There are many applications for digital cryptography of images , including copyright protection ,secret communication.

Copyright protection:

A secret copyright notice or watermark can be embedded inside an image to identify it as intellectual property. When an image is sold or distributed an identification of the recipient and time stamp can be embedded to identify potential pirates. A watermark can also serve to detect whether the image has been subsequently modified. Detection of an embedded watermark is performed by a statistical, correlation, or similarity test, or by measuring other quantity characteristic to watermark, in a stego image.

Feature tagging:

Captions, time stamps and other descriptive elements can be embedded inside an image. Copying a crypto image also copies all of embedded features and only parties who possess decoding crypto-key will be able to extract and view features. In an image databases, keywords can be embedded to facilitate search engines.

Secret communication:

In many situations, transmitting a cryptographic message draws unwanted attention. The use cryptographic technology may be restricted or forbidden by law. However the use of cryptography does not advertise covert communication and therefore avoids scrutiny of the sender, message and recipient's trade secret , blueprint ,or other sensitive information can be transmitted without altering potential attackers.

2.5 Limitations of algorithms used above:

1. LSB algorithm: fragile, since it depends on the LSB's there are bits that are most likely to be modified by a lossy compression algorithm, such as, JPEG. As a result embedded data would be probably lost. Any filtering process would also change the value of LSB's.

2. DWT algorithm: In DWT algorithm, experimental results shows that recovery technique can serve image enhancement but is not robust against JPEG compression, Cropping, and noise addition.

3. DCT algorithm: In DCT algorithm though the images extracted under the influence of various attacks are noisy, retrieval accuracy of the extracted image is very good.

3. VC USING LEAST BIT SUBSTITUTION TECHNIQUE

Least Bit Substitution technique is employed to embed the watermark text/image within the carrier image. The user is provided with a graphical user interface to read and process the carrier image, watermark as well as the security key and to store the crypto-image on disk.

3.1 Insertion Scheme

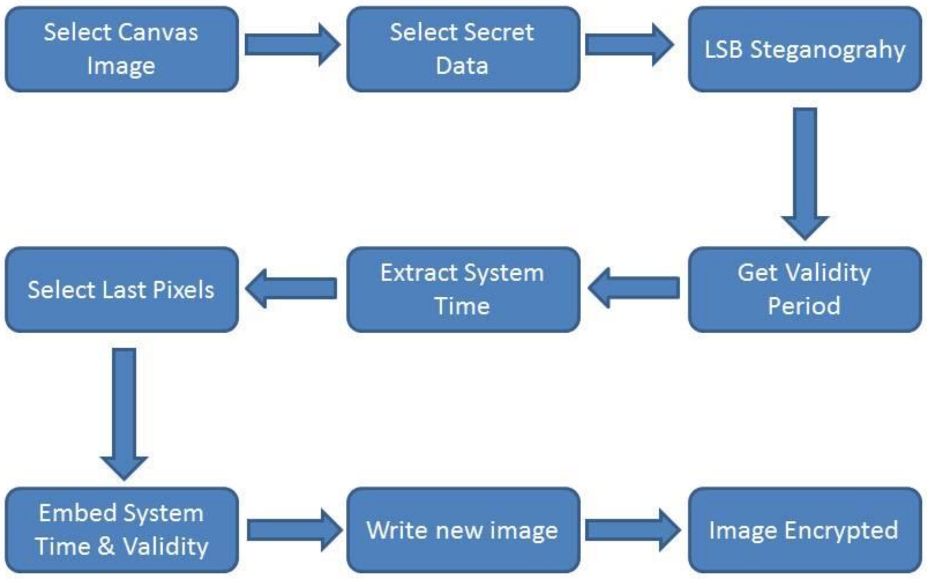


Figure 3.1 Encryption Scheme

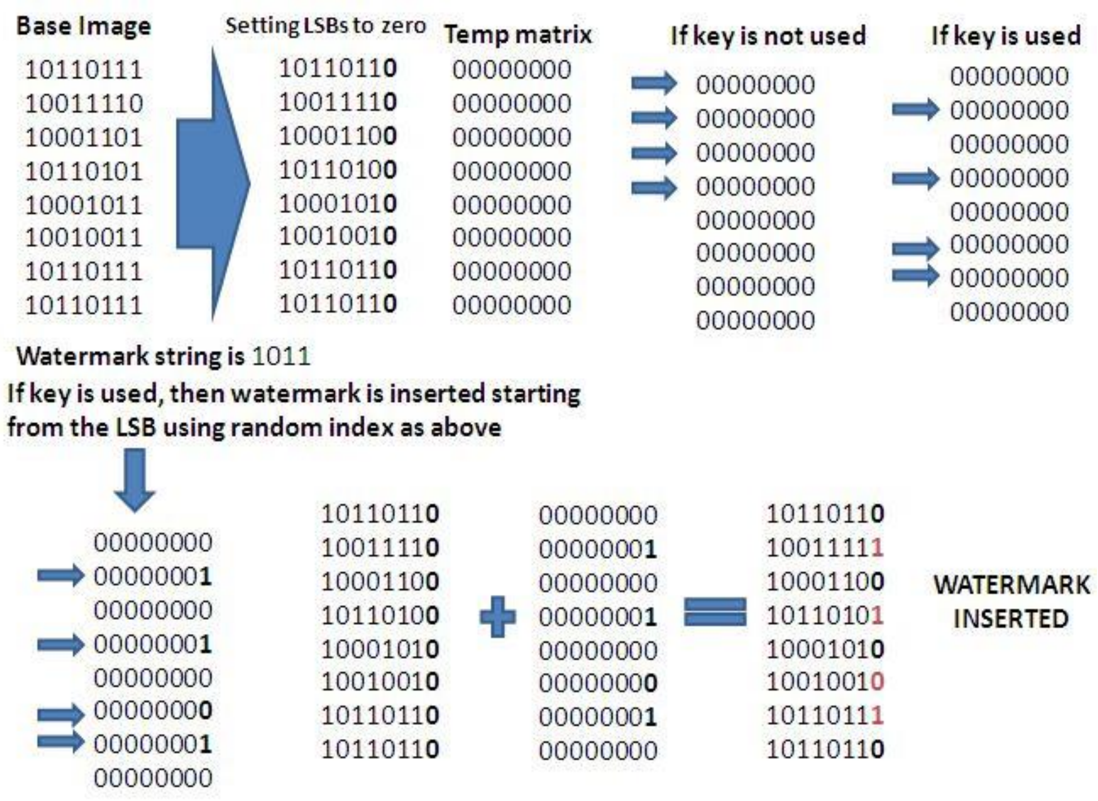


Figure 3.2 Example of the encryption scheme

3.2 Extraction Scheme

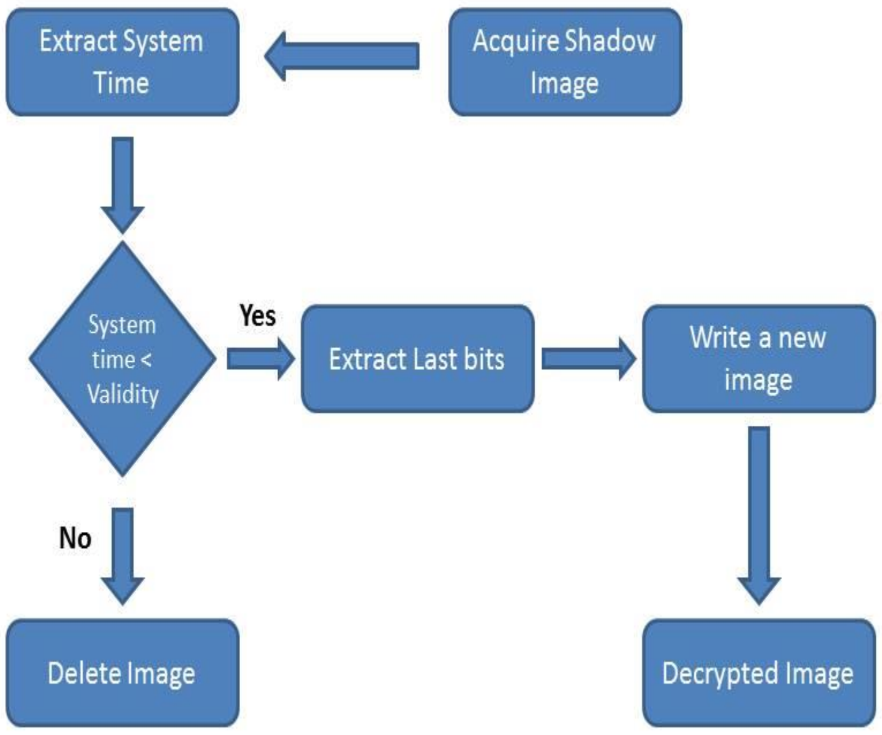


Figure 3.3 Decryption Scheme

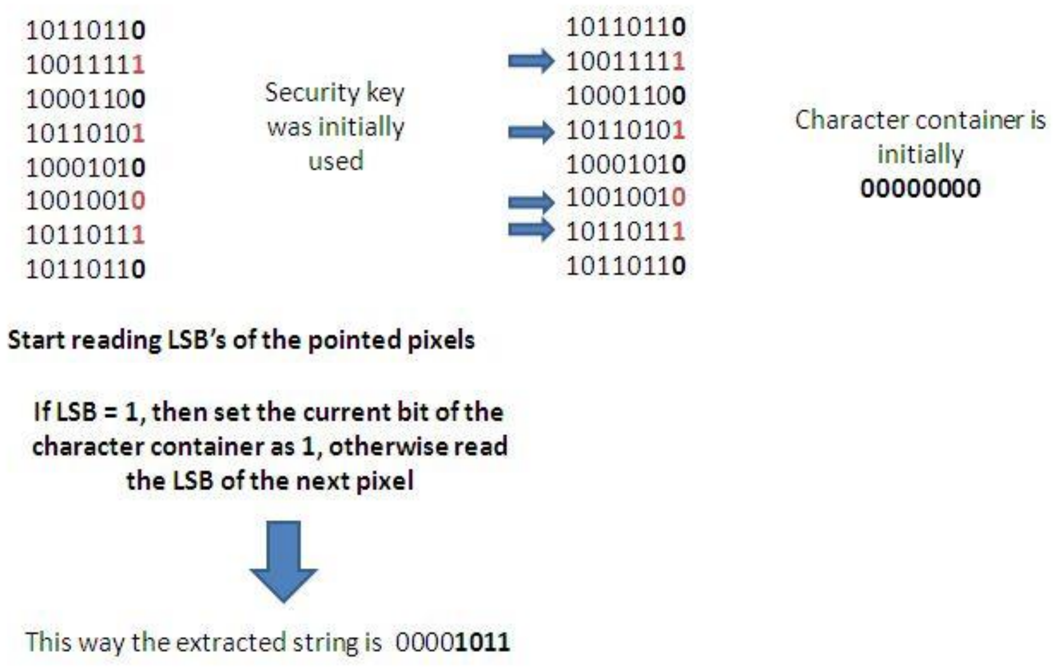

10110110
10011111
10001100
10110101
10001010
10010010
10110111
10110110
Security key was initially used
10110110
10011111
10001100
10110101
10001010
10010010
10110111
10110110
Character container is initially 00000000
Start reading LSB's of the pointed pixels
If LSB = 1, then set the current bit of the character container as 1, otherwise read the LSB of the next pixel
This way the extracted string is 00001011

Figure 3.4 Example of the Decryption scheme

.

3.3 Technology Used

MATLAB is a numerical computing environment and programming language. Maintained by MathWorks, MATLAB allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages. Although it is numeric only, an optional toolbox uses the MuPAD symbolic engine, allowing access to computer algebra capabilities. An additional package, Simulink, adds graphical multidomain simulation and Model-Based Design for dynamic and embedded systems.

MATLAB is very similar to most other platforms, which allow GUI development. Any GUI in MATLAB is in essence a collection of objects. Every object in the GUI has a unique handle (name). For instance, let's consider the following GUI. It is made up of three distinct objects, which are the following:

The *frame or (figure)* which is labeled '*Untitled*'. The second object is the *edit box*. And the third object is the '*Button*' which is labeled '*Push Button*'.

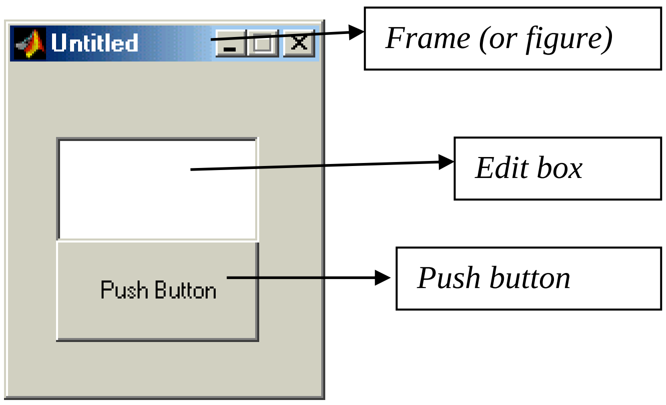
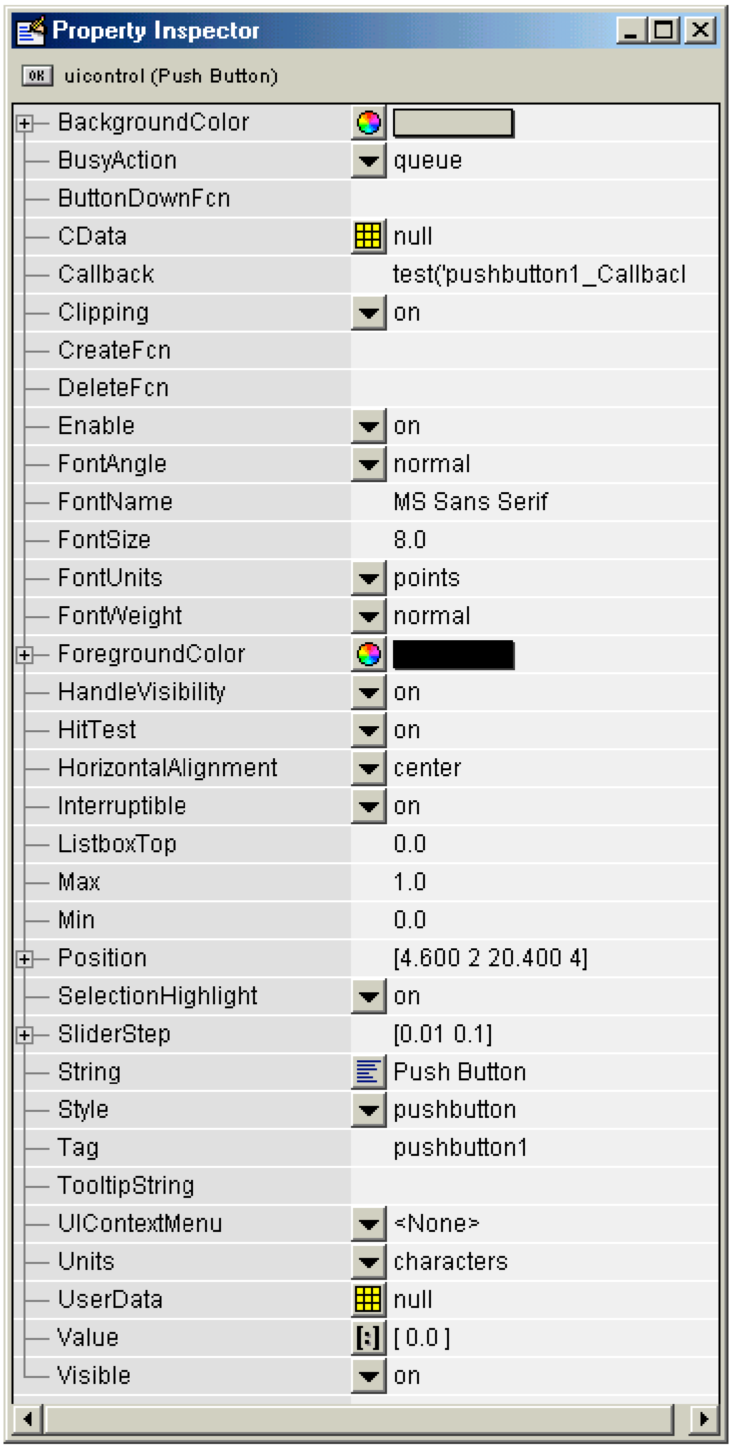


Figure 3.5 GUI figure

- ✓ GUI's are a collection of objects.
- ✓ Every object has a unique handle.
- ✓ Every object has properties.
- ✓ The user can alter these properties at design time.

Everything about the object is listed in the *property inspector* (see figure below) except for one very important property. This property is known as the **'*Handle*'**. The reason this property is not on the list is because the '*Handle*' is assigned at Runtime. The user does not get to specify the handle however the usermay obtain the handle in a number of different ways depending on the version of MATLAB being used.



BackgroundColor	▭
BusyAction	queue
ButtonDownFcn	
CData	null
Callback	test('pushbutton1_Callbacl
Clipping	on
CreateFcn	
DeleteFcn	
Enable	on
FontAngle	normal
FontName	MS Sans Serif
FontSize	8.0
FontUnits	points
FontWeight	normal
ForegroundColor	■
HandleVisibility	on
HitTest	on
HorizontalAlignment	center
Interruptible	on
ListboxTop	0.0
Max	1.0
Min	0.0
Position	[4.600 2 20.400 4]
SelectionHighlight	on
SliderStep	[0.01 0.1]
String	Push Button
Style	pushbutton
Tag	pushbutton1
TooltipString	
UIContextMenu	<None>
Units	characters
UserData	null
Value	[0.0]
Visible	on

Figure 3.6 Property Inspector

4. PROJECT IMPLEMENTATION

4.1 Encryption Interface

On Opening Encryption interface: the following interface is presented to the user.
1. Select the source (carrier) image.
2. Choose secret data.
3. Enter validity period.
4. Press Encrypt button.
5. TIFF image is stored on the desktop.

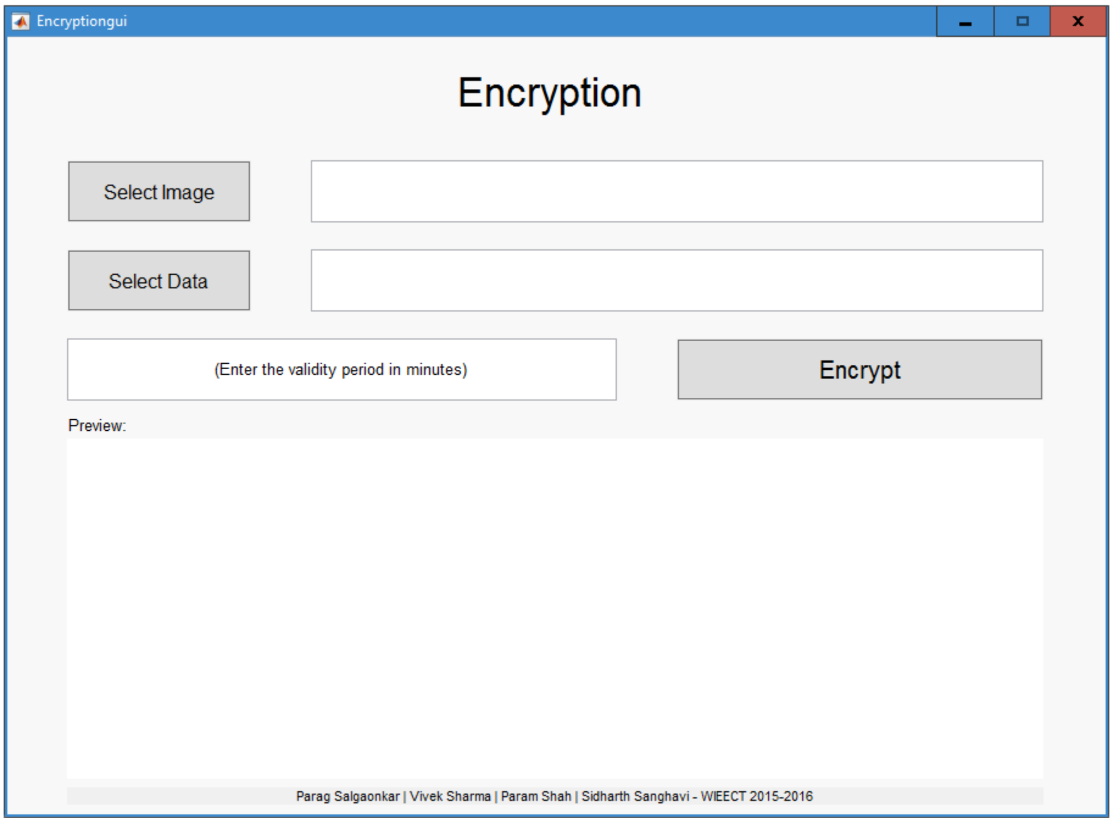
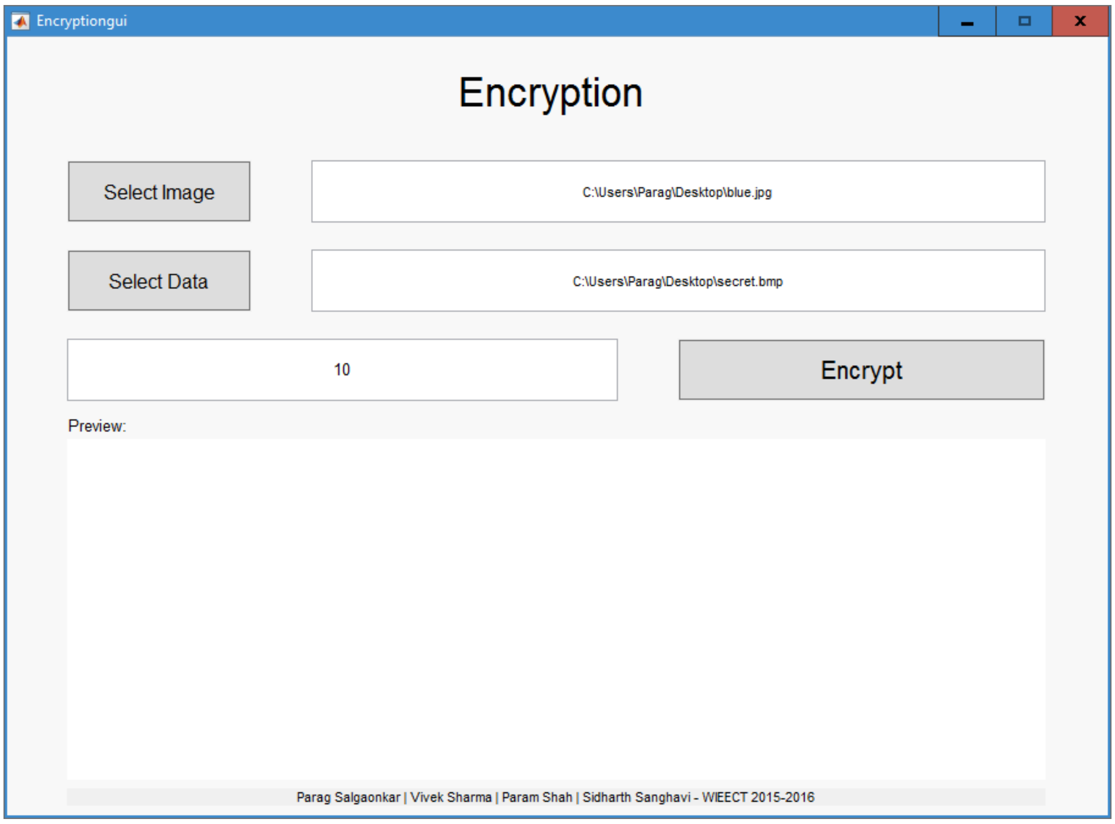

Encryptiongui
Encryption
Select Image
Select Data
(Enter the validity period in minutes)
Encrypt
Preview:
Parag Salgaonkar | Vivek Sharma | Param Shah | Sidharth Sanghavi - WIEECT 2015-2016

Figure 4.1 Encryption interface



C:\Users\Parag\Desktop\blue.jpg

C:\Users\Parag\Desktop\secret.bmp

10

Parag Salgaonkar | Vivek Sharma | Param Shah | Sidharth Sanghavi - WIEECT 2015-2016

Figure 4.2 Encryption example

For instance, we have chosen image carrier image by clicking the select image button. The source image (*blue.jpg*), output image filename (*newimage*) and format (.*tif*) are being specified. The image file to be encyrpted is specified (secret.*tif*). The validity period (*10*) is being specified. The Encrypt pushbutton is clicked

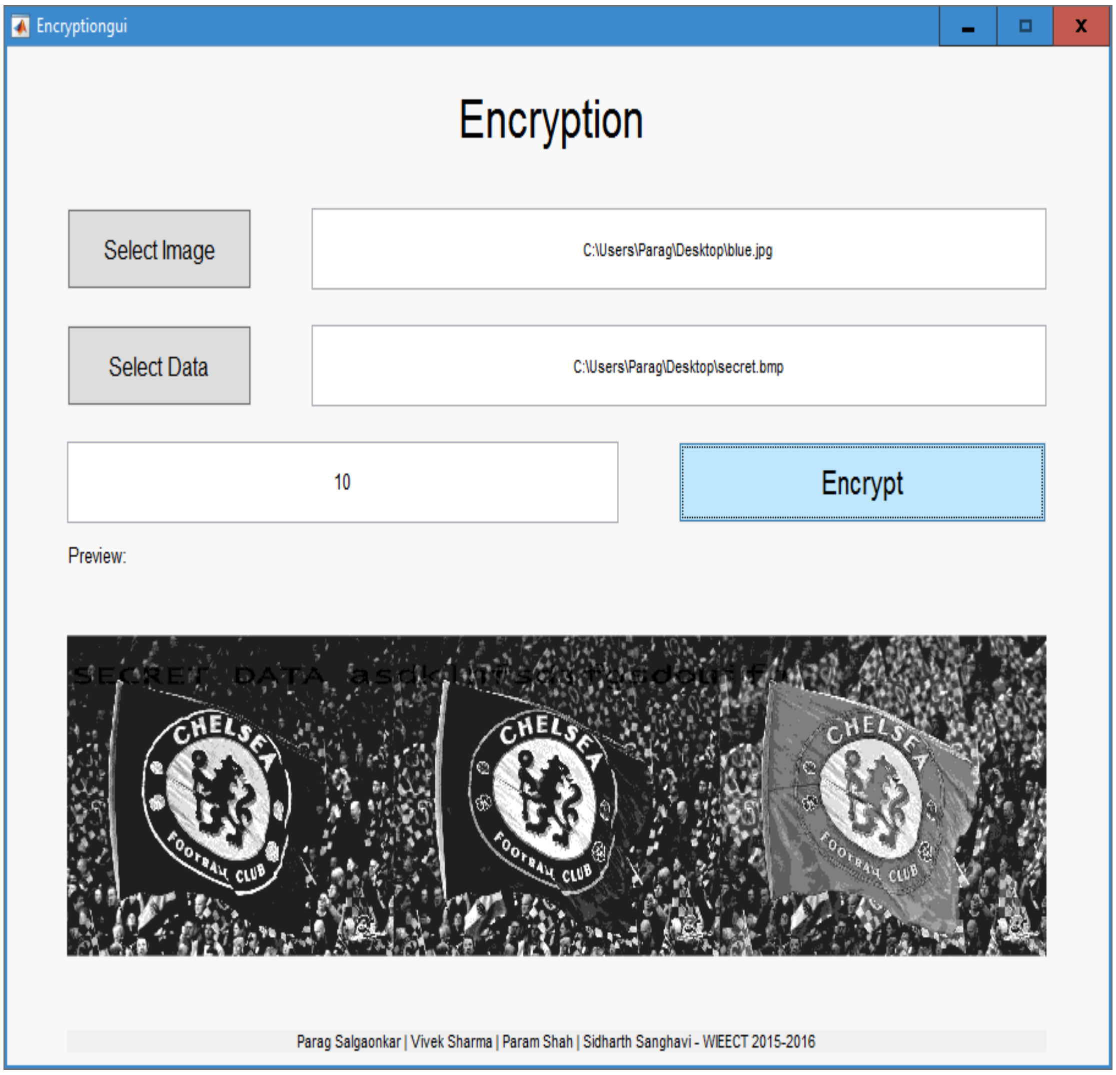

Encryptiongui
Encryption
Select Image
C:\Users\Parag\Desktop\blue.jpg
Select Data
C:\Users\Parag\Desktop\secret.bmp
10
Encrypt
Preview:
Parag Salgaonkar | Vivek Sharma | Param Shah | Sidharth Sanghavi - WIEECT 2015-2016

Figure 4.3 Encryption output

The Encrypted image is then displayed in the GUI.
The output image is also stored on desktop.

4.2 Decryption Interface

On Opening Decryption interface: the following interface is presented to the user.
1. Select the received image..
2. Press the dencrypt button.
3. Decrypted TIFF image is stored on the desktop.
4. If validity period expires, received image is deleted.

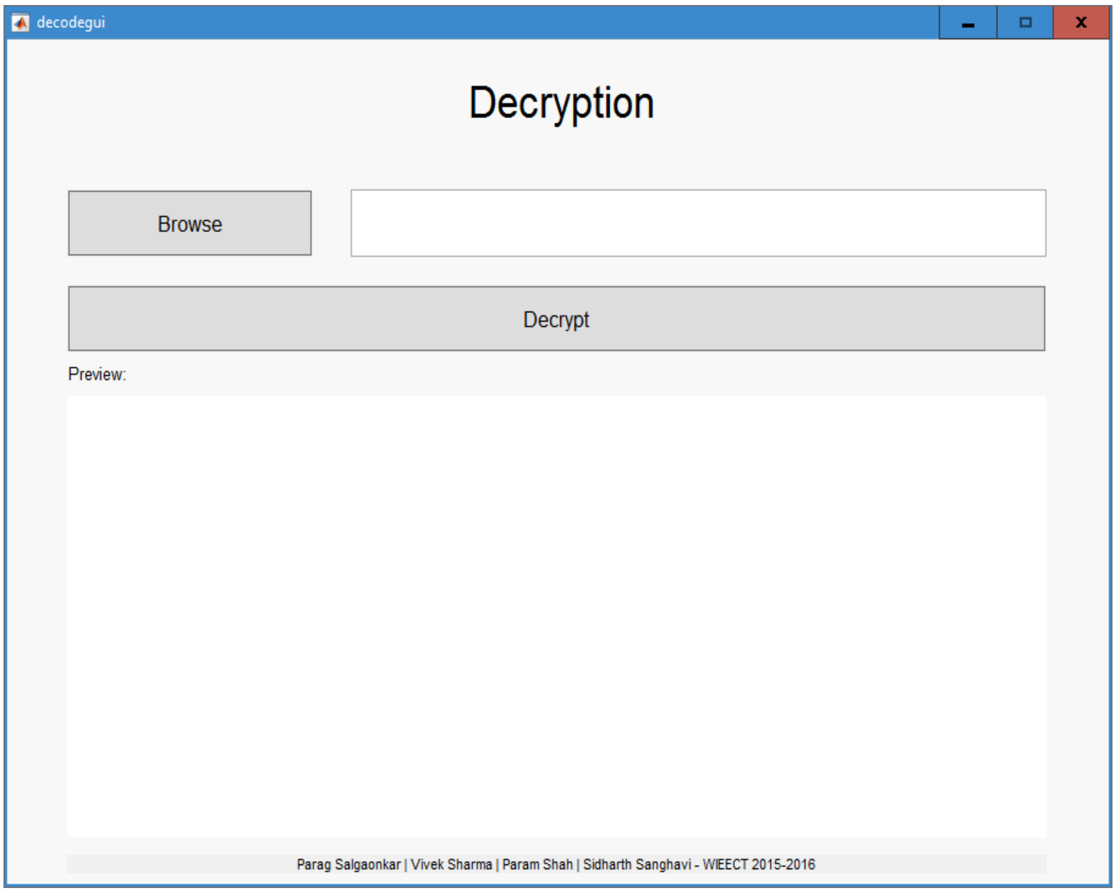

decodegui
Decryption
Browse
Decrypt
Preview:
Parag Salgaonkar | Vivek Sharma | Param Shah | Sidharth Sanghavi - WIEECT 2015-2016

Figure 4.4 Decryption interface

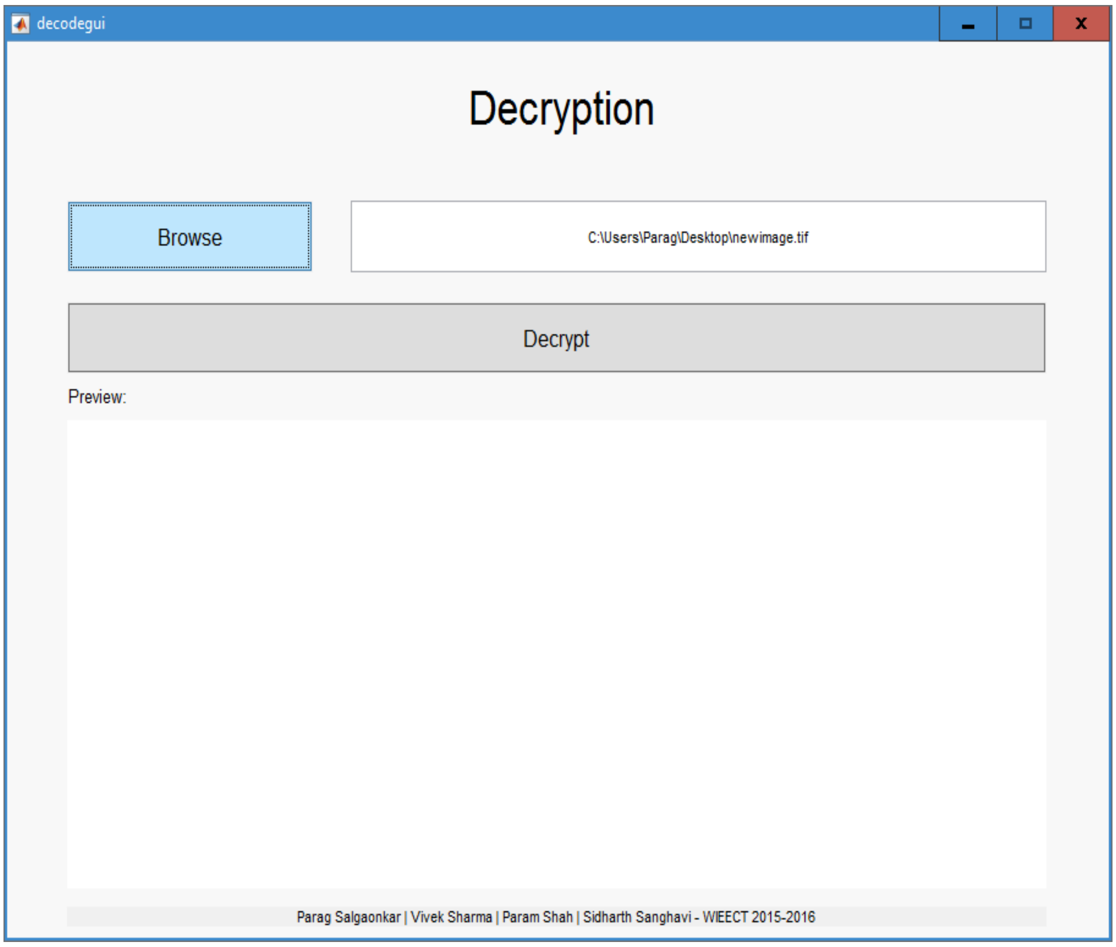
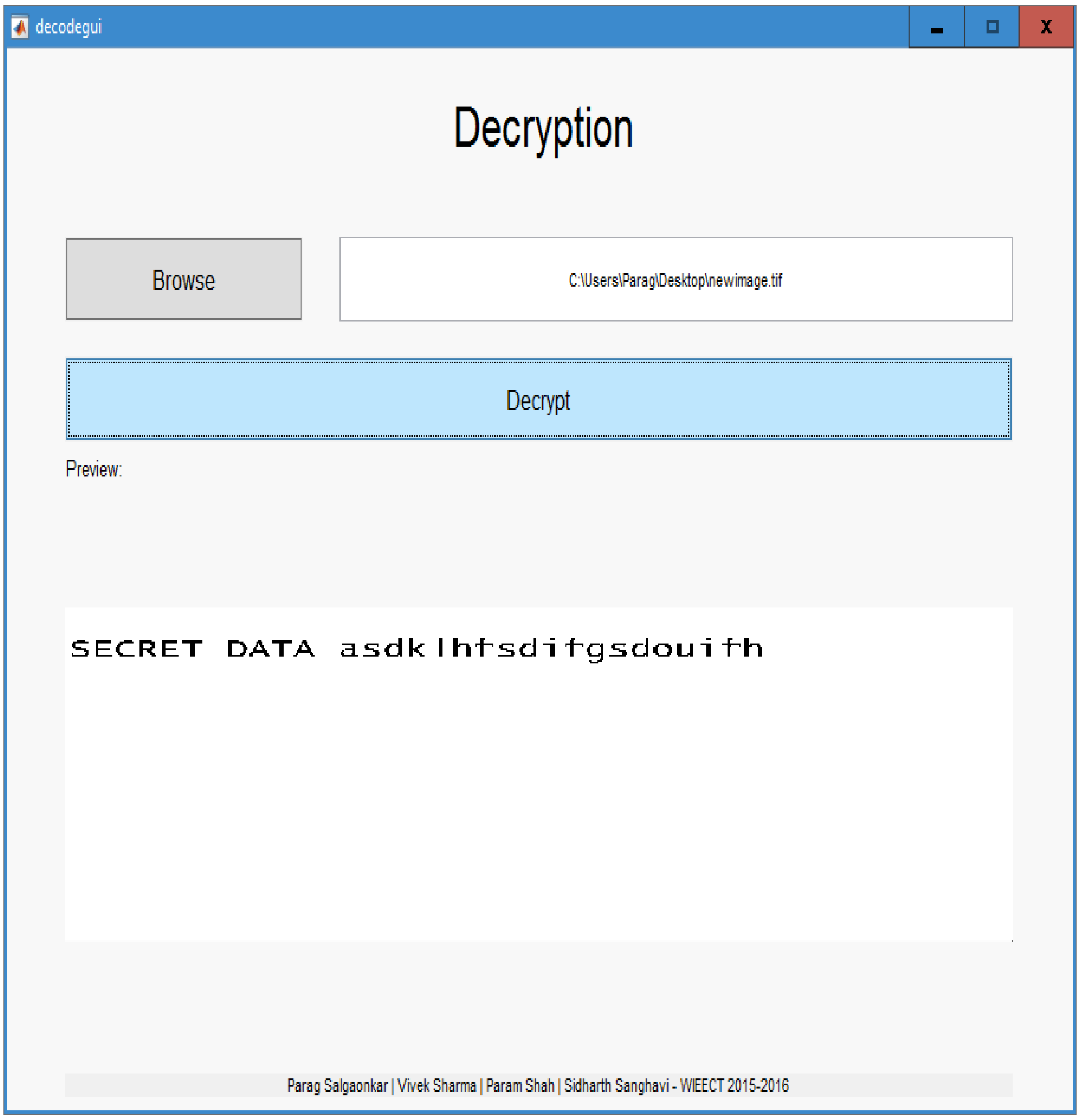

decodegui
Decryption
Browse
C:\Users\Parag\Desktop\newimage.tif
Decrypt
Preview:
Parag Salgaonkar | Vivek Sharma | Param Shah | Sidharth Sanghavi - WIEECT 2015-2016

Figure 4.5 Decryption example

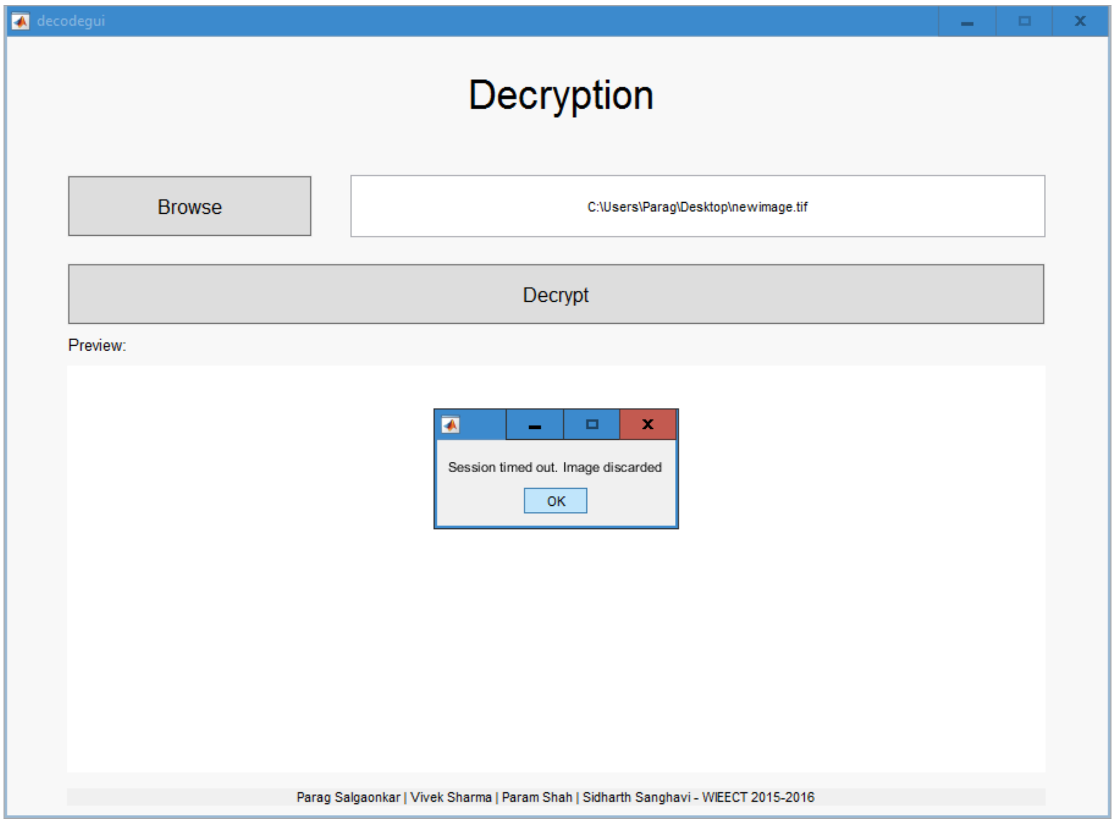


C:\Users\Parag\Desktop\newimage.tif

```
SECRET DATA asdklhfsdifgsdouifh
```

Parag Salgaonkar | Vivek Sharma | Param Shah | Sidharth Sanghavi - WIEECT 2015-2016

Figure 4.6 Decryption output



C:\Users\Parag\Desktop\newimage.tif

Session timed out. Image discarded

Parag Salgaonkar | Vivek Sharma | Param Shah | Sidharth Sanghavi - WIEECT 2015-2016

Figure 4.7 Decryption output (Session timed out)

SUMMARY AND CONCLUSION

Summary

The main objective was to develop an algorithm to embed the secret data into the carrier by means of time validity such that only the user within the valid time is authorized to extract secret data from the given image.

Conclusion

This system is developed as a standalone application in MATLAB. It is implemented on MATLAB R2015a version and the experimental result shows this system fulfill the aim and objective of the project. This system uses Colour Image Visual Cryptography and time constraint for password protection. This system will be a boon for the Bank customers and they can easily send sensitive information over any unprotected network. Once this system is deployed in java, all the computers in a network can be able to access this application through a browser without any software installation on their computer.

Future work

- Since it is a standalone application, the encryption and decryption time depends on the system configuration. So it is viable to shift the whole architecture over a cloud network with professional-level configuration, which could support multiple users over a network
- Since it is a standalone application, when more number of user are access the system the response time will be increased, to avoid this delay we can enhance this system to cloud environment, where based on the number of user access, cloud configuration will change and minimize the response time.
- Also we can add some more security features like Digital Signature in high level transaction process. Visual cryptography technique is used to make the data secure. Here the original data is embedded with time and date of the system which are sent through different communication channels from sender to receiver. Therefore the intruder has less chance to get the actual system time. But still it is not so secured, so it can be made more secure by introducing a symmetric key for both encryption and decryption process.

References

[1] H. Wang and S. Wang, ―Cyber warfare Steganography vs. Steganalysis,‖Commun. ACM, vol. 47, no. 10, pp. 76-82, 2004.

[2] M. Kutter and S. Winkler, ―A vision-based masking model for spread- spectrum image watermarking,‖ IEEE Trans. Image Processing, vol. II, pp. 16-25, Jan. 2002.

[3] C. K. Chan and L. M. Cheng, ―Hiding data in images by simple LSB substitution,‖ Pattern Recognition, pp. 469—474, Mar. 2004.

[4] M. Shirali-Shahreza, ―Steganography in MMS,‖ in Multitopic Conference, 2007. INMIC 2007. IEEE International, 2007, pp. 1-4.

[5] F. Liu1, C.K. Wu1, X.J. Lin, Colour visual cryptography schemes, IET Information Security, July 2008 .

[6] M. Naor and A. Shamir, ―Visual cryptography,‖ Advances in Cryptology-Eurocrypt‘94, pp. 1–12, 1995.

[7] Li Bai , A Reliable (k,n) Image Secret Sharing Scheme by, IEEE,2006.

[8] John F Koegel Buford, Multimedia Systems, Addison Wesley, 2000.

[9] S. J. Shyu, S. Y. Huanga,Y. K. Lee, R. Z. Wang, and K. Chen, ―Sharing multiple secrets in visual cryptography‖, Pattern Recognition, Vol. 40, Issue 12, pp. 3633 - 3651, 2007.

[10] Tzung-Her Chen, Kai-Hsiang Tsao, and Kuo-Chen Wei, ―Multi-Secrets Visual Secret Sharing‖, Proceedings of APCC2008, IEICE, 2008.

[11] Provos, N. (2001). Scanning USENET for Steganography.from http://niels.xtdnet.nl /stego /usenet. php.

[12] Schildt, H. The Complete Reference Java 2, Fifth Ed. TMH, Pp 799-839.

Appendix I

Source Code

Encryption Code

```
function varargout = Encryptiongui(varargin)
% ENCRYPTIONGUI MATLAB code for Encryptiongui.fig
%      ENCRYPTIONGUI, by itself, creates a new ENCRYPTIONGUI or
raises the existing
%      singleton*.
%
%      H = ENCRYPTIONGUI returns the handle to a new ENCRYPTIONGUI or
the handle to
%      the existing singleton*.
%
%      ENCRYPTIONGUI('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in ENCRYPTIONGUI.M with the given
input arguments.
%
%      ENCRYPTIONGUI('Property','Value',...) creates a new
ENCRYPTIONGUI or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before Encryptiongui_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to Encryptiongui_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Encryptiongui

% Last Modified by GUIDE v2.5 24-Apr-2016 12:01:44

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Encryptiongui_OpeningFcn, ...
                   'gui_OutputFcn',  @Encryptiongui_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Encryptiongui is made visible.
function Encryptiongui_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Encryptiongui (see VARARGIN)

% Choose default command line output for Encryptiongui
handles.output = hObject;
global y;
handles.y = varargin;
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Encryptiongui wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Encryptiongui_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in imgbtn.
function imgbtn_Callback(hObject, eventdata, handles)
% hObject    handle to imgbtn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[f1, p1]= uigetfile({'*.jpg';'*.bmp';'*.tif'}, 'File Selector');
handles.img1 = strcat(p1, f1);
set(handles.img1edt,'string',handles.img1);
handles.img1 = imread(handles.img1);
guidata(hObject,handles)

% --- Executes on button press in img2btn.
function img2btn_Callback(hObject, eventdata, handles)
% hObject    handle to img2btn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
[f2, p2]= uigetfile({'*.bmp'}, 'File Selector');
handles.a = strcat(p2, f2);
set(handles.img2edt,'string',handles.a);
handles.a = imread(handles.a);
guidata(hObject,handles)

% --- Executes on button press in encryptbtn.
function encryptbtn_Callback(hObject, eventdata, handles)
% hObject    handle to encryptbtn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
tic;
if isfield(handles,'img1')
    if isfield(handles, 'a')
        handles.img1 = double(handles.img1);
[row, col] = size(handles.img1);

handles.img2 = imresize(handles.a, [row, col]);
handles.img2 = double(handles.img2);
for x=1:1:row
    for y=1:1:col
        temp1 = dec2bin(handles.img1(x,y),8);
        temp2 = dec2bin(handles.img2(x,y),8);
        temp1(8) = temp2(1);
        temp1(7) = temp2(2);
        temp1(6) = temp2(3);
        temp1(5) = temp2(4);
        temp1(4) = temp2(5);
        handles.stego(x,y) = bin2dec(temp1);
    end
end



handles.stego = uint8(handles.stego);

t = fix(clock);

m = t(2);   %month
handles.stego(row, col) = m;

d = t(3);   %day
handles.stego(row, col-1) = d;

hh = t(4);  %hours
handles.stego(row, col-2) = hh;

mm = t(5)  %minutes
handles.stego(row, col-3) = mm;

handles.stego(row, col-4) = y; %validity period

imwrite(handles.stego, 'C:\Users\Parag\Desktop\newimage.tif');
    end
end
```

```
toc;
axes(handles.axes1)
imshow(handles.stego)
guidata(hObject,handles);


function img1edt_Callback(hObject, eventdata, handles)
% hObject    handle to img1edt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of img1edt as text
%        str2double(get(hObject,'String')) returns contents of
img1edt as a double


% --- Executes during object creation, after setting all properties.
function img1edt_CreateFcn(hObject, eventdata, handles)
% hObject    handle to img1edt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function img2edt_Callback(hObject, eventdata, handles)
% hObject    handle to img2edt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of img2edt as text
%        str2double(get(hObject,'String')) returns contents of
img2edt as a double


% --- Executes during object creation, after setting all properties.
function img2edt_CreateFcn(hObject, eventdata, handles)
% hObject    handle to img2edt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.y = str2num(get(handles.edit3,'string'));

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3
as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

Decryption code

```
function varargout = decodegui(varargin)
% DECODEGUI MATLAB code for decodegui.fig
%      DECODEGUI, by itself, creates a new DECODEGUI or raises the 
existing
%      singleton*.
%
%      H = DECODEGUI returns the handle to a new DECODEGUI or the 
handle to
%      the existing singleton*.
%
%      DECODEGUI('CALLBACK',hObject,eventData,handles,...) calls the 
local
%      function named CALLBACK in DECODEGUI.M with the given input 
arguments.
%
%      DECODEGUI('Property','Value',...) creates a new DECODEGUI or 
raises the
%      existing singleton*.  Starting from the left, property value 
pairs are
%      applied to the GUI before decodegui_OpeningFcn gets called.  
An
%      unrecognized property name or invalid value makes property 
application
%      stop.  All inputs are passed to decodegui_OpeningFcn via 
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows 
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help decodegui

% Last Modified by GUIDE v2.5 22-Apr-2016 20:55:36

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @decodegui_OpeningFcn, ...
                   'gui_OutputFcn',  @decodegui_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before decodegui is made visible.
function decodegui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to decodegui (see VARARGIN)

% Choose default command line output for decodegui
handles.output = hObject;
clc
global imgd;
handles.imgd = varargin;
global addr;
handles.addr = varargin;
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes decodegui wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = decodegui_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function dbrowseedt_Callback(hObject, eventdata, handles)
% hObject    handle to dbrowseedt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dbrowseedt as text
%        str2double(get(hObject,'String')) returns contents of
dbrowseedt as a double


% --- Executes during object creation, after setting all properties.
function dbrowseedt_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dbrowseedt (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in decryptbtn.
function decryptbtn_Callback(hObject, eventdata, handles)
% hObject    handle to decryptbtn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
tic;
handles.imgd = double(handles.imgd);
[row col] = size(handles.imgd);
t = fix(clock);
m = t(2);
d = t(3);
hh = t(4);
mm = t(5);
m1 = handles.imgd(row,col);
d1 = handles.imgd(row,col-1);
hh1 = handles.imgd(row,col-2);
mm1 = handles.imgd(row,col-3);
val = handles.imgd(row,col-4);

if(m==m1 & d==d1 & hh==hh1 & mm>mm1 & mm<=mm1+val)
for x=1:1:row
    for y=1:1:col
        temp = dec2bin(handles.imgd(x,y),8);
        if(temp(8)=='0')
            temp = '00000000';
            handles.secret_data(x,y) = bin2dec(temp);
        else
            temp = '11111111';
            handles.secret_data(x,y) = bin2dec(temp);
        end
    end
end
handles.secret_data = uint8(handles.secret_data);
imwrite(handles.secret_data,
'C:\Users\Parag\Desktop\newimage22.tif');
axes(handles.axes1)
imshow(handles.secret_data)

else
    err1 = msgbox('Session timed out. Image discarded');
    delete(handles.addr);
end
toc;
guidata(hObject,handles);

% --- Executes on button press in dbrowsebtn.
function dbrowsebtn_Callback(hObject, eventdata, handles)
% hObject    handle to dbrowsebtn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[f2, p2] = uigetfile({'*.tif'},'File Selector');
```

```
handles.imgd = strcat(p2, f2);
handles.addr = handles.imgd;
set(handles.dbrowseedt,'string',handles.imgd);
handles.imgd = imread(handles.imgd);
guidata(hObject,handles)
```