A MINI PROJECT REPORT ON

VIDEO STEGANOGRAPHY

Submitted in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

Computer Science and Technology

By

MATURU PRANEETH (20A81A0632)

Under the Esteemed Supervision of

Mr. M. BHANU RANGA RAO M.Tech

Department of Computer Science and Technology

SRI VASAVI ENGINEERING COLLEGE (Autonomous)

(Affiliated to JNTUK, Kakinada)

Pedatadepalli, Tadepalligudem-534101, A.P 2022-23

SRI VASAVI ENGINEERING COLLEGE (Autonomous)

Department Of Computer Science and Technology

Pedatadepalli, Tadepalligudem

Certificate

This is to certify that the Project Report entitled "VIDEO STEGANOGRAPHY" submitted By **MATURU PRANEETH (20A81A0632)** for the award of the degree of Bachelor of Technology in the Department of Computer Science and Technology during the academic year 2022-2023.

Name of Project Guide

Mr. M. BHANU RANGA RAO M Tech

Head of the Department

Dr.Jaya Kumari M Tech, Ph D
Professor & HOD.

External Examiner

DECLARATION

We hereby declare that the project report entitled "VIDEO STEGANOGRAPHY" Submitted by us to Sri Vasavi Engineering College (Autonomous), Tadepalligudem, affiliated to JNTUK Kakinada in partial fulfilment of the requirement for the award of the degree of B.Tech in Computer Science and Technology of Bonafide project work carried out by us under the guidance of Mr. M. BHANU RANGA RAO. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this institute or any other institute or University.

Project Associates

M. PRANEETH (20A81A0632)

ACKNOWLEDGEMENT

First and foremost, we sincerely salute to our esteemed institute **SRI VASAVI ENGINEERING COLLEGE**, for giving us this golden opportunity to fulfil our warmdream to become an engineer.

Our sincere gratitude to our project guide **Mr.M. BHANU RANGA RAO,** Department of Computer Science and Technology, for his timely cooperation and valuable suggestions while carrying out this project.

We express our sincere thanks and heartful gratitude to **Dr. D. Jaya Kumari**, Professor & Head of the Department of Computer Science and Technology, for permitting us to do our project.

We express our sincere thanks and heartful gratitude to **Dr. G.V.N.S.R. RatnakaraRao**, Principal, for providing a favourable environment and supporting us during the development of this project.

Our special thanks to the management and all the teaching and non-teaching staff members, Department of Computer Science and Technology, for their support and cooperation in various ways during our project work. It is our pleasure to acknowledge the help of all those respected individuals.

We would like to express our gratitude to our parents, friends who helped to complete this project.

Project Associates

M. PRANEETH (20A81A0632)

TABLE OF CONTENTS

1.ABSTRACT

The project entitled Video Steganography is the application developed to embed a video file in another video signal. It is concerned with embedding information in an innocuous cover Speech in a secure and robust manner. This system makes the Files more secure by using the concepts Steganography and Cryptography.

Steganography, poor cousin of Cryptography is the art of hiding messages inside other messages such that the very existence of the message is unknown to third party. The goal of cryptography is to make data unreadable by a third party, the goal of Steganography is to hide the data from a third party Through the use of advanced computer software, authors of images and software can place a hidden trademark in their product, allowing them to keep a check on piracy.

2.Literature Survey

SNo	Author	Title	Advantages	Disdavantages
1.	Abhinav Thakur	Secure video Steganography based on DWT(2015)	Better performance in terms of high Embedding	initially all signals aregenerally represented in time domain.
2.	Ahmed Salem	Metadata hiding for Unmanned video based on digital(2016)	saving the storage as well as increasing the data safety from a potential loss	There is less accuarcy
3.	Said E	A security enhanced robust video embedding (2015)	Provides large capacity of Steganography due to trailing coefficients and high robustness	The disadvantages of this system are that it works only on
4.	Wong M	A secure video steganography based on LSB technique	easy and computationally	There is no authentication

3.System Study and Analysis

3.1 Problem Statement:

- A special problem for steganography is the conversion of digital files to different formats or with different compression levels. Both can affect the embedded information, and the technology needs to be robust against this type of attack and signal modification.
- The usage of Internet in the world is increasing very highly. Present day transactions are considered to be "untrusted" in terms of security, i.e. they are relatively easy to be hacked. We also have to consider the transfer of large amount of data through the network which will give errors at the time of transferring and only the single level of security

3.2 Proposed System:

- In this project, we have proposed Enhanced TEA(Tiny Encryption Algorithm) with embedding(ETEA) text into video. Enhanced TEA provides security during transmission of data transfer. In TEA, only encryption was possible. But in our proposed algorithm ETEA, encryption and embedding both are combined to provide high-level of security to the data so that it couldn’t be hacked.
- In this technology, the end user identifies an video file, which is going to act as the carrier of data. The data file is also selected and then to achieve greater speed of transmission the data file and video file are sent. Prior to this the data is embedded into the video and then sent. The video if hacked or interpreted by a third party user will open up in any video player but not displaying the data.

3.3 Functional Requirements:

- **Security:** In this, we make the data encrypt or decrypt
    - **Encrypt**: In this,we provide password for the data file and change into human unreadable format.
    - **Decrypt**: In this,we need to enter password for the data file and change into human unreadable to readable format.
- **Stegnography**
    - **Embed**: In this, we need to attach video file and encrypted file.
    - **Dembed :** In this, we to provide video file then it seperates the video file and encrypted file.

3.4 Non-Functional Requirements :

This project is intended to meet the following non functional requirements:

- Program should be free from errors
- This should provide high security for the secret data
- Extensibility
- Reusability

Software Requirements :

- **Operating System**

Windows NT/2000 or Above(Client/Server).
Java (jdk-18)

Hardware Requirements :

- Processor(intel i3 Processor or above)
- Ram: 4GB
- Rom:128GB

4.DESIGN

The System Design includes the maintenance of the secure file transfer service with a prescribed encryption format and split at the interested level of encryption, and embed process and the receiving service at the other end with de-embed and decryption process. The design also includes the provision of facility to the user to manipulate the concerned information according to his personal use and communication process.

The design also needs to provide the communication channel to the user to communicate with other registered users through the mailing services in a reliable and secured format. Authorization and authentication services are preferred most for this purpose. The System Design includes the maintenance authorization services, File and directory services with a prescribed encryption format at the interested level of encryption and the receiving service at the other end with decryption process. The design also includes the provision of facility to the user to manipulate the concerned information according to his personal use.

The design of Video Steganography system, basically involve the interface architecture, Security services, and communication system.

In the interface design we involve with the design of the user interface with GUI standards and a proper navigation system where the user need to enter into the flow of transactions authorization services are check and further access is provided into the system. Then the user needs to select into the operations provided through the GUI where encryption, embedding, de-embedding, Decryption, and sending of the file, General information and exit are provided.

Here the Encryption and decryption and services are provided connecting to the security services module where the encryption and decryption are carried out using the cryptographic standards implementing the Tiny algorithm.

After decryption process is completed the user is selecting the file for encryption. After encryption of the file is completed the user is to select the file for embedding it to the video file and sending through the network to the desired user by specifying the targeted users system IP address in the panel designed. Then the system gets connected to the targeted user and delivers the file in video format after which the user working with the Video Steganography software should go for the option De-Embed Files and decrypt the file by selecting the file path by which the file gets decrypted the file and is viewed on the system.

4.2 Modules

The Modules of the system are:

1) Tiny Algorithm Implementation Module
2) Stenography Module
3) GUI Module

Tiny Encryption Algorithm:

The Tiny Encryption Algorithm (TEA) is a cryptographic algorithm designed to minimize memory footprint and maximize speed. It is a Feistel type cipher that uses operations from mixed (orthogonal) algebraic groups. This research presents the cryptanalysis of the Tiny Encryption Algorithm. In this research we inspected the most common methods in the cryptanalysis of a block cipher algorithm. TEA seems to be highly resistant to differential cryptanalysis, and achieves complete diffusion (where a one bit difference in the plaintext will cause approximately 32 bit differences in the cipher text) after only six rounds. Time performance on a modern desktop computer or workstation is very impressive.

As computer systems become more pervasive and complex, security is increasingly important. Cryptographic algorithms and protocols constitute the central component of systems that protect network transmissions and store data. The security of such systems greatly depends on the methods used to manage, establish, and distribute the keys employed by the cryptographic techniques. Even if a cryptographic algorithm is ideal in both theory and implementation, the strength of the algorithm will be rendered useless if the relevant keys are poorly managed.

The following notation is necessary for our discussion.

• Hexadecimal numbers will be subscripted with “h,” e.g., 10 = 16. h

Bitwise Shifts: The logical shift of x by y bits is denoted by x << y. The logical right shift of x by y bits is denoted by x >> y.

Bitwise Rotations: A left rotation of x by y bits is denoted by x <<< y. A right rotation of x by y bits is denoted by x >>> y.

Exclusive-OR: The operation of addition of n-tuples over the field (also known as 2F exclusive-or) is denoted by $x \oplus y$.

The Tiny Encryption Algorithm is a Feistel type cipher that uses operations from mixed (orthogonal) algebraic groups. A dual shift causes all bits of the data and key to be mixed repeatedly. The key schedule algorithm is simple; the 128-bit key K is split into four 32-bit blocks K = (K[0], K[1], K[2], K[3]). TEA seems to be highly resistant to differential cryptanalysis and achieves complete diffusion (where a one bit difference in the plaintext will cause approximately 32 bit differences in the cipher text). Time performance on a

workstation is very impressive.
Block ciphers where the cipher text is calculated from the plain text by repeated application of the same transformation or round function. In a Feistel cipher, the text being encrypted is split into two halves. The round function, F, is applied to one half using a sub key and the output of F is (exclusive-or-ed (XORed)) with the other half. The two halves are then swapped. Each round follows the same pattern except for the last round where there is often no swap. The focus of this thesis is the TEA Feistel Cipher.

Steganography:

Stegnography is art of hiding information in ways that prevent the detection of hidden messages. Stegnography derived from Greek, literally means “Covered Writing”. It includes a vast array of secret communications methods that conceal the message’s very existence. Theses methods are including invisible inks, microdots, character arrangement, digital signature, and covert channels and spread spectrum communications.

In this technology, the end user identifies an video file, which is going to act as the carrier of data. The data file is also selected and then to achieve greater speed of transmission the data file and video file are sent. Prior to this the data is embedded into the video and then sent. The image if hacked or interpreted by a third party user will open up in any video player but not displaying the data. This protects the data from being invisible and hence is secure during transmission. The user in the receiving end uses another piece of code to retrieve the data from the video file.

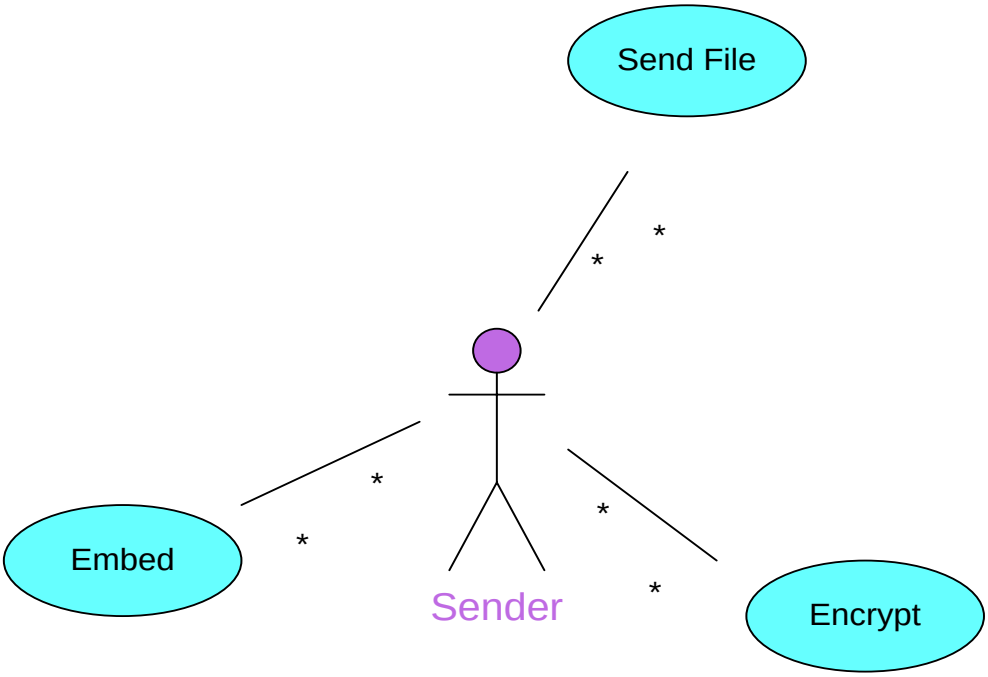
The module deals with identifying the hidden data in the video file. The module receives the video file that is then browsed to remove the associated data. The data is then removed from the video file.

Graphical User Interface:

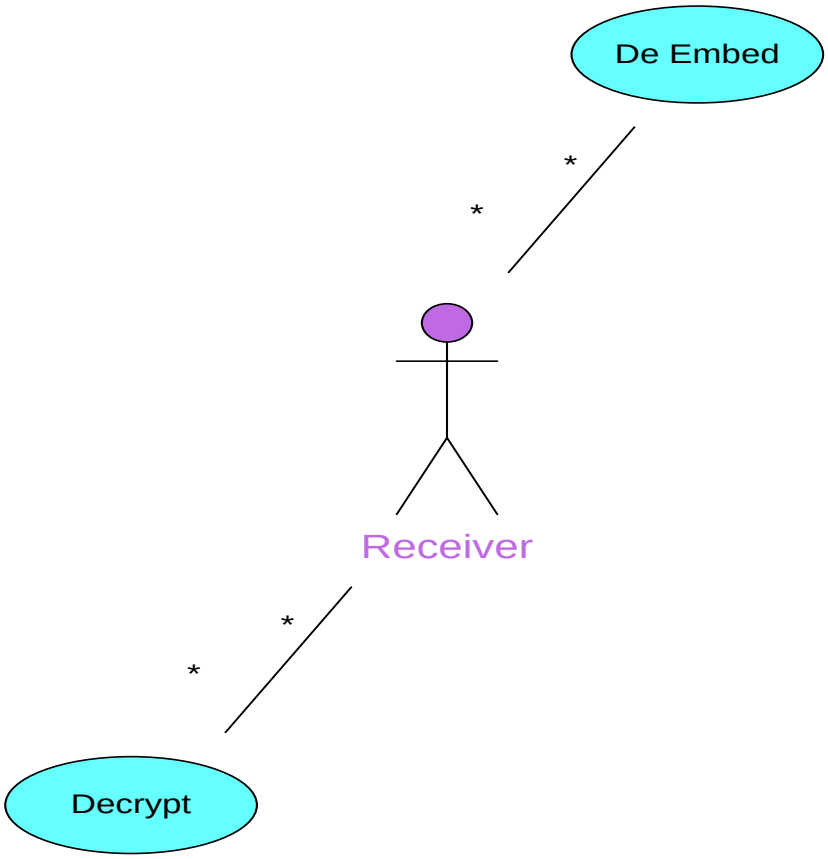
This project is developed using graphics in java swings. The options available are displayed in a menu format, like in an online editor. Clicking on any particular menu item through mouse or through keyboard a dropdown menu is displayed, listing all the options available under that menu item and the user can select the needed actions according to their wish.

4.3 Use case Diagram

Sender:



Receiver:



4.4Use case Description:

Use case name	Encrypt
Participating actors	Sender
Flow of events	The user-selected file will be encrypted with a given key.
Entry Condition	User must select the file and must give the key for encryption.
Exit condition	Successful or Un Successful Encryption of file.
Quality Requirements	Display proper error messages while Encryption.

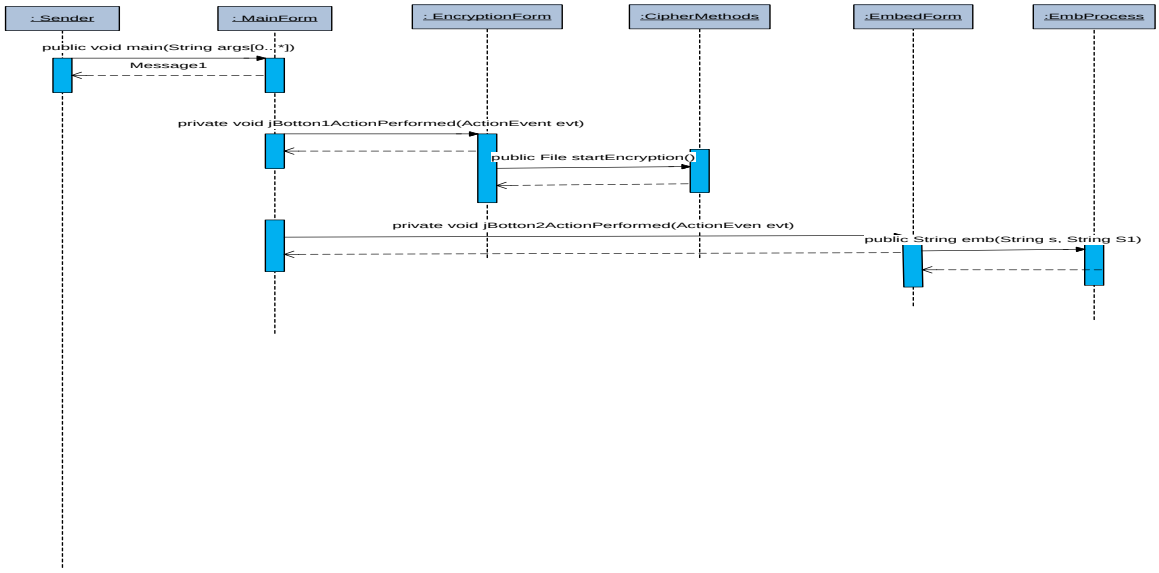
Use case name	Decrypt
Participating actors	Receiver
Flow of events	The user-selected file will be decrypted with a proper key.
Entry Condition	User must select the file and must give the key for decryption.
Exit condition	Successful or Un Successful Decryption of file.
Quality Requirements	Display proper error messages while Decryption.

Use case name	Embed
Participating actors	Sender
Flow of events	The user-selected encrypted file will be embedding with selected video file.
Entry Condition	User must select the one encrypted file and one video file for embedding.
Exit condition	Successful or Un Successful Embedding process.
Quality Requirements	Display proper error messages while Embedding two files.

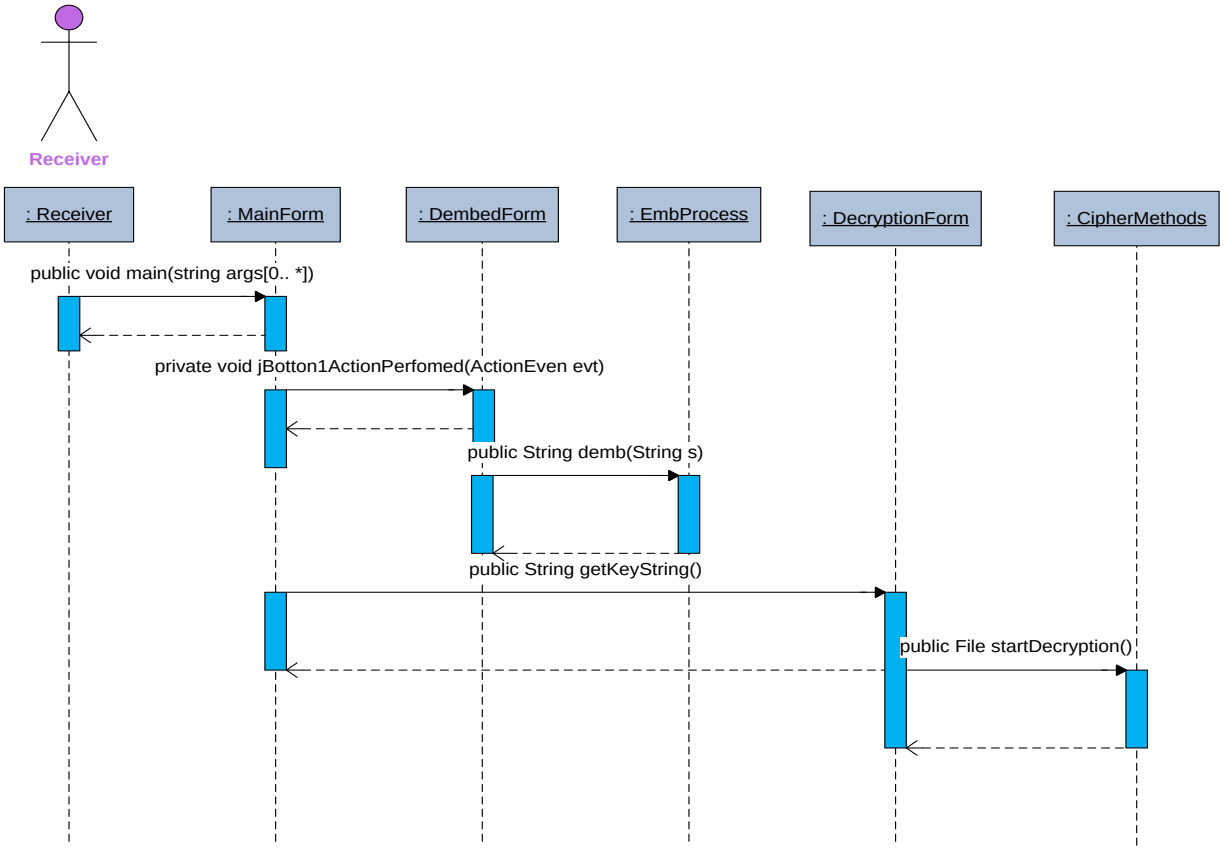
Use case name	De-Embed
Participating actors	Receiver
Flow of events	The user-selected video file will be de-embedding to encrypted file.
Entry Condition	User must select the video file for de-embedding.
Exit condition	Successful or Un Successful De-embedding of file.
Quality Requirements	Display proper error messages while De-embedding.

4.5 Sequence Diagrams

Sender:



Receiver:



FEATURES OF THE LANGUAGE USED

About Java

Initially the language was called as “oak” but it was renamed as “Java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer’s language.
- Java is cohesive and consistent.
- Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.

Finally, Java is to Internet programming where C was to system programming.

Applications and Applets

An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java’s ability to create Applets makes it important. An Applet is an application designed, to be transmitted over the Internet and executed by a Java –compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and dynamically change.

FEATURES OF JAVA

Security

Every time you that you download a “normal” program, you are risking a viral infection. Prior to Java, most users did not download executable programs frequently, and those who did scanned them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both of these concerns by providing a “firewall” between a networked application and your computer. When you use a Java-compatible Web browser, you can safely download Java applets without fear of virus infection or malicious intent.

Portability

For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed .As you will see, the same mechanism that helps ensure security also helps create portability. Indeed, Java’s solution to these two problems is both elegant and efficient.

The Byte code

The key that allows the Java to solve the security and portability problem is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to execute by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.

Translating a Java program into byte code helps makes it much easier to run a program in a

wide variety of environments. The reason is, Once the run-time package exists for a given system, any Java program can run on it.

Although Java was designed for interpretation, there is technically nothing about Java that prevents on-the-fly compilation of byte code into native code. Sun has just completed its Just In Time (JIT) compiler for byte code. When the JIT compiler is a part of JVM, it compiles byte code into executable code in real time, on a piece-by-piece, demand basis. It is not possible to compile an entire Java program into executable code all at once, because Java performs various run-time checks that can be done only at run time. The JIT compiles code, as it is needed, during execution.

Java Virtual Machine (JVM)

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that's has been generated by the compiler will not corrupt the machine that it's loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code.

.java .class

The above picture shows the development process a typical Java programming uses to produce byte codes and executes them. The first box indicates that the Java source code is

located in a. Java file that is processed with a Java compiler called **JAVA**. The Java compiler produces a file called a. class file, which contains the byte code. The class file is then loaded across the network or loaded locally on your machine into the execution environment is the Java virtual machine, which interprets and executes the byte code.

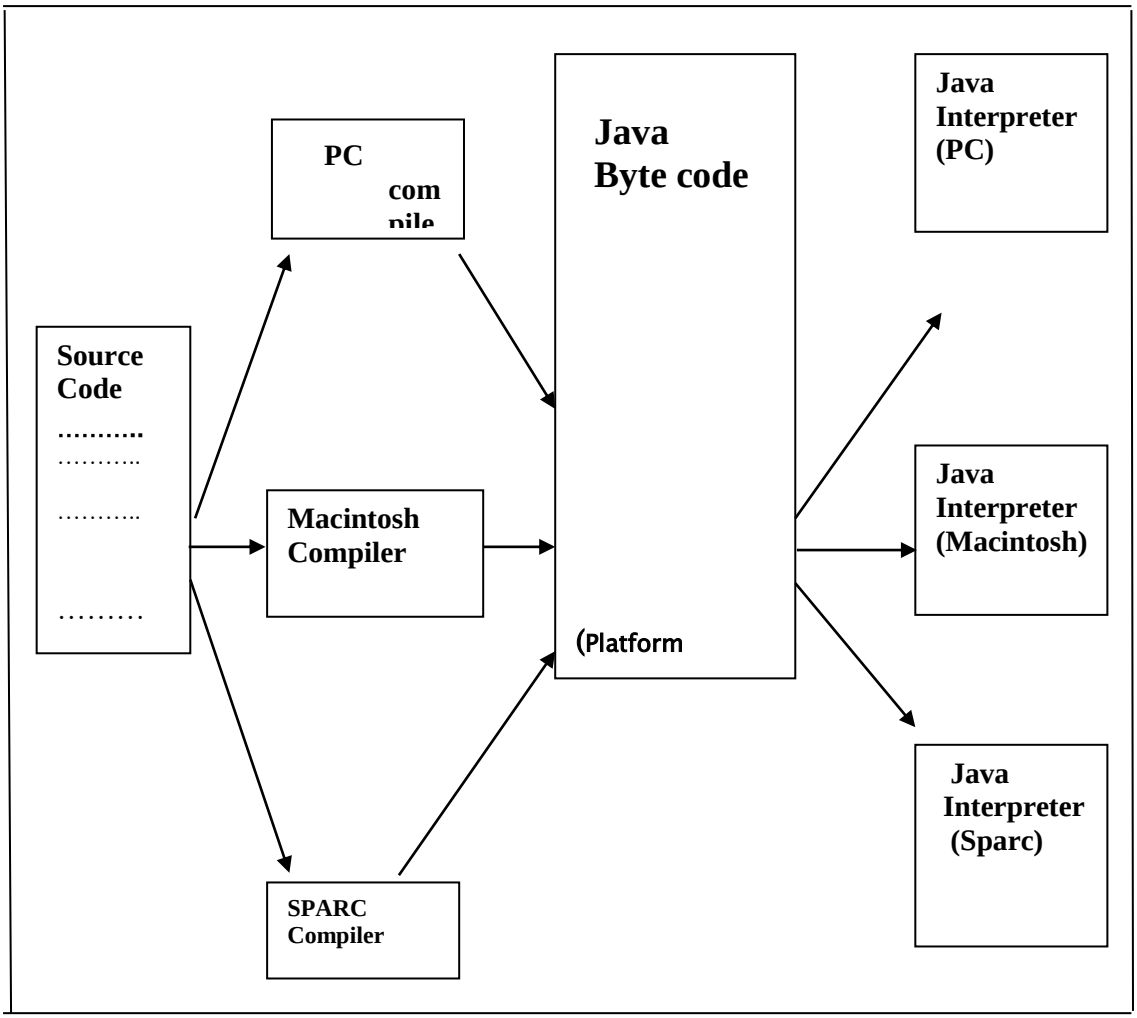
Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

Compilation of Code

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

Compiling and interpreting Java Source Code



During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be a Intel Pentium Windows 95 or SunSARC station running Solaris or Apple Macintosh running system and all could receive code from any computer through Internet and run the Applets.

Simple

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ programmer, learning Java will be even easier. Because Java inherits the C/C++ syntax and many of the object oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

Object-Oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

Robust

The multi-platform environment of the Web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and run time. Java virtually eliminates the problems of memory management and de-allocation, which is completely automatic. In a well-written

Java program, all run time errors can –and should –be managed by your program.

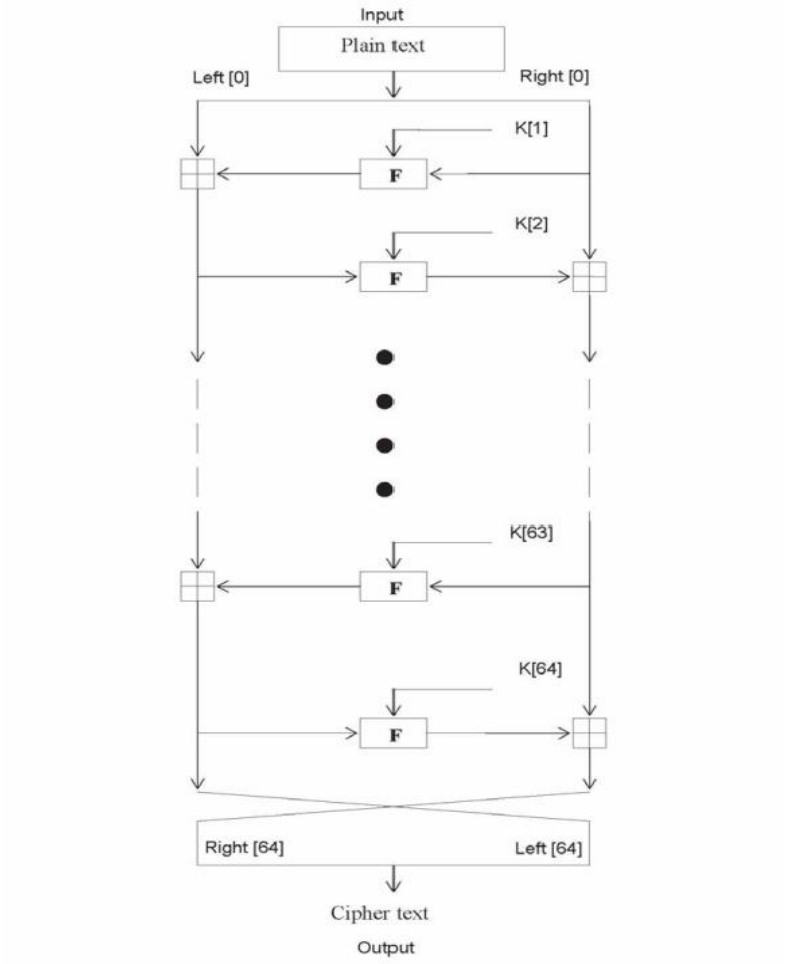
6.IMPLEMENTATION

The Tiny Encryption Algorithm (TEA) is a steganographic algorithm designed to minimize memory footprint and maximize speed. It is a Feistel type cipher that uses operations from mixed (orthogonal) algebraic groups. This research presents the cryptanalysis of the Tiny Encryption Algorithm.

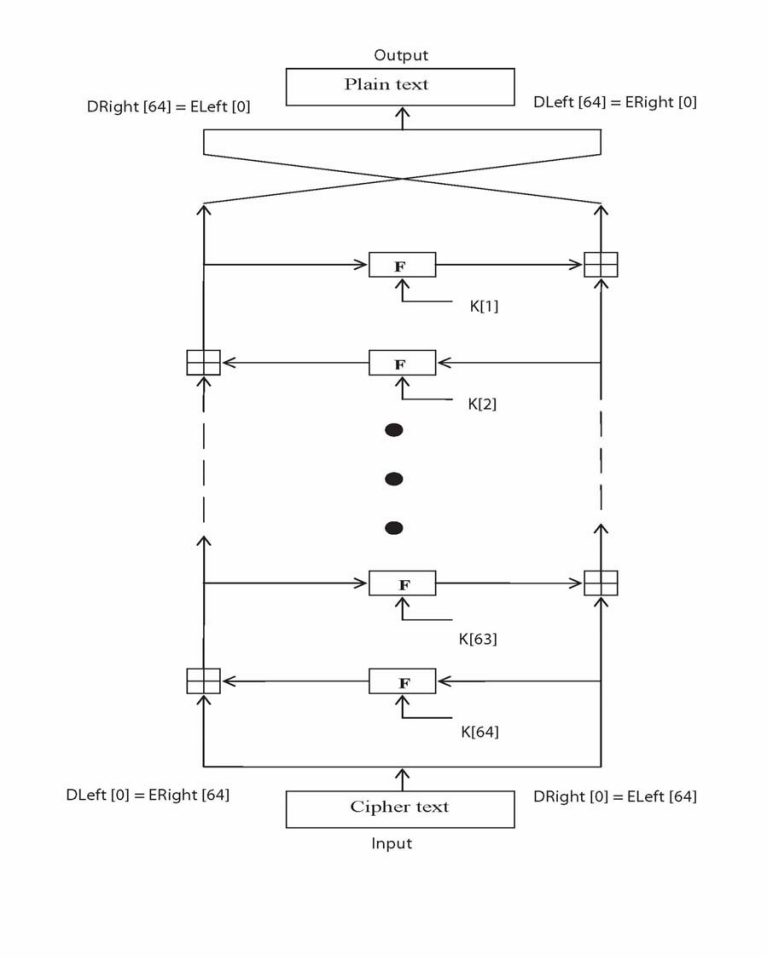
Enhanced TEA (Tiny Encryption Algorithm) with embedding (ETEA) text into video.In TEA, only encryption was possible. But in our proposed algorithm ETEA, encryption and embedding both are combined to provide high-level of security to the data so that it couldn't be hacked.

The Tiny Encryption Algorithm is a Feistel type cipher that uses operations from mixed (orthogonal) algebraic groups. A dual shift causes all bits of the data and key to be mixed repeatedly. The key schedule algorithm is simple; the 128-bit key K is split into four 32-bit blocks K = (K[0], K[1], K[2], K[3]). TEA seems to be highly resistant to differential cryptanalysis and achieves complete diffusion (where a one bit difference in the plaintext will cause approximately 32 bit differences in cipher text).

6.1ENCRIPTION PROCESS



6.2 DECRYPTION PROCESS



6.3 Testing

The test procedure used in the testing process is **Black box testing**. Test cases are analyzed accordingly.

Black Box Testing

This test involves the manual evaluation of the flow from one module to the other and check accordingly for the process flow. This process of testing is with the following criteria

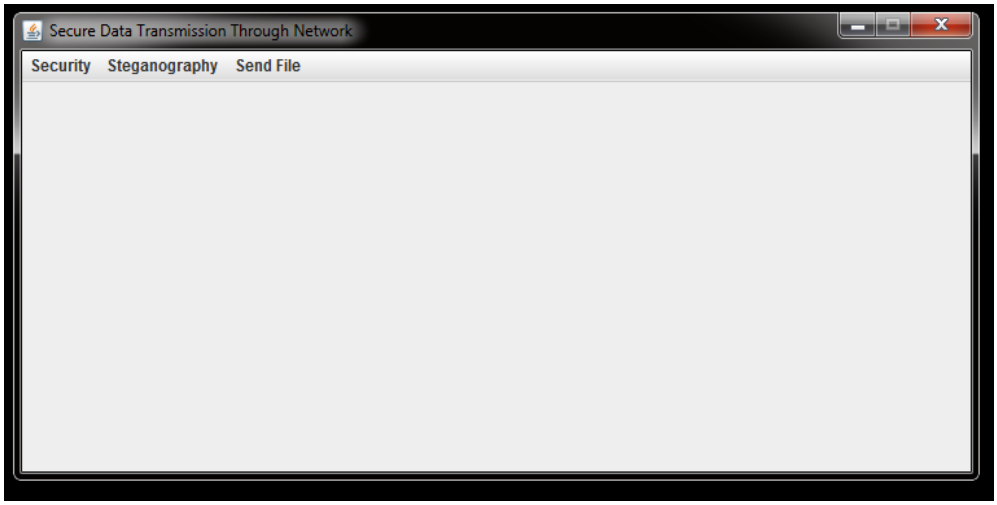
- Encryption process
- Embedding Process
- Retrieving Process
- Decryption process
- Sending file to another peer
- Key prescription Information display
- Exit process

Case Generation Report:

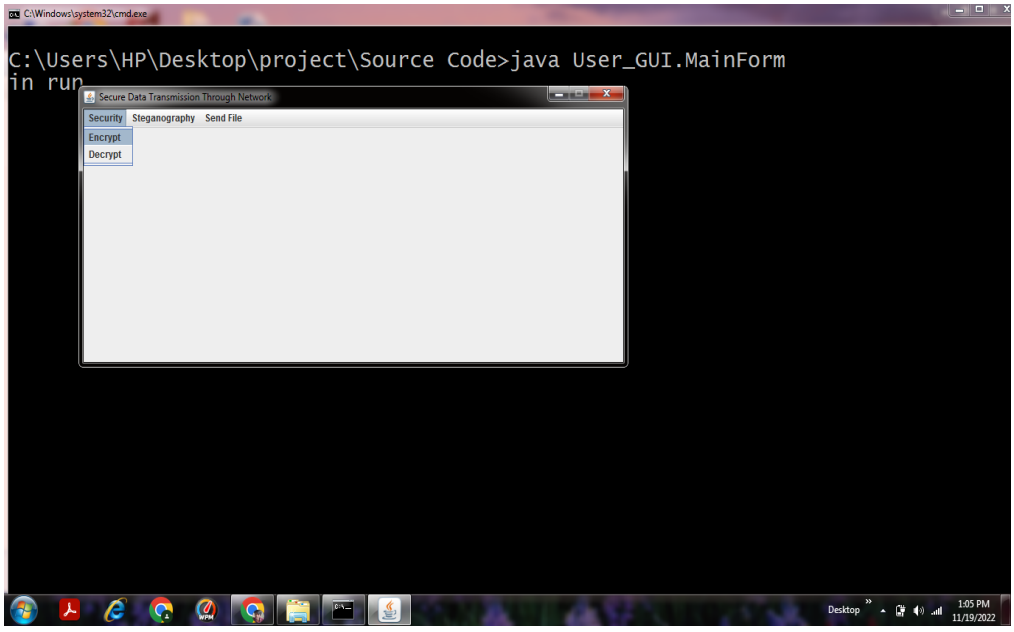
Test Type	Case	Expected Result
Operational / Unit / Functional Test	Encryption	Receive the file to be encrypted and encrypt according to the key and save
-do-	Decryption	Receive the file to be decrypted and decrypt with same key and save
-do-	Sending file	Receives the file from destination and transmit.
-do-	Exit	Ends the current login session
-do-	Embed	Receive the encrypted file and receive the Video file to embed and save in the same location
-do-	Retrieve	Receives the Folder contains video file to retrieve the encrypted file and save in to the same folder

6.4 Methodology

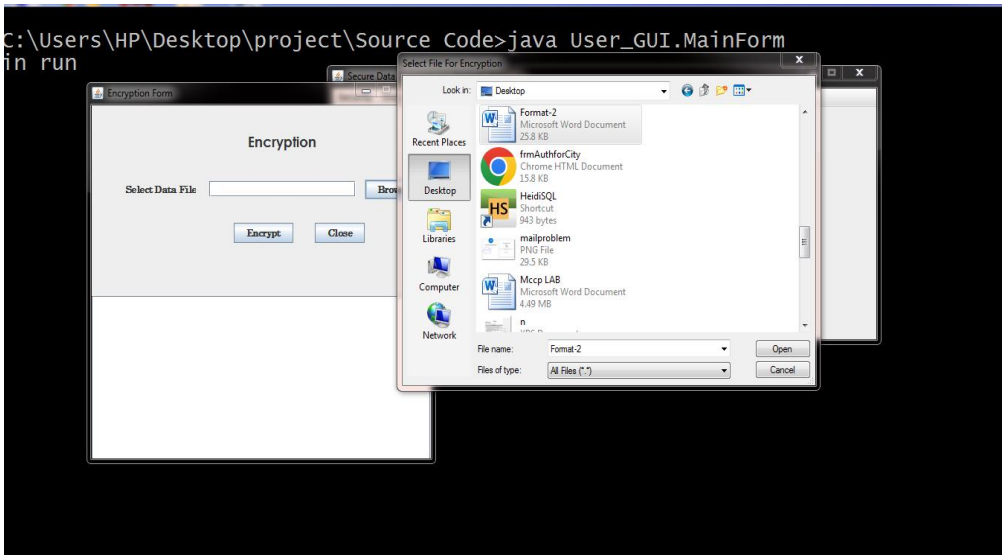
Output Screens:



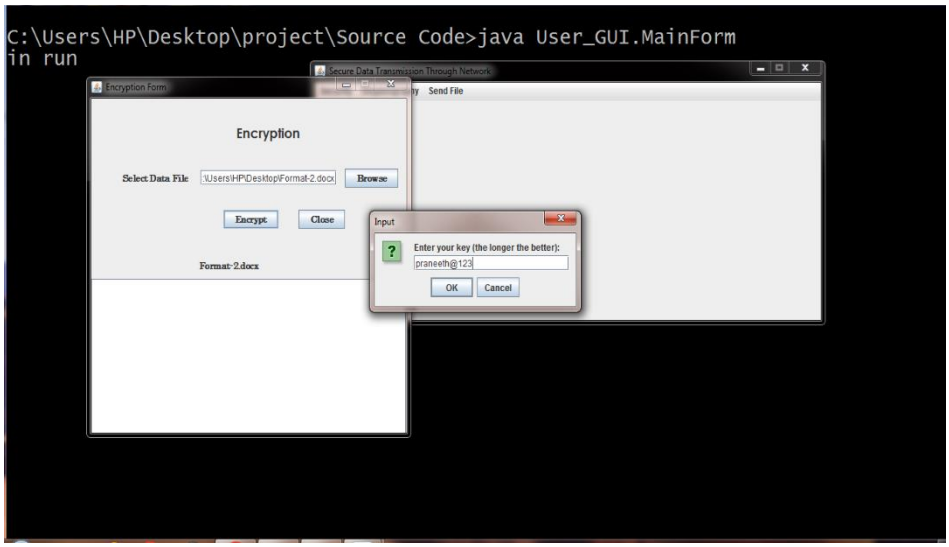
Encripting:



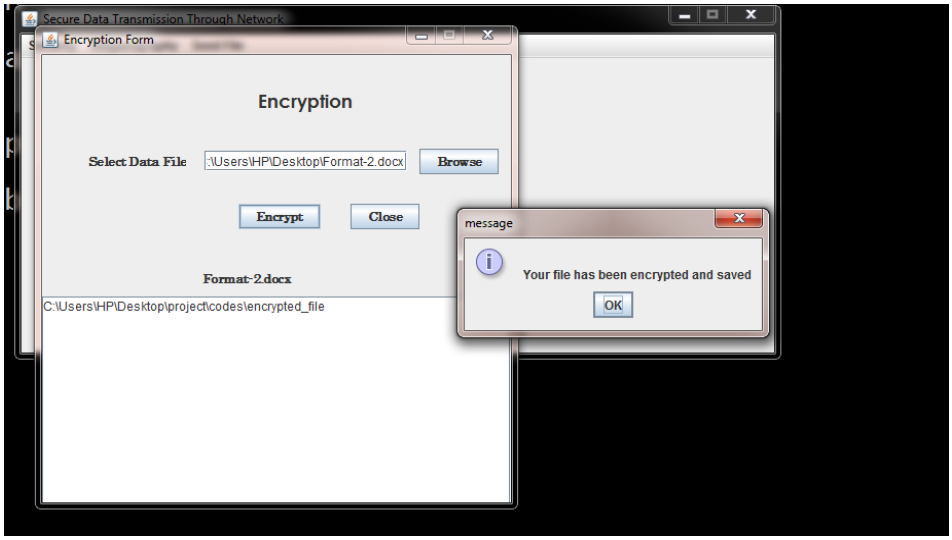
Selecting File to Encript:



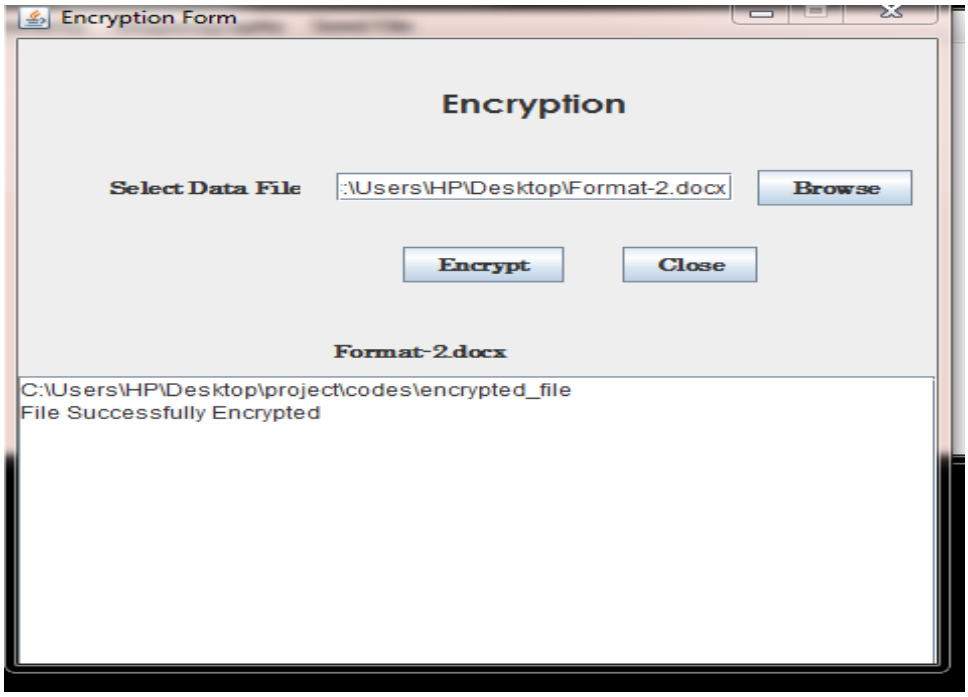
Setting password to File:



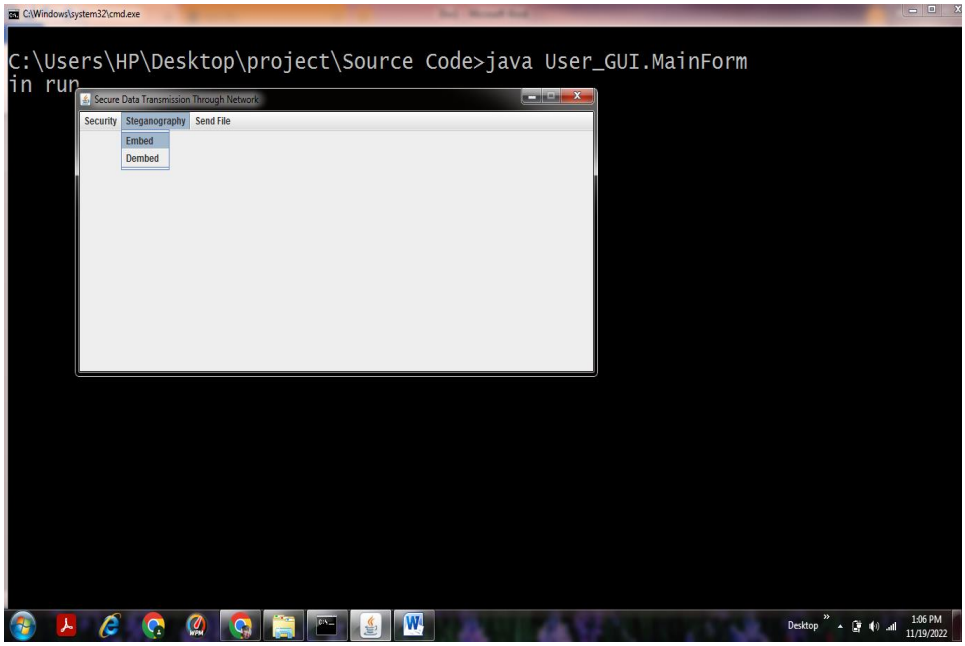
Save Encripted File:



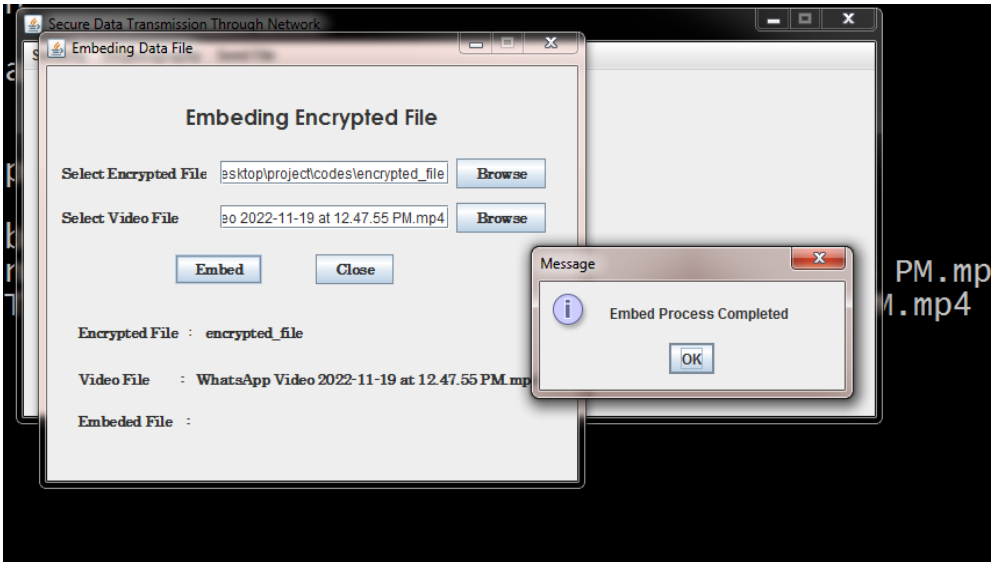
File Encripted:



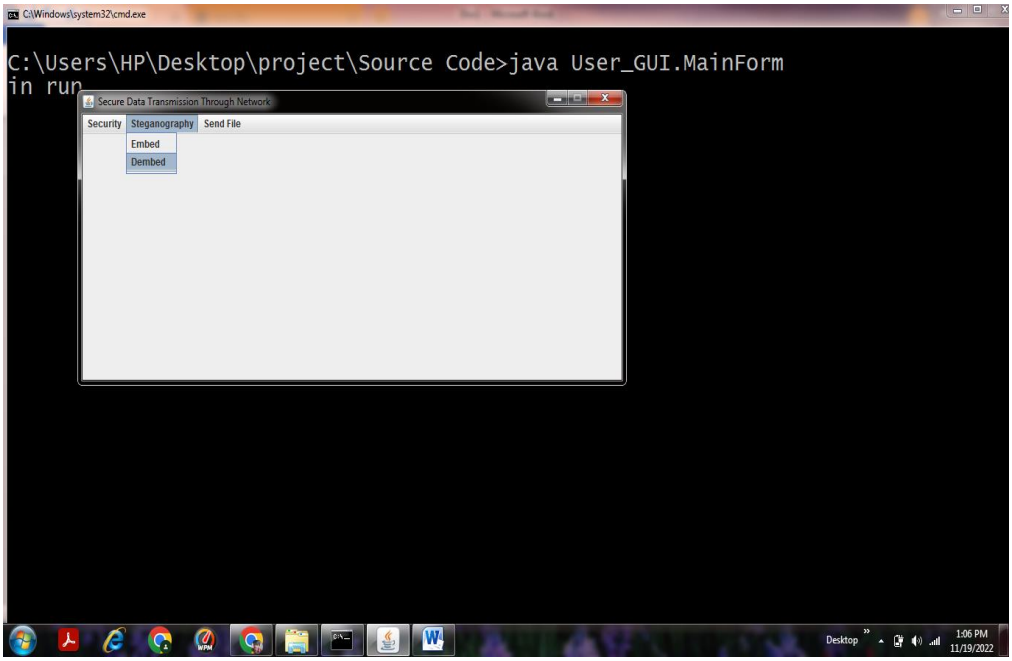
Embedding:

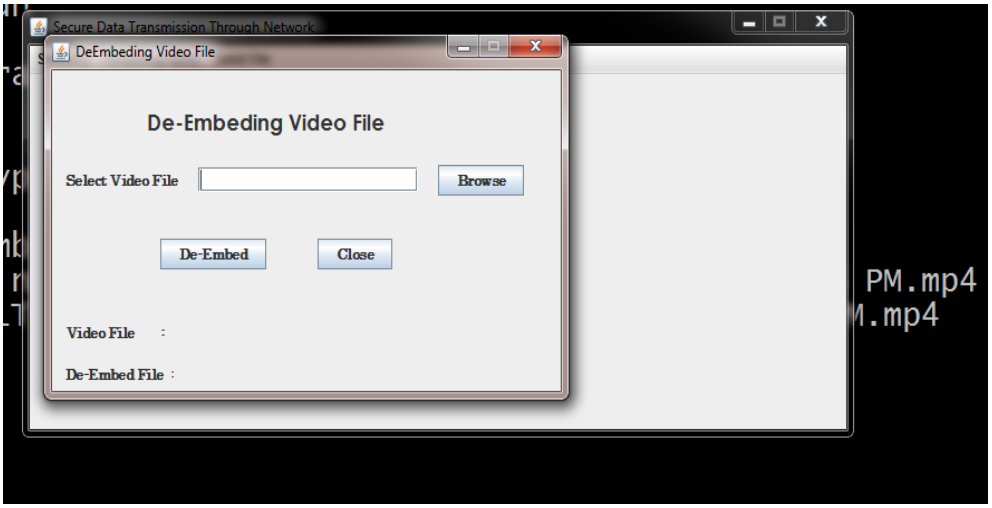

C:\Windows\system32\cmd.exe
C:\Users\HP\Desktop\project\Source Code>java User_GUI.MainForm
in run
Secure Data Transmission Through Network
Security
Steganography
Send File
Embed
Dembed
Desktop
1:06 PM
11/19/2022

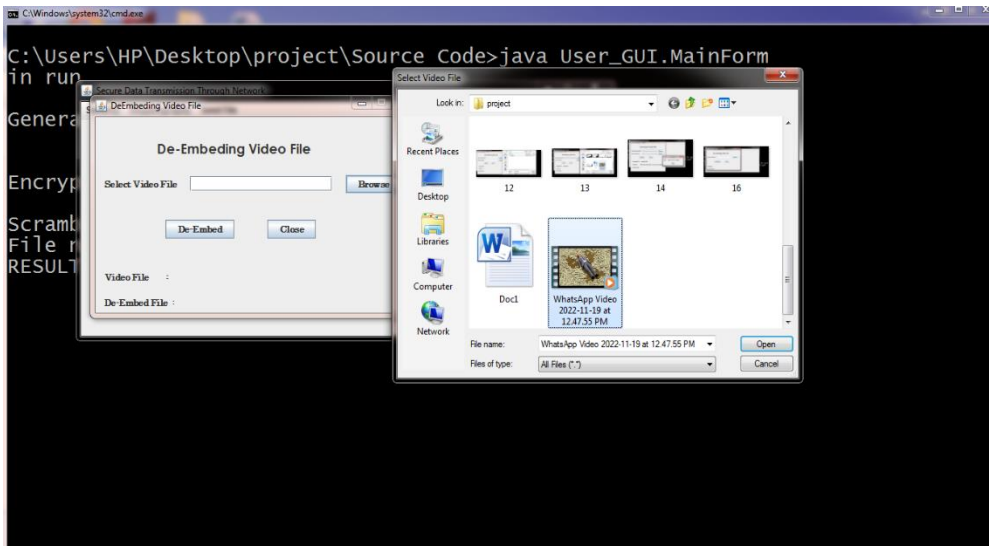

Embeding Data File
Embeding Encrypted File
Select Encrypted File
Browse
Select Video File
Browse
Embed
Close
Encrypted File :
Video File :
Embeded File :

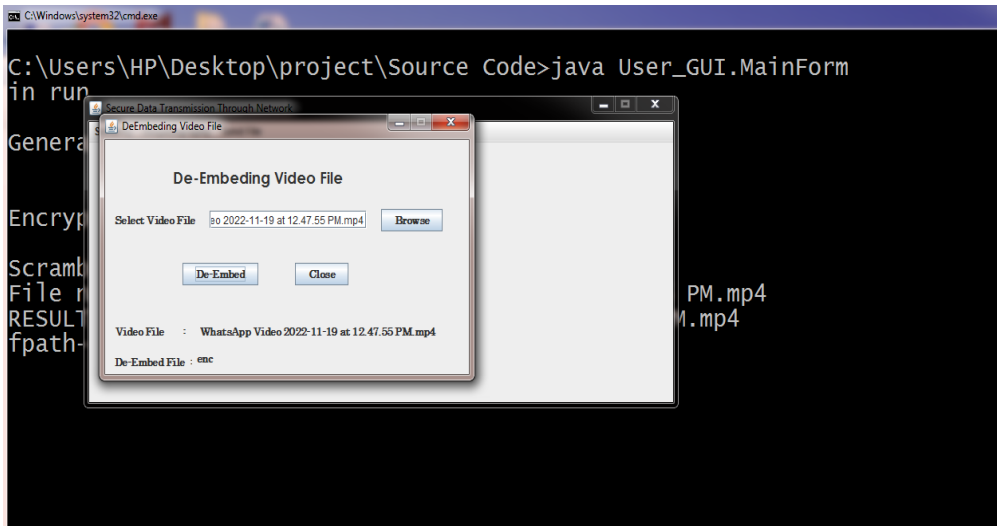

Secure Data Transmission Through Network
Embeding Data File
Embeding Encrypted File
Select Encrypted File
esktop\project\codes\encrypted_file
Browse
Select Video File
eo 2022-11-19 at 12.47.55 PM.mp4
Browse
Embed
Close
Encrypted File : encrypted_file
Video File : WhatsApp Video 2022-11-19 at 12.47.55 PM.mp
Embeded File :
Message
Embed Process Completed
OK
PM.mp
M.mp4

Dembedding:

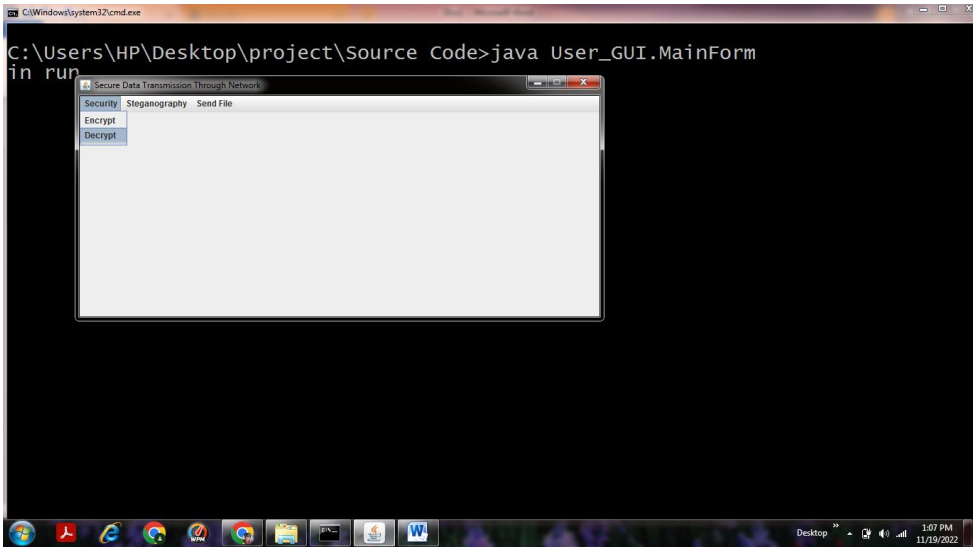

C:\Windows\system32\cmd.exe
C:\Users\HP\Desktop\project\Source Code>java User_GUI.MainForm
in run
Secure Data Transmission Through Network
Security
Steganography
Send File
Embed
Dembed
Desktop
1:06 PM
11/19/2022

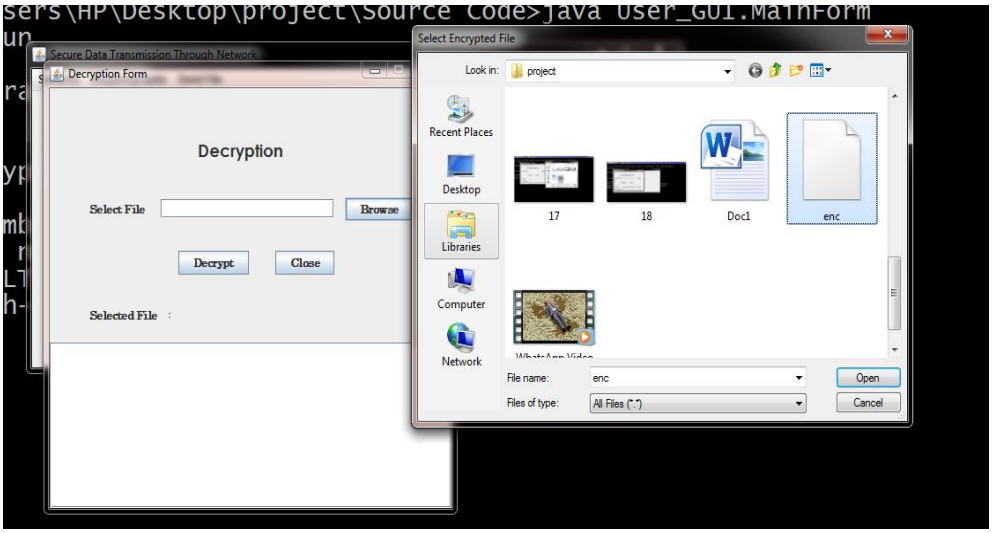

Secure Data Transmission Through Network
DeEmbeding Video File
De-Embeding Video File
Select Video File
Browse
De-Embed
Close
Video File :
De-Embed File :
PM.mp4
M.mp4

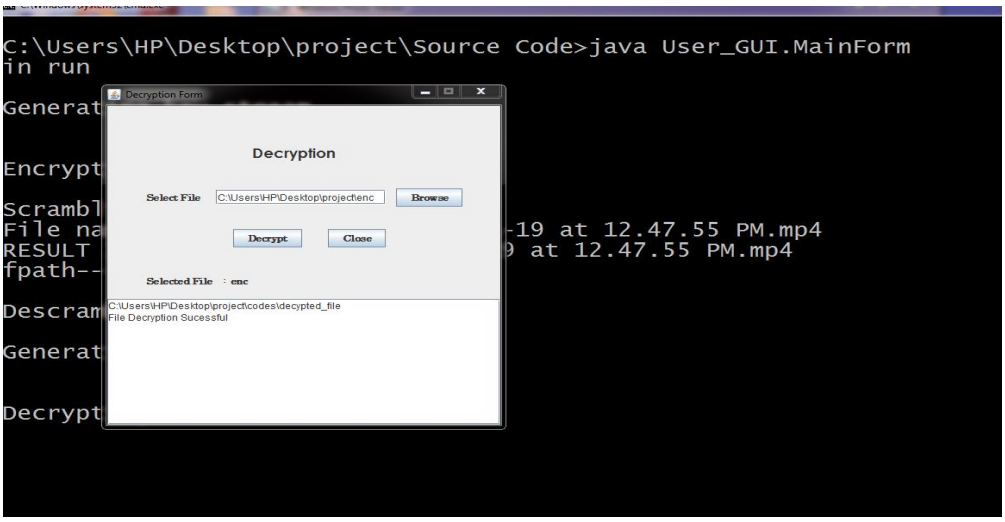

C:\Windows\system32\cmd.exe
C:\Users\HP\Desktop\project\Source Code>java User_GUI.MainForm
Secure Data Transmission Through Network
DeEmbeding Video File
De-Embeding Video File
Select Video File
Browse
De-Embed
Close
Video File :
De-Embed File :
Select Video File
Look in: project
Recent Places
Desktop
Libraries
Computer
Network
12
13
14
16
Doc1
WhatsApp Video 2022-11-19 at 12.47.55 PM
File name: WhatsApp Video 2022-11-19 at 12.47.55 PM
Files of type: All Files (*.*)
Open
Cancel


C:\Windows\system32\cmd.exe
C:\Users\HP\Desktop\project\Source Code>java User_GUI.MainForm
Secure Data Transmission Through Network
DeEmbeding Video File
De-Embeding Video File
Select Video File
Browse
De-Embed
Close
Video File : WhatsApp Video 2022-11-19 at 12.47.55 PM.mp4
De-Embed File : enc
PM.mp4
M.mp4

Decripting:


C:\Users\HP\Desktop\project\Source Code>java User_GUI.MainForm
in run
Secure Data Transmission Through Network
Security
Steganography
Send File
Encrypt
Decrypt


Secure Data Transmission Through Network
Decryption Form
Decryption
Select File
Browse
Decrypt
Close
Selected File :
Select Encrypted File
Look in:
project
Recent Places
Desktop
Libraries
Computer
Network
17
18
Doc1
enc
File name:
enc
Files of type:
All Files (*.*)
Open
Cancel


C:\Users\HP\Desktop\project\Source Code>java User_GUI.MainForm
in run
Decryption Form
Decryption
Select File
C:\Users\HP\Desktop\project\enc
Browse
Decrypt
Close
Selected File : enc
C:\Users\HP\Desktop\project\codes\decypted_file
File Decryption Sucessful

RESULT ANALYSIS

The result of this paper is analyzed with different videos as input. This result provides the double security to any given input. First level security is provided through encrypting the secret information and second level security is provided through Steganography. The Steganography hides secret information behind the innocent cover media. There are different algorithms for implementing the encryption of secret data. The one used in this paper is DES encryption algorithm. The reason behind using the DES algorithm is, it is the simplest and fastest algorithm for encryption of data. The secret data is encrypted using the 16 rounds of encryption techniques. These 16 rounds contain P-box, S-box, E-box of operations. Along with this 16 rounds of encryption the 16 round of key substitution is also performed, which provides 16 different subkeys at each step. These subkeys provide better security for encrypting data. Secret key is the important part of the encryption. Receiver decrypts the secret information using the same secret key provided by sender. A single bit change in the secret key may retrieve the wrong information. At the next level the encrypted data is hidden behind the cover media. The cover media may be audio, video, text, and protocol. This paper is implemented with the video as the cover media.

The result of this paper is that the sender can send his secret message to the receiver, by hiding the message behind the video. There are 3 inputs that we have to accept from user. First is the cover media that is video. The video can be of any size and format of video specified is ".mp4". if the video size is less we can get the fast results. The second input is the secret key, which is private and known only to sender and receiver. The key can be of any length, but smaller key length provides more accurate results. Third input is the secret data to hide. The secret data can be provided as text or select the text file as document to hide. The other options provided are encryption of secret data, after the encryption we can also view the encrypted data in the notepad which is saved in "tempoutput". Once the encryption is done, the data is hidden behind the video. And we can play the encrypted video as shown in the below figure. The encrypted video is saved as output video, so that at the receiver side we can select the same video. At decryption section first the hidden data is retrieved and then DES decryption is performed on the data and final original data is printed. This all options are designed in the user interface. The user interface is designed using the MATLAB tool.

Attainment of Objective

Objectives:

- To provide high security for the valuable information
- The main objective of steganography is to hide a secret message inside harmless cover media in such a way that the secret message is not visible to the observer
- To abstract technique of getting secret data using decryption

Future Work

There is a lot of scope for future work in the implementation, the functionality of the GUI is rather limited, we used other software as well to evaluate the experiments, especially a spreadsheet to compute the correct bpnc values for each extraction. We would want Stegosaurus to calculate bpnc's automatically and let the user browse through the results of previous computations. This would also require a more structured backend, such as a database storing all important information about the feature files or possibly the features directly which would make much larger sets of features tractable. It would be interesting to see if an actual detector confirms the distance meaurements we have found. This detector can be an SVM or a simpler classifier such as an averaged perceptron.

References

S No	TITLE	AUTHOR	PUBLICATION	EDITION
1.	THE COMPLETE REFERENCE JAVA2	PATRICK NAUGHTON HERBERT SCHILDT	TATA McGraw HILL	1994
2.	JAVA CERTIFICATION	JAWORSICK	TECH.MEDIA Publication	1998
3.	COMPLETE JAVA2	ROBERT HELLERS ERNEST	DPB PUBLICATIONS	1998
4.	CRYPTOGRAPHY	JOHN E. HERSHEY DEMYSTIFIED	TATA Mc Graw HILL	2004
5.	MASTERING JAVA SECURITY	RICH HELTON JOHENNIE HELTON	WILEY Dreamtech	2002
6.	SOFTWARE ENGINEERING	ROGER PRESSMAN	TATA Mc Graw HILL	2004