A 3D STEGANALYTIC COMPUTATION AND STEGANALYSIS SAFE WATERMARKING

A PROJECT REPORT

Submitted by

R.SWETHA 211420104283

S.PREETHI 211420104332

G.PRIYANKA 211420104333

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2024

A 3D STEGANALYTIC COMPUTATION AND STEGANALYSIS SAFE WATERMARKING

A PROJECT REPORT

Submitted by

R.SWETHA 211420104283

S.PREETHI 211420104332

G.PRIYANKA 211420104333

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"A 3D STEGANALYTIC COMPUTATION AND STEGANALYSIS SAFE WATERMARKING"** is the bonafide work of **"R.SWETHA [211420104283], S.PREETHI [211420104332] and G.PRIYANKA [211420104333]"** who carried out the project work under my supervision.

SIGNATURE

Dr.L.JABASHEELA M.E.,Ph.D.,
PROFESSOR & HEAD,
HEAD OF THE DEPARTMENT

Department of Computer Science
and Engineering,
Panimalar Engineering College,
Chennai-600 123.

SIGNATURE

Mrs. R. SALINI, M.Tech.,(Ph.D).,
SUPERVISOR
ASSISTANT PROFESSOR,

Department of Computer Science
and Engineering,
Panimalar Engineering College,
Chennai-600 123.

Submitted for the Project Viva-Vice examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION BY THE STUDENT

We **R . SWETHA [211420104283] , S . PREETHI [211420104332]** and **G.PRIYANKA [211420104333]"** here by declare that this project report titled "**A 3D STEGANALYTIC COMPUTATION AND STEGANALYSIS SAFE WATERMARKING**" under the guidance of **Mrs. R. SALINI, M.Tech.,(Ph.D)** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

R. SWETHA

S.PREETHI

G.PRIYANKA

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**.,for his benevolent words and fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors,**Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express my heartfelt thanks to **Dr. L. JABASHEELA,M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to Project Coordinator **Dr.K.VALARMATHI,M.E.,Ph.D** and Project Guide **Mrs. R. SHALINI, M.Tech.,(Ph.D)** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

R.SWETHA

S.PREETHI

G.PRIYANKA

ABSTRACT

We propose a straightforward yet productive steganalytic calculation for watermarks implanted by two best in class 3D watermarking calculations by Cho et al. The fundamental perception is that while in a spotless model the methods/fluctuations of Cho et al's. Standardized histogram canisters are relied upon to take after a Gaussian circulation, in a checked model their appropriation will be bimodal. The proposed calculation assesses the quantity of containers through a comprehensive pursuit and after that the nearness of a watermark is chosen by a carefully fit typicality test or a t-test. We additionally propose an alteration of Cho et al's. Watermarking calculations with the watermark implanted by changing the histogram of the spiral directions of the vertices. Instead of focusing on persistent measurements, for example, the mean or change of the qualities in a canister, the proposed watermarking adjusts a discrete measurement, which here is the stature of the histogram container, to accomplish watermark implanting. Trial comes about show that the adjusted calculation offers not just better resistance against the steganalytic assault we grew, yet additionally an enhanced strength/limit exchange off.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTACT	i
	LIST OF FIGURES	iv
	LIST OF ABBREVIATIONS	v
1	**INTRODUCTION**	6
	1.1 Overview	7
	1.2 Problem Definition	8
2	**LITERATURE REVIEW**	9
3	**THEORETICAL BACKGROUND**	15
	3.1 Implementation Environment	16
	3.2 System Architecture	17
	3.3 Proposed Methodology	18
	3.4 ER Diagram	19
	3.5 Data Flow Diagram	20
	3.6 UML Diagram	22
	3.6.1 Usecase Diagram	22
	3.6.2 Class Diagram	24
	3.6.3 Activity Diagram	26
	3.6.4 Sequence Diagram	28
	3.7 Hardware Environment	29
	3.8 Software Environment	29
4	**SYSTEM IMPLEMENTATION**	30
	4.1 Module Design Specification	31
	4.1.1 Slicing	31
	4.1.2 Motion Compensation	32

	4.1.3 Motion Vector Prediction	32
	4.1.4 Block Transformation and Encoding	32
	4.1.5 Macroblock Ordering	33
	4.2 Algorithm	34
	4.2.1 Embedded Algorithm Description	34
	4.2.2 Cryptography Algorithm	35
5	**RESULTS & DISCUSSION**	36
	5.1 Performance Parameters/Testing	37
	5.1.1 Testing Objectives	37
	5.1.2 Testing Levels	37
6	**CONCLUSION AND FUTURE ENHANCEMENTS**	39
	APPENDICES	41
	6.1 Source Code	41
	6.2 Screenshots	51
	6.3.Plagiarism Report	56
	6.4.Paper Publication	57
	REFERENCES	58

LIST OF FIGURES

FIG NO.	FIGURE DESCRIPTION	PAGE NO.
1	System Architecture	17
2	ER Diagram	19
3	DFD 0	20
4	DFD 1	20
5	OVERALL DFD	21
6	Usecase Diagram	22
7	Class Diagram	24
8	Activity Diagram	26
9	Sequence Diagram	28

LIST OF ABBREVATIONS

S.NO.	ABBREVIATION	EXPANSION
1	VLC	Visible Light Communication
2	JPEG	Joint Photographic Expert Group
3	MPEG	Moving Picture Experts Group.
4	FMO	Future Mode of Operation
5	DCT	Discrete Cosine Transform
6	LSB	Local Standard Time.
7	WDM	Wavelength Division Multiplexing
8	AVI	Audio Video Interleave
9	SVD	Singular Value Decomposition
10	DTCWT	Dual-tree Complex Wavelet Transform
11	PSNR	Peak Signal to Noise Ratio

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

This is an overview of a research paper aims to develop a steganalytic algorithm for detecting watermarks embedded by two state-of-the-art 3D watermarking algorithms developed by Cho et al. The key insight is that while the histogram bins in Cho et al.'s algorithms are expected to follow a Gaussian distribution in a clean model, they exhibit a bimodal distribution in a watermarked model.The proposed algorithm involves a comprehensive search to evaluate the number of bins, followed by the application of a custom-fit normality test or a t-test to determine the presence of a watermark. Additionally, the project suggests modifying Cho et al.'s watermarking algorithms by embedding the watermark through changes in the histogram of the spiral directions of the vertices. Instead of focusing on traditional persistent measurements like mean or variance, this approach adjusts discrete measurements, specifically the height of the histogram bin, to embed the watermark.Experimental results demonstrate that the modified algorithm offers improved resistance against the developed steganalytic attack and achieves an enhanced trade-off between robustness and capacity. This suggests that the proposed method may provide better security and reliability compared to existing techniques.

1.2 PROBLEM DEFINITION

The problem involves addressing the challenge of detecting watermarks embedded by state-of-the-art 3D watermarking algorithms developed by Cho et al. You aim to develop a steganalytic algorithm capable of identifying the presence of such watermarks, leveraging the observation that in a clean (unwatermarked) model, the fluctuations of Cho et al.'s standardized histogram bins follow a Gaussian distribution, whereas in a watermarked model, this distribution becomes bimodal.

CHAPTER 2
LITERATURE REVIEW

(1) TITLE AND JOURNAL:

Toward Authentication of Videos: Integer Transform Based Motion Vector Watermarking (2022)

AUTHORS:

RAFI ULLAH , SULTAN DAUD KHAN1, MOHIB ULLAH , (MEMBER, IEEE), FADI AL-MACHOT, AND HABIB ULLAH.

DESCRIPTION:

Nowadays, digital content like videos, audio and images are widely used as evidence in criminal courts and forensic laboratories. Due to the advanced low-cost and easily available multime- dia/communication tools and softwares, manipulation of the content is a no-brain task. Thus, the protection of digital content originality is a challenge for the content owners and researchers before it can be produced in court or used for some other purpose.

ADVANTAGES:

- The effectiveness of the proposed approach is validated through various quality metrics such as peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), normalized coefficients (NC), and bit error rate (BER).
- Video frames are fully protected in both spatial and transform domains, enhancing the security and robustness of the watermarking technique. Embedding the watermark in the approximation subbands of wavelet transform before embedding ensures comprehensive protection.

DISADVANTAGES:

- While the proposed technique aims to determine attacked regions concisely, the process of detecting and extracting the watermark from the video may require sophisticated algorithms and techniques.

(2) TITLE AND JOURNAL:

DDCA: A Distortion Drift-Based Cost Assignment Method for Adaptive Video Steganography in the Transform Domain (2022)

AUTHORS:

YI CHEN, HONGXIA WANG, KIM-KWANG RAYMOND CHOO, SENIOR MEMBER, PEISONG HE, ZORAN SALCIC, LIFE SENIOR MEMBER, DALI KAAFAR, XUYUN ZHANG

DESCRIPTION:

Cost assignment plays a key role in coding performance and security of video steganography. Existing cost assignment methods (for adaptive video steganography) are designed for specific transform coefficients rather than all transform coefficients. In addition, existing video steganographic frameworks do not allow Syndrome-Trellis Codes (STCs) to modify all transform coefficients in both intra-coded and inter-coded frames at the same time

ADVANTAGES:

- By allowing Syndrome-Trellis Codes (STCs) to modify all transform coefficients, the project potentially makes it harder to detect the presence of hidden messages within the video.
- The framework seems to be applicable to both intra-coded and inter-coded frames, potentially offering wider applicability.

DISADVANTAGES:

- The introduction of a new cost assignment method and potentially modifying all transform coefficients might increase the complexity of the video steganography process compared to existing methods. This could lead to higher computational costs.

(3) TITLE AND JOURNAL:

6G Networks Physical Layer Security Using RGB Visible Light Communications (2021)

AUTHORS:

S. SODERI , AND R. DE NICOLA

DESCRIPTION:

Visible Light Communication (VLC) is a key technology for the sixth-generation (6G) wireless communication thanks to the possibility of using artificial environmental lights as a data transfer channel. Although VLC systems are more resistant against interference and less susceptible to security vulnerabilities like most wireless networks, VLC is even inherently susceptible to eavesdropping attacks.

ADVANTAGES:

- The project proposes a combination of watermarking and jamming to make eavesdropping significantly harder. Watermarking hides the message within the signal, while jamming creates a "secure zone" around the receiver where the hidden message is harder to extract.
- The approach utilizes RGB LEDs, a common component in VLC systems, for both watermarking and jamming, potentially reducing the need for additional hardware.

DISADVANTAGES:

- Combining watermarking, WDM, and jamming might introduce complexity compared to simpler security methods. This could lead to higher processing power requirements and potential challenges in implementation.

(4) TITLE AND JOURNAL:

Blind Camcording-Resistant Video Watermarking in the DTCWT and SVD Domain (2022)

AUTHORS:

MD. ASIKUZZAMAN , HANNES MAREEN , NOUR MOUSTAFA, KIM-KWANG RAYMOND CHOO , AND MARK R. PICKERING

DESCRIPTION:

Video watermarking techniques can be used to prevent unauthorized users from illegally distributing videos across (social) media networks. However, current watermarking solutions are unable to embed a perceptually invisible watermark which is robust to the distortions introduced by camcording.

ADVANTAGES:

- The technique claims to be resistant to camcording attacks, which are known to disrupt traditional watermarks. This includes distortions like compression, noise addition, frame rate changes, and geometric distortions.
- The approach aims to embed watermarks that are invisible to the human eye, preserving the original video quality.

DISADVANTAGES:

- The technique utilizes a combination of Dual-Tree Complex Wavelet Transform (DTCWT) and Singular Value Decomposition (SVD), which might be computationally expensive compared to simpler methods.
- The paper's evaluation might benefit from a wider range of video content and potentially adversarial attacks to strengthen the claims of robustness.

(5) **TITLE AND JOURNAL:**

Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques (2021)

AUTHORS:

NAZEEH GHATASHEH 1, ISMAIL ALTAHARWA 2, AND KHALED ALDEBEI AZEEH GHATASHEH 1, ISMAIL ALTAHARWA 2, AND KHALED ALDEBEI

DESCRIPTION:

Data compression is an important part of information security because compressed data is more secure and easy to handle. Effective data compression technology creates efficient, secure, and easy-to-connect data. There are two types of compression algorithm techniques, lossy and lossless..

ADVANTAGES:

- The proposal combines data compression with RSA encryption, potentially offering a two-layer defense against unauthorized access. Compressing the data might make it harder to decipher even if intercepted, and encryption adds another layer of security.
- By compressing the data before transmission, the technique aims to reduce transmission time, especially over slow internet connections.

DISADVANTAGES:

- The system utilizes a combination of lossy (DWT) and lossless (Huffman) compression. Lossy compression introduces some data alteration, which might not be suitable for all data types, especially critical information.

CHAPTER 3
THEORETICAL BACKGROUND

3.1 IMPLEMENTATION ENVIRONMENT

These libraries offer functionalities for video encoding, decoding, and manipulation, which are crucial for working with compressed video formats like H.264. This includes desktop environments like Windows, macOS, or Linux distributions. Additionally, cloud-based platforms with sufficient computational resources can be utilized for large-scale processing tasks To evaluate the performance of the information hiding methods and techniques, a diverse dataset of compressed videos encoded using the H.264 standard . This dataset should include videos of various resolutions, frame rates, and content types to ensure comprehensive testing and validation. The system should implement a variety of information hiding techniques tailored for compressed video, focusing on methods that manipulate the underlying coding structure of H.264.The implemented system should include modules for evaluating the performance of the information hiding techniques. This includes measuring the payload capacity (number of bits that can be inserted), bitstream size overhead (increase in file size due to embedded data), video quality degradation (evaluated using metrics like PSNR, SSIM), and computational complexity (processing time and resource utilization).While the focus is on H.264-specific methods, image-based information hiding techniques may also be reviewed and implemented if applicable to compressed video. These techniques can provide additional insights and comparisons, especially regarding payload capacity and video quality. Throughout the implementation process, thorough documentation should be maintained, detailing the design, implementation, and evaluation of the system and its components. This documentation will aid in understanding the system's functionality and reproducibility of results. By implementing the proposed system in the described environment, researchers can effectively survey, implement, and evaluate information hiding methods tailored for compressed video, particularly within the H.264 coding structure.

3.2 SYSTEM ARCHITECTURE

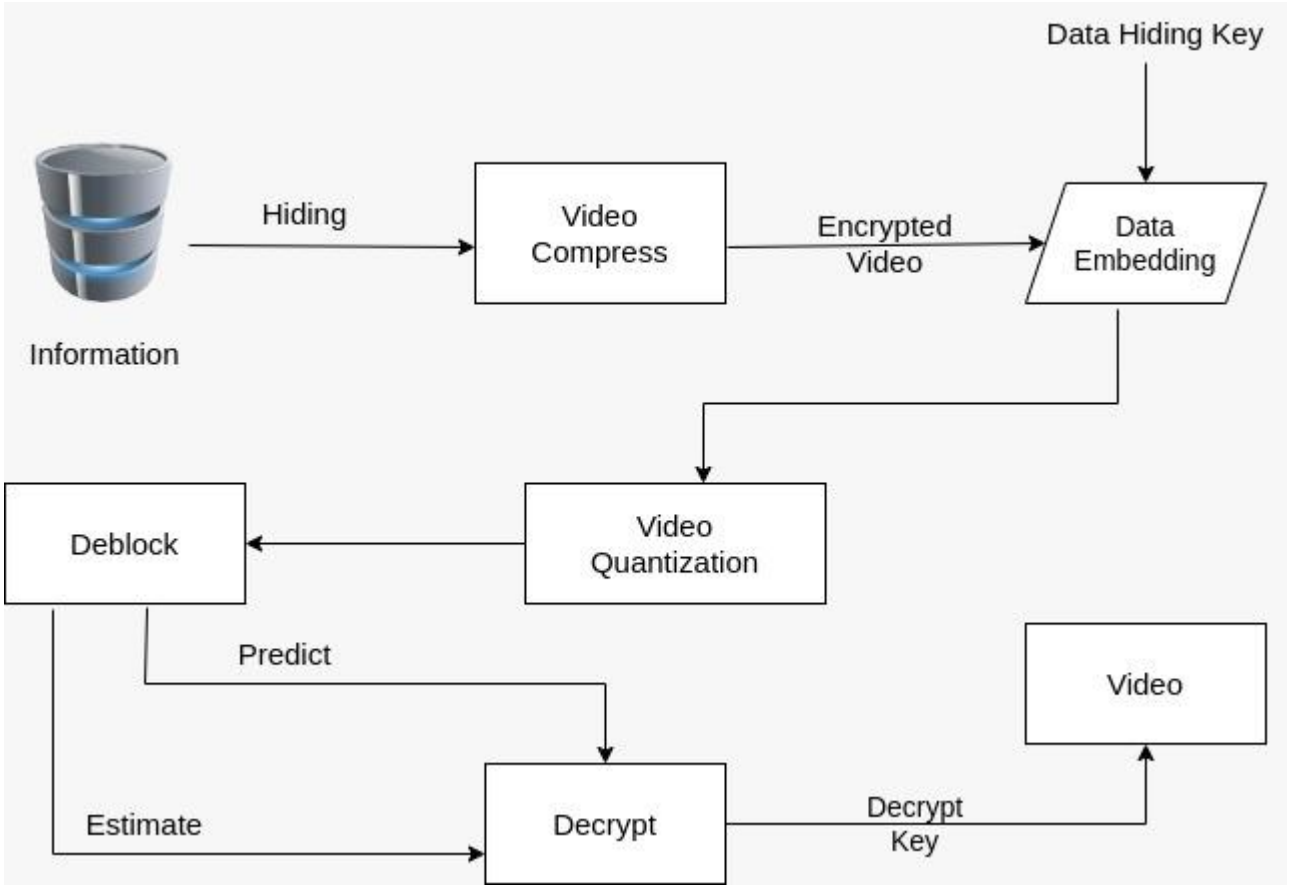


Fig.1 SYSTEM ARCHITECTURE

Information block represents the original data that you want to hide within a cover video. Deblock block might indicate a process of dividing the original data into smaller blocks. Quantization reduces the precision of the data by rounding or mapping values to a smaller set. In the context of image or video compression, quantization can reduce the number of bits needed to represent the data.Predict block likely refers to a prediction step used in video compression techniques like motion estimation. The predictor estimates the value of a pixel based on surrounding pixels and removes the predicted value from the original data. This reduces spatial redundancy in the video. Encrypt: This block represents the encryption of the data before hiding. Encryption scrambles the data using a secret key, making it unintelligible without the key. Embedding block represents the process of embedding the encrypted data into the

compressed video. The diagram doesn't show how the embedding is done, but common techniques include modifying least significant bits (LSBs) of pixels or coefficients.Decrypt block represents the decryption of the hidden data using the same secret key used for encryption.Estimate block likely refers to the estimation process used in video compression decoding. Here, the decoder estimates missing information based on surrounding data.Deblock block might again indicate a process of combining smaller blocks back into the original data.Data block represents the extracted hidden data after decryption.

3.3 PROPOSED METHODOLOGY

We survey on information hiding methods designed specifically for compressed video, illustrate possible hiding venues within the H.264 coding structure for information hiding, and review their applications. We considered H.264 (instead of the latest compression standard, i.e., H.265) because of its rich literatures in various applications. Here, we emphasize on the techniques that manipulate the underlying coding structure of H.264 to realize data embedding and how each of the techniques affects the payload (i.e., the number of bits that can be inserted into the host video), bitstream size overhead, video quality and computational complexity. Nevertheless, at times, information hiding methods designed for image are also reviewed since they can be readily applied to compressed video.

3.4 ER DIAGRAM

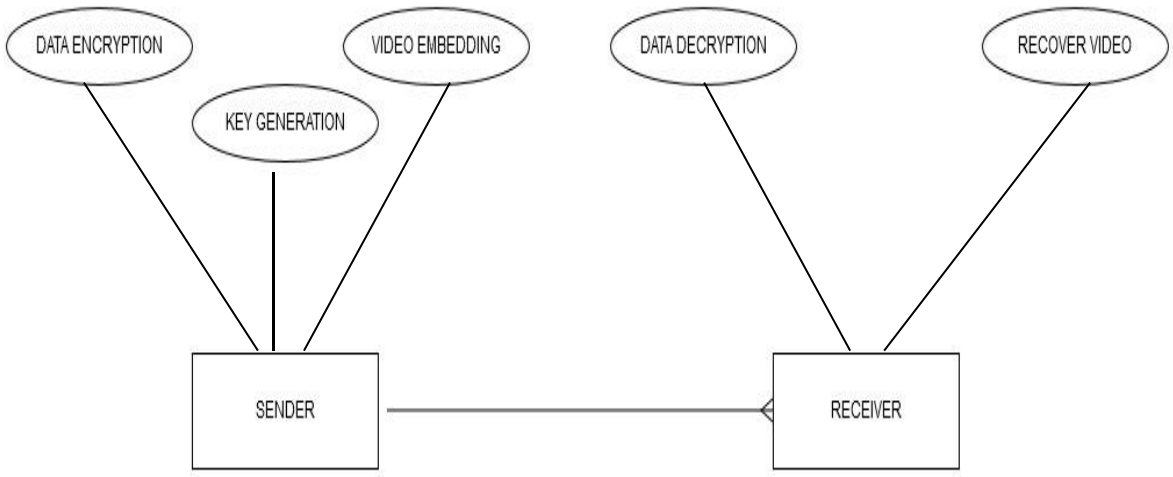


Fig.2 ER DIAGRAM

An ER diagram, which stands for Entity-Relationship Diagram, is a graphical representation that shows the relationships between different entities (things, concepts, or objects) within a system, typically a database. It uses specific symbols like rectangles, diamonds, and lines to depict these entities, their attributes (properties), and how they connect with each other.Video entity represents the video files to be encrypted. It likely has attributes such as VideoID (primary key),Filename, Path, Format and Size. Encryption Key entity represents the keys used to encrypt the videos. It likely has attributes such as KeyID (primary key), KeyData. User entity represents the users of the system, potentially the ones who encrypt or decrypt videos. It likely has attributes such as UserID (primary key), Username and Password. Encrypts relationship connects Videos and Encryption Keys. A video can be encrypted with one key, and a key can be used to encrypt multiple videos. The cardinality ratio between them is likely 1:N, meaning one video can be encrypted with one key, but one key can be used for many videos. Owns relationship connects Users and Encryption Keys. A user can own one or more encryption keys, and a key can be owned by one user. The cardinality ratio here is likely M:N, meaning a user can have many keys, and a key can be owned by multiple users (assuming shared access).

3.5 DATA FLOW DIAGRAM

DFD 0

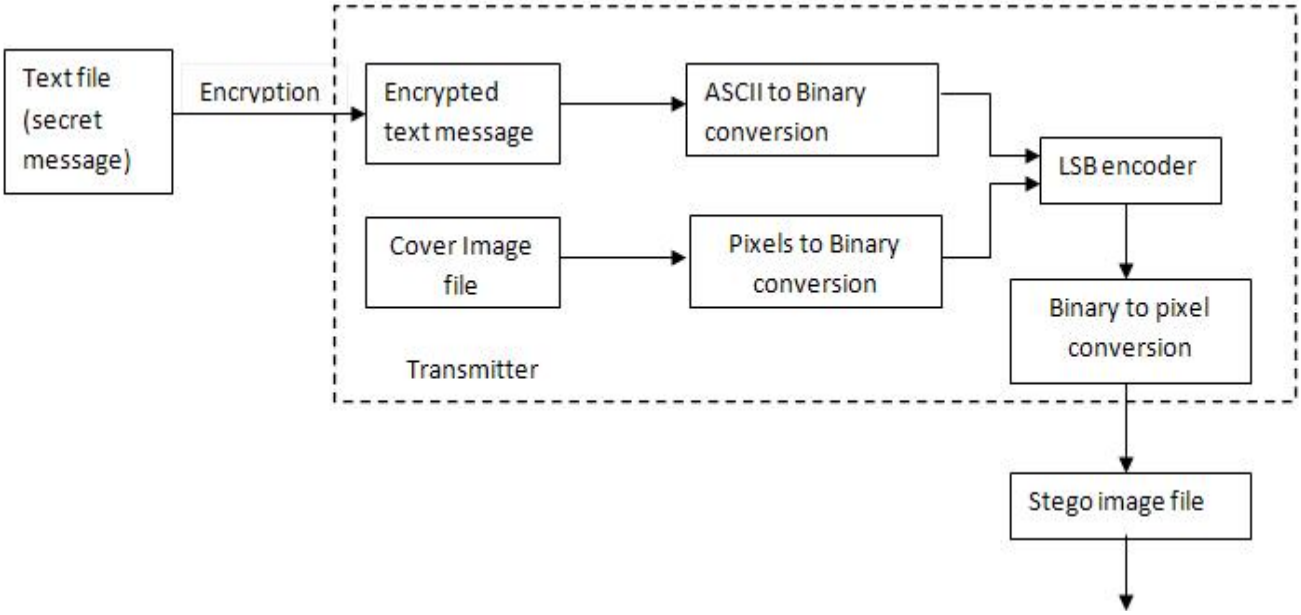


Fig.3 DFD 0

DFD 1

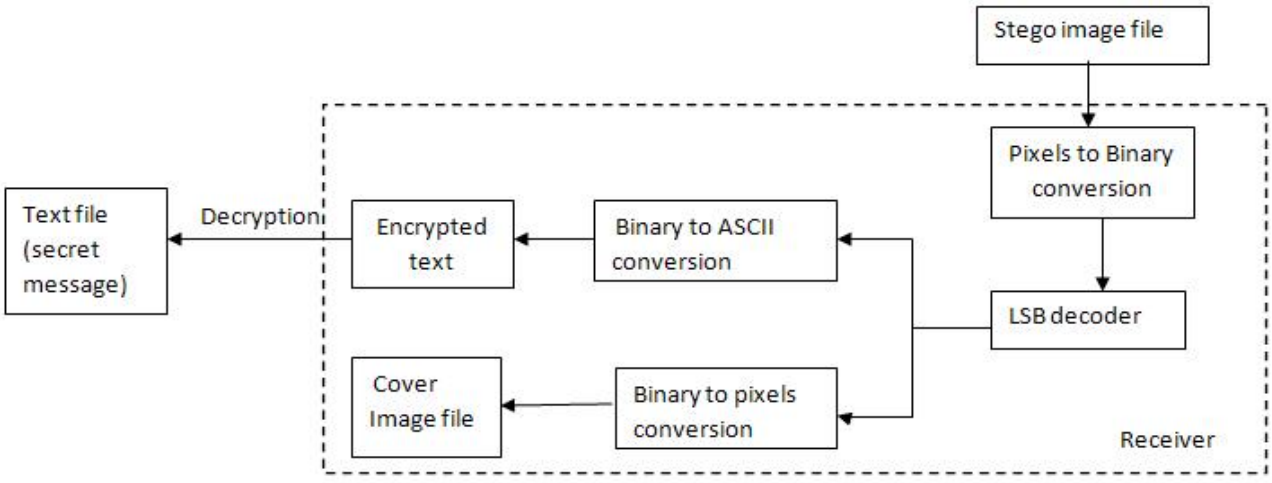


Fig.4 DFD 1

OVERALL DFD

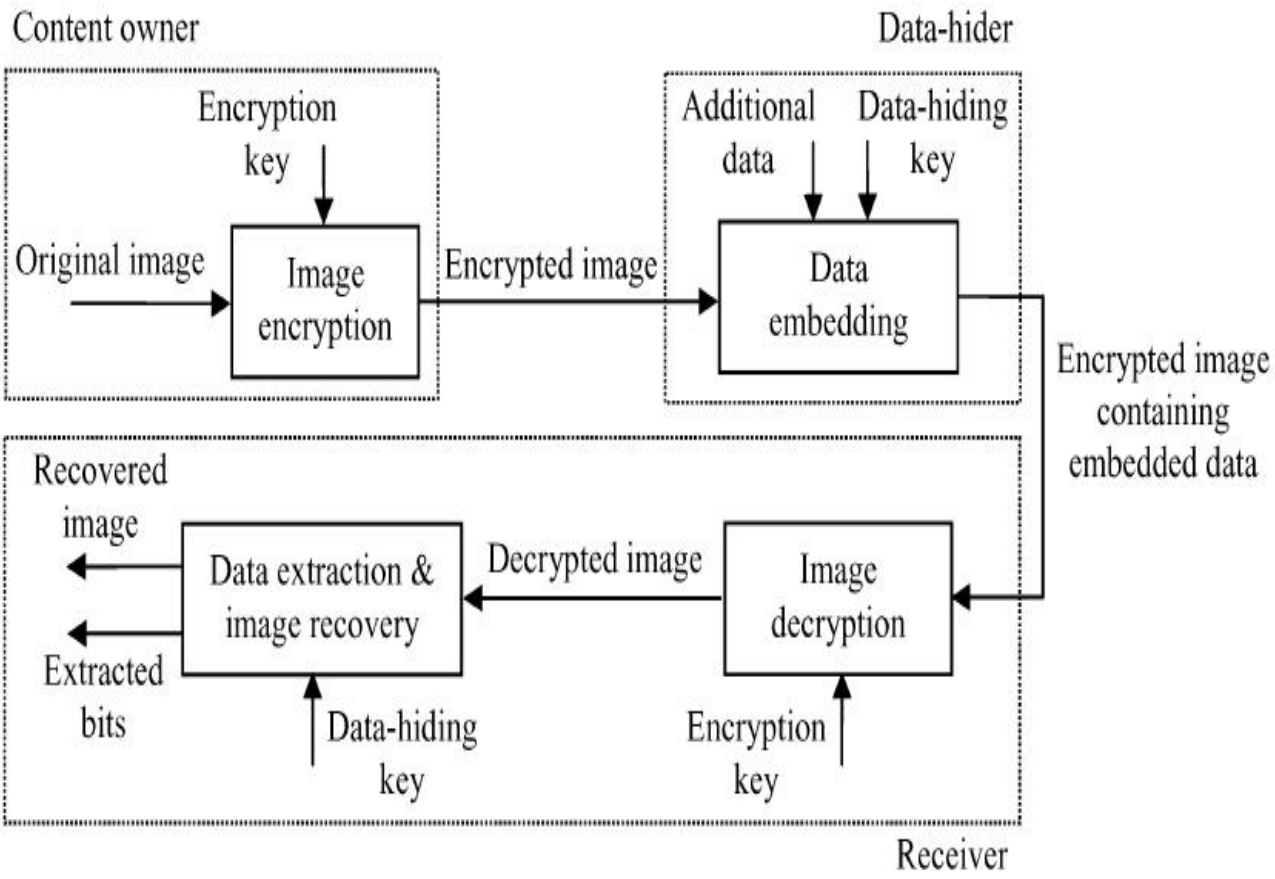


Fig.5 OVERALL DFD

3.6 UML DIAGRAMS

3.6.1 USECASE DIAGRAM

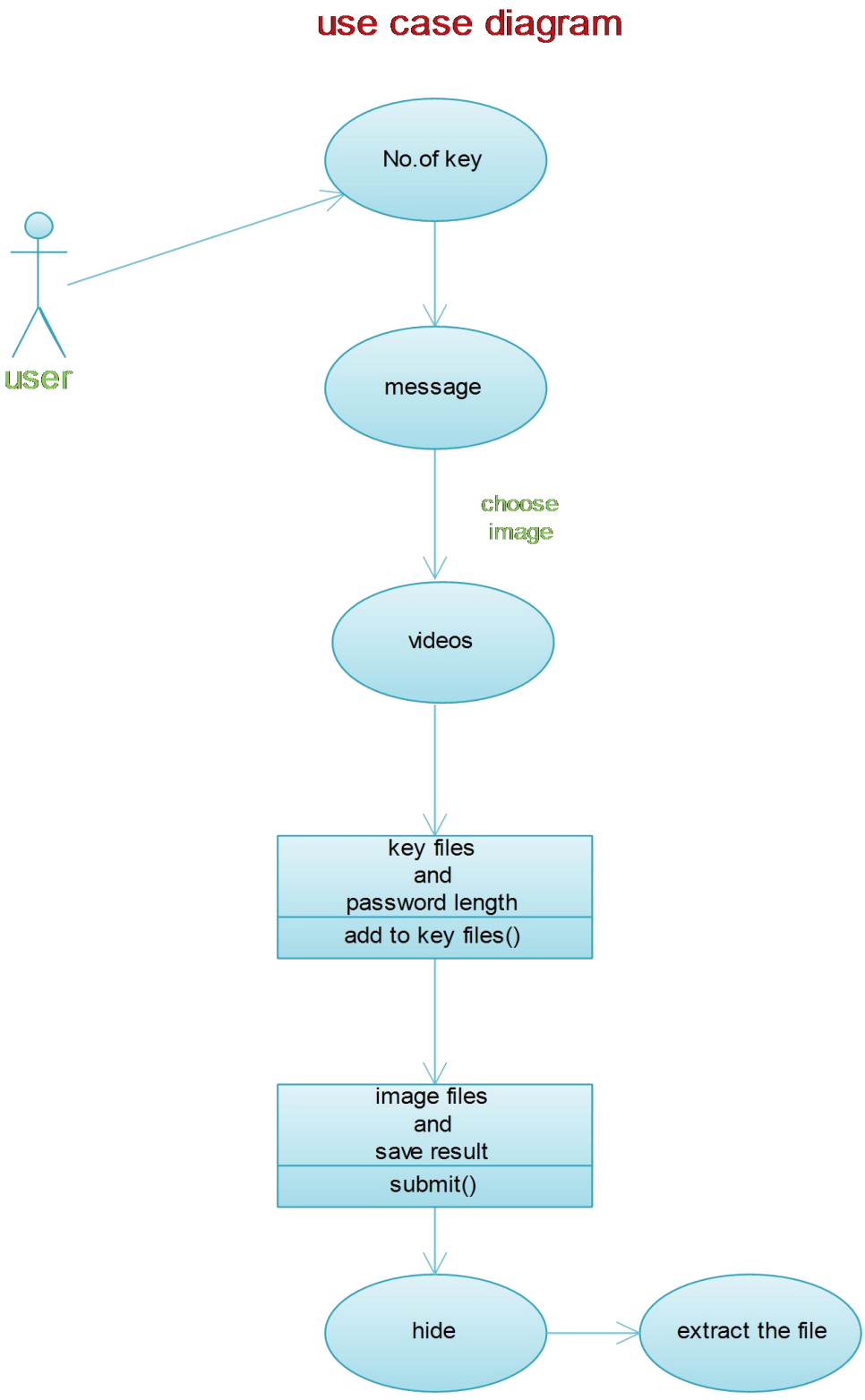


Fig.6 USECASE DIAGRAM

A use case diagram is a visual representation of the interactions between a system and its users or external systems. It's a fundamental tool in the Unified Modeling Language (UML) for capturing system requirements and functionality. User represents the person who interacts with the system to extract a file. Extract File is the main functionality where the user initiates the process of extracting a file from the database.Key Selection system might prompt the user to enter a key or identifier to specify the file they want to extract. This key could be a filename, ID number, or any unique identifier associated with the file in the database.Password Length Check system might enforce a minimum password length to ensure a strong password is used when accessing the database. This is likely for security purposes.Verify Key and Password validates the key or identifier entered by the user along with the password. This ensures the user has proper access rights to extract the file.If the key and password are valid, the system locates and extracts the requested file from the database.Show File/Hide section likely refers to displaying the extracted file to the user or hiding it for security reasons. The specific behavior depends on the system's design.

3.6.2 CLASS DIAGRAM

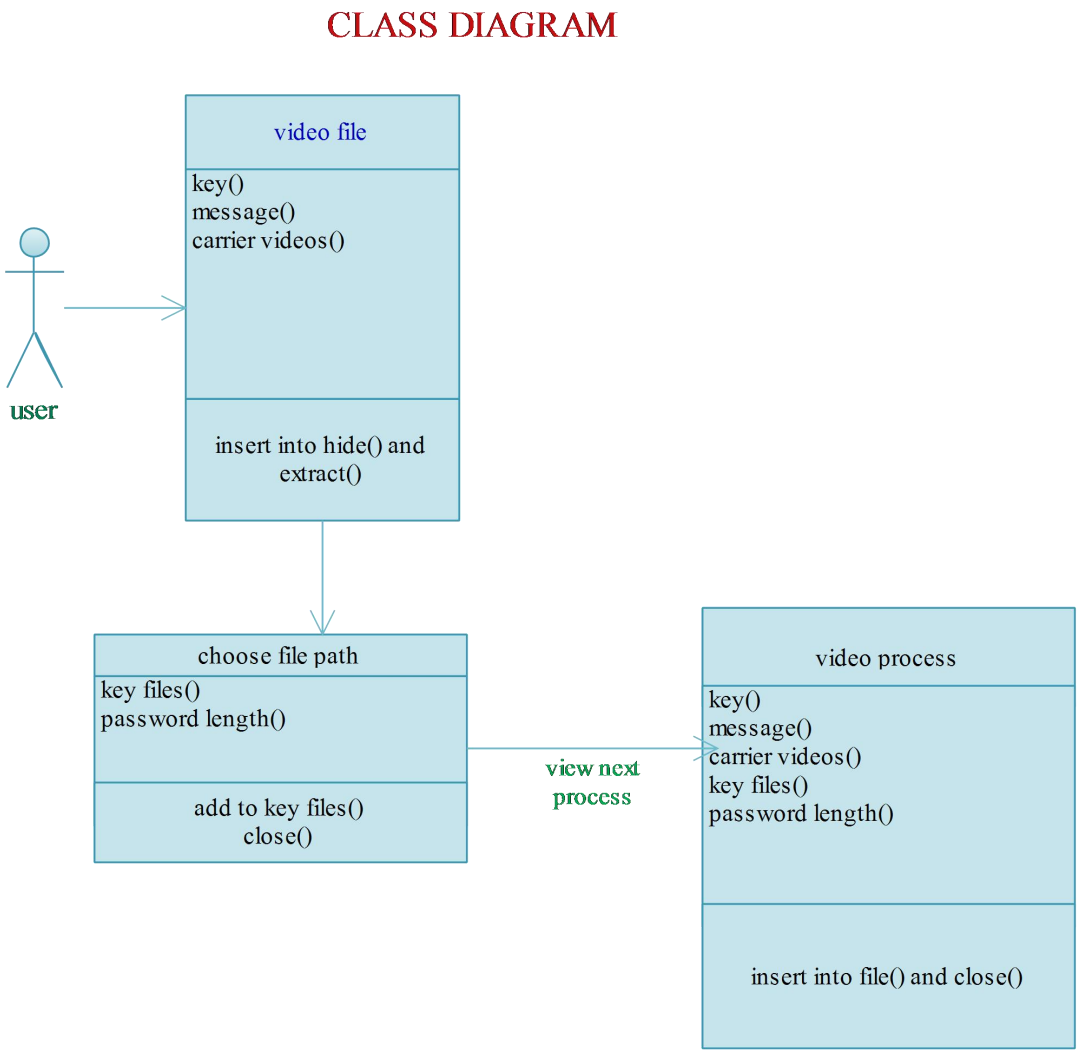


Fig.7 CLASS DIAGRAM

A class diagram is a type of structural diagram used in the Unified Modeling Language (UML) to visually represent the classes, attributes, methods, and relationships within a software system. It acts as a blueprint, capturing the static structure of the system and showing how its various components interact.VideoFile class likely represents a video file on the storage device. It might contain attributes like filename, path, and video format. KeyFile class likely represents a key file used for encryption or decryption purposes. It might contain attributes like filename, path, and password length. VideoProcess class seems to be responsible for processing the video data. It might have methods for tasks like insert into file and close (possibly inserting processed video data into a new file and

closing it extract from file and close (possibly extracting video data from a file and closing it choose file path (selecting the path of the video file for processing)TECT class name is unclear without context. It might be an abbreviation for a specific video processing technology or library. It has methods for carrier videos (possibly related to extracting or manipulating carrier video streams)key files (possibly related to using key files in the video processing tasks) videos (possibly a general method for handling video data) VideoProcess uses VideoFile (the processing class interacts with video files).VideoProcess uses KeyFile (the processing class interacts with key files, possibly for encryption/decryption).Association might be an association between VideoFile and KeyFile, indicating that a video file might have an associated key file for secure storage or transmission.

3.6.3 ACTIVITY DIAGRAM

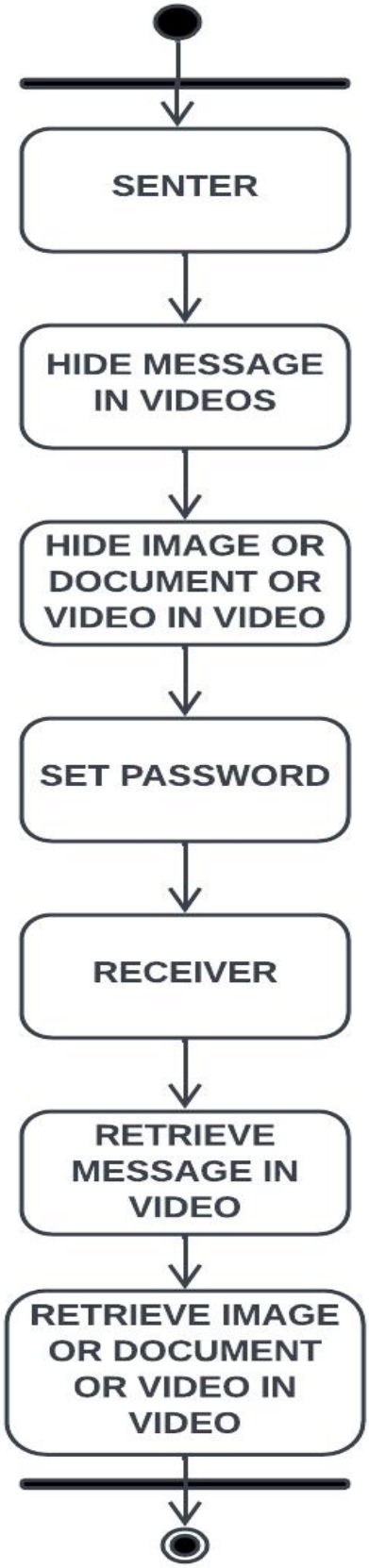


Fig.8 ACTIVITY DIAGRAM

The activity diagram you sent me depicts a process for hiding a message inside video. **Sender**-Hide image or document or video in video: This step suggests that the sender can hide various kinds of data within a video file.Set Password: A password is likely required to decrypt or reveal the hidden message later. **Receiver**-Retrieve message in video: The receiver can use the password to extract the hidden message from the video. Retrieve image or document or video in video: Similarly, the receiver can use the password to retrieve a hidden image, document or another video from the host video. In conclusion, this activity diagram outlines a method for concealing data such as images, documents and even other videos inside a video file. It also demonstrates how the recipient retrieves the hidden information using a password.

3.6.4 SEQUENCE DIAGRAM

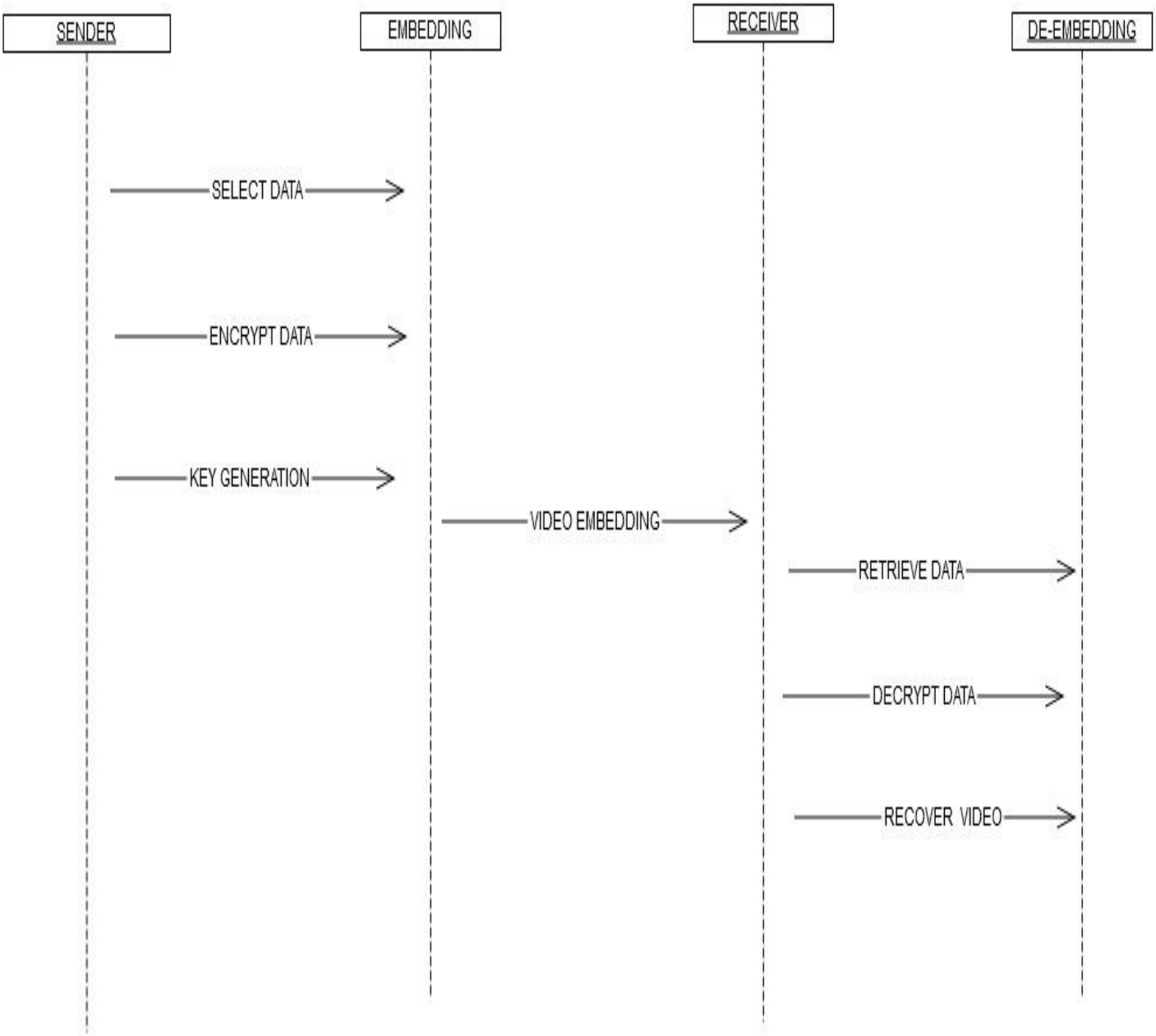


Fig.9 SEQUENCE DIAGRAM

A sequence diagram, also known as an event diagram or interaction diagram, is a type of Unified Modeling Language (UML) diagram specifically designed to visualize the interactions between objects in a system over time. It shows the sequential order of messages exchanged between objects participating in specific interaction or scenario. Top Toolbar section likely contains menus and buttons for common video editing

functions like File, Edit, and Effects. Project Panel might show a hierarchical view of your video project, including video clips, audio tracks, and other elements.Preview Window is a large window likely displays a preview of your edited video, allowing you to see the effects of your edits in real-time. Timeline section represents the timeline of your video project. It shows the video and audio tracks laid out sequentially, allowing you to arrange clips, add transitions, and make precise edits.

3.7 HARDWARE ENVIRONMENT

- Processor – Pentium III
- RAM – 4GB
- Hard Disk – 260GB

3.8 SOFTWARE ENVIRONMENT

- Operation System – 7/8/10
- Front-end - Java
- Back-end - MySql
- Tool - Netbeans 7.3.1

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 MODULE DESIGN SPECIFICATION

There are five modules:

- Slicing
- Motion Compensation
- Motion vector prediction
- Block Transformation and Encoding
- Macro block Ordering

4.1.1 SLICING

In general, a coded picture is divided into one or more slices. Slices are self-contained and can be decoded and displayed independently of other slices. Hence, intraprediction of DCT coefficients and coding parameters of a macro block is restricted to previous macro blocks within the same slice. This feature is important to suppress error propagation within a picture due to the nature of variable length coding. In regular encoding, when FMO is not used, slices contain a sequence of macro blocks in raster scan order. However, FMO allows the encoder to create what is known as slice groups. Each slice group contains one or more slices and macro blocks can be assigned in any order to these slices. The assignment of macro blocks to different groups is signaled by a syntax structure called the “slice group id”.

Slice types:

H.264 defines five different slice types: I, P, B, SI and SP.

I slices or “Intra” slices describe a full still image, containing only references to itself. A video stream may consist only of I slices, but this is typically not used. However, the first frame of a sequence always needs to be built out of I slices.

P slices or “Predicted” slices use one or more recently decoded slices as a reference (or “prediction”). The prediction is usually not exactly the same as the actual picture content, so a “residual” may be added.

B slices or “Bi-Directional Predicted” slices work like P slices with the exception that former *and future* I or P slices (in playback order) may be used as reference pictures. For this to work, B slices must be decoded *after* the following I or P slice.

4.1.2 MOTION COMPENSATION

Since MPEG-1, motion compensation is a standard coding tool for video compression. Using motion compensation, motion between frames can be encoded in a very efficient manner. A typical P-type block copies an area of the last decoded frame into the current frame buffer to serve as a prediction. If this block is assigned a nonzero motion vector, the source area for this copy process will not be the same as the destination area. It will be moved by some pixels, allowing to accomodate for the motion of the object that occupies that block. Motion vectors need not be integer values: In H.264, motion vector precision is one-quarter pixel (oneeighth pixel in chroma). Interpolation is used to determine the intensity values at non-integer pixel positions. Additionally, motion vectors may point to regions outside of the image. In this case, edge pixels are repeated.

4.1.3 MOTION VECTOR PREDICTION

Because adjacent blocks tend to move in the same directions, the motion vectors are also encoded using prediction. When a block’s motion vector is encoded, the surrounding blocks’ motion vectors are used to estimate the current motion vector. Then, only the difference between this prediction and the actual vector is stored

4.1.4 BLOCK TRANSFORMATION AND ENCODING

The basic image encoding algorithm of H.264 uses a separable transformation. The mode of operation is similar to that of JPEG and MPEG, but the transformation used is not an 8x8 DCT, but an 4x4 integer transformation derived from the DCT. This transformation is very simple and fast; it can be computed using only additions/subtractions and binary shifts. It decomposes the image into its spacial

frequency components like the DCT, but due to its smaller size, it is not as prone to high frequency “mosquito” artifacts as its predecessors An image block B is transformed to $B0$ using the following formula. The necessary post-scaling step is integrated into quantization (see below) and therefore omitted:

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}$$

$$B' = MBM^T$$

The basic functionality of the H.264 image transformation process is as follows: For each block, the actual image data is subtracted from the prediction. The resulting residual is transformed. The coefficients of this transform are divided by a constant integer number. This procedure is called quantization; it is the only step in the whole encoding process that is actually lossy. The divisor used is called the quantization parameter; different quantization parameters are used for luma and chroma channels. The quantized coefficients are then read out from the 4x4 coefficient matrix into a single 16-element scan. This scan is then encoded using sophisticated (lossless) entropy coding. In the decoder, these steps are performed in reversed order.

4.1.5 MACROBLOCK ORDERING

In this paper, we make use of the explicit assignment of macroblocks to slice groups to hide messages in the video stream. Since macroblocks can be arbitrary assigned to slice groups, we propose to use the slice group ID of individual macroblocks as an indication of message bits. Assume for instance that two slice groups are used, the allocation of a macroblock to slice group 0 indicates a message bit of 0 and the allocation of macroblock to slice group 1 indicates a message bit of 1. Hence, one message bit per macroblock can be carried.

4.2 ALGORITHM

4.2.1 EMBEDDED Description

Content-Based Filtering Technique Algorithms analyze words, the occurrence of words, and the distribution of words and phrases inside the content of e-mails and segregate them into spam non-spam categories .Case Base Spam Filtering Method Algorithms trained on well-annotated ham/non-spam marked emails try to classify the incoming mails into two categories. Heuristic or Rule-Based ham Filtering Technique Algorithms use pre-defined rules in the form of a regular expression to give a score to the messages present in the e-mails. Based on the scores generated, they segregate emails into ham non-spam categories. The Previous Likeness Based Spam filtering Technique Algorithms extract the incoming mails' features and create a multidimensional space vector and draw points for every new instance. Based on the naïve bayes classifier algorithm, these new points get assigned to the closest class of ham and non-spam. Adaptive Spam Filtering Technique Algorithms classify the incoming mails in various groups and, based on the comparison scores of every group with the defined set of groups, ham and non-spam emails got segregated.

STEPS:

Step 1: Extract Bit set of Message, Bit={M0, M1,……, M65535 }

Step 2: The Pixels of cover image, Pixel = {pixel0, pixel,…, pixel65535}

Step 3: Extract LSB-1 set of the cover image, LSB1={A0, A1,…,A65535}.

Step 4: Extract LSB-2 set of the cover image, LSB2={B0, B1,…, B65535}.

Step 5: For i=1 to message length does

ALGORITHM :

For i=1

{

If Mi= =Bi

```
Then do nothing
 Else
{
If Mi= =1 and Bi= =0 Then 40
{
Bi=Mi;
Ai=0;
Pixel (i)-=1
}
 Else If Mi= =0 and Bi= =1 Then
 {
Bi=Mi;
Ai=1;
Pixel (i)+=1
 }
}
}
```

CRYPTOGRAPHY ALGORITHM

Cryptography is technique of securing information and communications through use of codes so that only those person for whom the information is intended caunderstand it and processit. Thus preventing unauthorized access to information. The prefix “crypt” means “hidden”and suffix “graphy” means “writing”. In Cryptography the techniques which are use toprotect information are obtained from mathematical concepts and a set of rule based calculations known as algorithms to convert messages in ways that make it hard to decode it.These algorithms are used for cryptographic key generation, digital signing, verification to protect data privacy, web browsing on internet and to protect confidential transactions such as credit card and debit card transactions.

CHAPTER 5
RESULTS & DISCUSSION

5.1 PERFORMANCE PARAMETERS/TESTING

5.1.1 TESTING OBJECTIVES

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding error, if it exists. The tests are inadequate to detect possibly present errors. The software more or less confirms to the quality and reliable standards

5.1.2 TESTING LEVELS

System testing is stage of implementation which is aimed at ensuring that the system works accurately and efficient before live operation commences. Testing is vital the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

Unit Testing

In the lines of strategy, all the individual functions and modules were put to the test independently. By following this strategy all the errors in coding were identified and corrected. This method was applied in combination with the White and Black Box testing Techniques to find the errors in each module.

Integration Testing

Data can be lost across the interface; one module can have an adverse effect on others. Integration testing is a systematic testing for constructing program structure. While at the same time conducting tests to uncover errors associated within the interface. Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order sets and conducted. The objective is to take unit tested modules and

combine them test it as a whole. Thus, in the integration-testing step all the errors uncovered are corrected for the next testing steps.

Validation Testing

The outputs that come out of the system are as a result of the inputs that go into the system. The correct and the expected outputs that go into the system should be correct and proper. So this testing is done to check if the inputs are correct and they are validated before it goes into the system for processing.

Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required. This is done in regard to the following point:

- Input screen design
- Output screen design

An acceptance test has the objective of selling the user on the validity and reliability of the system. It verifies that the system's procedures operate to system specifications and that the integrity of important data is maintained. Performance of an acceptance test is actually the user's show. User motivation is very important for the successful performance of the system. After that a comprehensive report is prepared.

CHAPTER 6
CONCLUSION & FUTURE ENHANCEMENT

CONCLUSION

As demonstrated by the experimental results, the developed steganalysis is able to detect the presence of watermarks. While the proposed steganalytic algorithm was specifically designed to target two watermarking algorithms, the main idea could possibly be applied on several other algorithms that embed watermarks by altering a specific statistic of a histogram of the vertex set of the model. The proposed watermarking algorithm is based on a discrete statistic of the histogram of radial coordinates, the difference in the height of adjacent bins. The experimental results demonstrate that it outperforms in terms of robustness against malicious watermark removal attacks and against steganalytic attacks, while at the same time it also offers some improvement in terms of embedding distortion. In the future, we will work to develop more advanced steganalytic techniques for detecting the presence of watermark messages embedded in 3D models, and parallelly, develop 3D watermarking/steganographic algorithms that not only have better anti-steganalytic behavior, but also offer improved robustness/distortion trade-offs.

FUTURE ENHANCEMENTS

Implementing the suggested features would significantly enhance your steganalysis and watermarking project for 3D models. Integration of machine learning boosts detection accuracy, while dynamic watermarking and adaptive embedding techniques improve security by making it harder for adversaries to detect or remove watermarks. Encryption ensures the confidentiality of embedded information, while multimodal watermarking strengthens security across different domains. Real-world testing, user-friendly interfaces, collaboration with industry partners, benchmarking, and comprehensive documentation further contribute to the project's robustness, adaptability, and effectiveness in addressing 3D model security challenges.

APPENDICES

6.1 SOURCE CODE

```
package Hiding;

/**
 * File Name = BackEndHandler.java  Version 2.0.0 Class Name = BackEndHandler
 *
 * Copyright (c) 2005 - vinoth
 *
 * @author vinoth
 * @date Saturday, March 12, 2005
 */
import enc.Sample;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.crypto.*;
import javax.crypto.spec.*;

public class BackEndHandler extends Thread {

    public static final short EMBED_MESSAGE = 0;
    public static final short EMBED_FILE = 1;
    public static final short RETRIEVE_MESSAGE = 2;
    public static final short RETRIEVE_FILE = 3;
    public static final short EDIT_MASTER = 4;
    private short operation;
    private WindowAdapter client;
```

```
    private JFileChooser fileChooser;
    private MyFileView fileView;
    private File masterFile, dataFile, outputFile;
    private int result, result2;

    public BackEndHandler(WindowAdapter client, short operation) {
        this.client = client;
        this.operation = operation;

        // Setup file chooser
        fileChooser = new JFileChooser("./");
        fileChooser.setFileSelectionMode(fileChooser.FILES_ONLY);
        fileChooser.setDialogType(fileChooser.CUSTOM_DIALOG);
        //MyFileView fileView = new MyFileView();
        //fileView.putIcon("jpg", new ImageIcon("Images/image.jpg"));
        //fileView.putIcon("gif", new ImageIcon("Images/image.jpg"));
        //fileView.putIcon("bmp", new ImageIcon("Images/image.jpg"));
        //fileChooser.setFileView(fileView);
        fileChooser.setAccessory(new FilePreviewer(fileChooser));

        // Create and set the file filter
        MyFileFilter filter1 = new MyFileFilter(new String[]{"bmp", "jpg", "gif", "tif"},
"Picture files");
        MyFileFilter filter2 = new MyFileFilter(new String[]{"mp3", "wav", "ram",
"wma"}, "Audio files");
        MyFileFilter filter3 = new MyFileFilter(new String[]{"mpg", "wmv", "dat"},
"Video files");
        fileChooser.addChoosableFileFilter(filter1);
        fileChooser.addChoosableFileFilter(filter2);
        fileChooser.addChoosableFileFilter(filter3);
```

```
}

public void run() {
    if (!chooseMasterFile()) {
        return;
    }

    if (operation == EMBED_MESSAGE || operation == EMBED_FILE) {
        if (!chooseOutputFile()) {
            return;
        }
    }

    if (operation == EMBED_FILE) {
        if (!chooseDataFile()) {
            return;
        }
    }

    SteganoInformation steg;
    switch (operation) {
        case EMBED_MESSAGE:
            new EmbedMessageGUI(this);
            break;
        case EMBED_FILE:
            new EmbedFileGUI(this);
            break;
        case RETRIEVE_MESSAGE:
            steg = new SteganoInformation(masterFile);
            if (steg.isEster()) {
```

```
                showEster(steg);
            } else if (!steg.isValid()) {
                JOptionPane.showMessageDialog(null, "File '" + masterFile.getName()
                        + "' does not contain any message or file\nembedded using
Seganograph 2.0.0 or later!", "Invalid Steganograph file!",
JOptionPane.WARNING_MESSAGE);
            } else {
                new PreRetrieveGUI(steg, PreRetrieveGUI.RETRIEVE_MESSAGE);
            }
            break;
        case RETRIEVE_FILE:
            steg = new SteganoInformation(masterFile);
            if (steg.isEster()) {
                showEster(steg);
            } else if (!steg.isValid()) {
                JOptionPane.showMessageDialog(null, "File '" + masterFile.getName()
                        + "' does not contain any message or file\nembedded using
Seganograph 2.0.0 or later!", "Invalid Steganograph file!",
JOptionPane.WARNING_MESSAGE);
            } else {
                new PreRetrieveGUI(steg, PreRetrieveGUI.RETRIEVE_FILE);
            }
        }
    }

    // Method for choosing input file
    public boolean chooseMasterFile() {
        int result;
        do {
            result = fileChooser.showDialog(null, "Select Master file");
```

```
            if (result == fileChooser.APPROVE_OPTION) {
                masterFile = fileChooser.getSelectedFile();
                if (masterFile.getName().contains(".avi")) {
                    if (!checkFileExistency(masterFile)) {
                        continue;
                    } else {
                        break;
                    }
                } else {
                    JOptionPane.showMessageDialog(fileChooser, "INVALID FILE
FORMAT(Choose .avi)");
                }
            }
        } while (result != fileChooser.CANCEL_OPTION);

        if (result == fileChooser.CANCEL_OPTION) {
            return false;
        } else {
            return true;
        }
    }

    // Method for choosing output file
    public boolean chooseOutputFile() {
        int result;
        do {
            File previousFile = fileChooser.getSelectedFile();
            result = fileChooser.showDialog(null, "Select output file");
            if (result == fileChooser.APPROVE_OPTION) {
                outputFile = fileChooser.getSelectedFile();
```

```
            if (outputFile.exists()) {
                result2 = JOptionPane.showConfirmDialog(null, "File " +
outputFile.getName() + " already exists!\nWould you like to OVERWRITE it?", "File
already exists!", JOptionPane.YES_NO_OPTION);
                if (result2 == JOptionPane.NO_OPTION) {
                    if (previousFile != null) {
                        fileChooser.setSelectedFile(previousFile);
                    }
                    continue;
                }
            }
            break;
        }
    } while (result != fileChooser.CANCEL_OPTION);

    if (result == fileChooser.CANCEL_OPTION) {
        return false;
    } else {
        return true;
    }
}

// Method for choosing data file
public boolean chooseDataFile() {
    do {
        result = fileChooser.showDialog(null, "Select Data file");
        if (result == fileChooser.APPROVE_OPTION) {
            dataFile = fileChooser.getSelectedFile();
            if (!checkFileExistency(dataFile)) {
                continue;
```

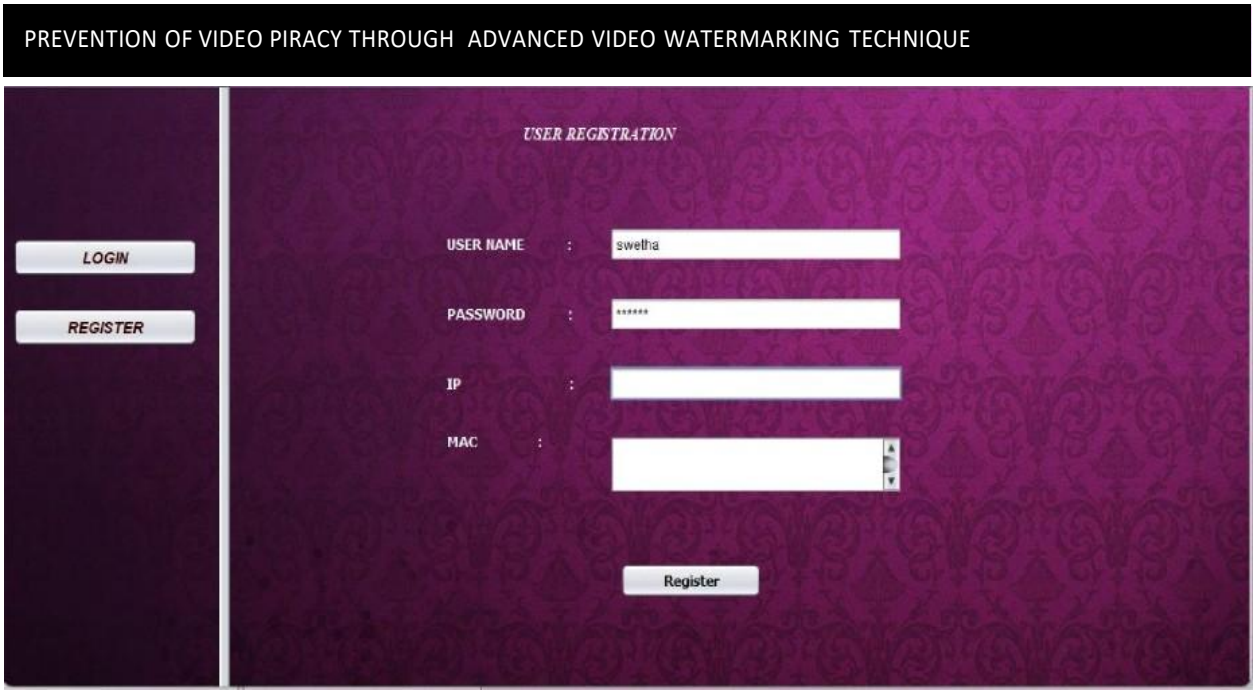
```
            } else {
                break;
            }
        }
    } while (result != fileChooser.CANCEL_OPTION);

    if (result == fileChooser.CANCEL_OPTION) {
        return false;
    } else {
        return true;
    }
}

// Accessor methods
public File getMasterFile() {
    return masterFile;
}

public File getOutputFile() {
    return outputFile;
}

public File getDataFile() {
    return dataFile;
}

// Mutator methods
public void setMasterFile(File file) {
    masterFile = file;
}
```

```
    public void setOutputFile(File file) {
        outputFile = file;
    }

    public void setDataFile(File file) {
        dataFile = file;
    }

    // Checks whether given file actually exists
    private boolean checkFileExistency(File file) {
        if (!file.exists()) {
            JOptionPane.showMessageDialog(null, "File " + file.getName() + " does not
exist!", "Inexistent file!", JOptionPane.ERROR_MESSAGE);
            return false;
        }

        return true;
    }

    private void showMessage(String message, String title) {
        JOptionPane.showMessageDialog(null, message, title,
JOptionPane.WARNING_MESSAGE);
    }

    private void showEster(SteganoInformation steg) {
        Object message[] = new Object[3];
        message[0] = new MyJLabel("This is an encrypted zone.", Color.red, Color.gray);
        message[1] = new JLabel("Please enter password to continue.");
        JPasswordField pass = new JPasswordField(10);
```

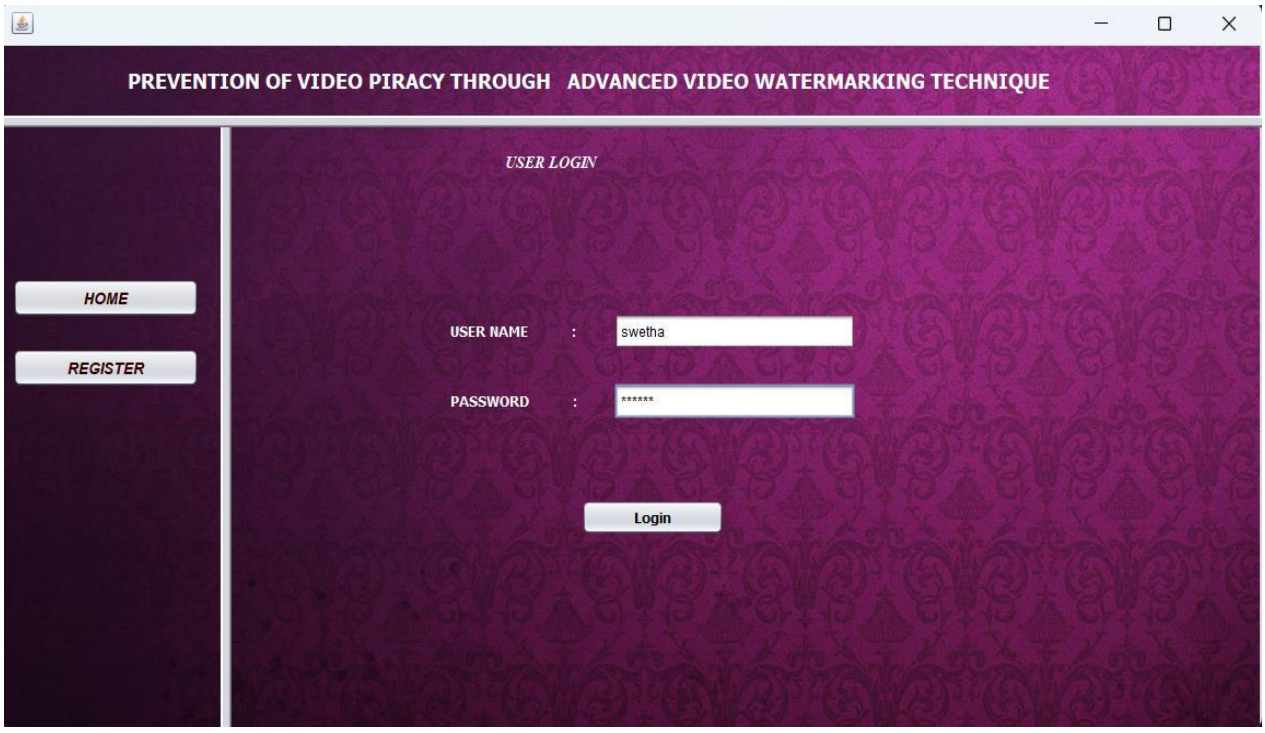
```
        message[2] = pass;

        String options[] = {"Retrieve Text", "Cancel"};
        int result = JOptionPane.showOptionDialog(null, message, "Encrypted zone",
JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null,
options, options[0]);

        if (result == 1) {
           return;
        }

        String password = new String(pass.getPassword());
        String PaS = new Sample().Retirve(steg.getFile().getName());
        System.out.println("--------------------------------------------------" + PaS);
        if (password.equals(PaS)) {
           if (password.length() < 8) {
              JOptionPane.showMessageDialog(null, "This was not the right password!",
"Invalid password", JOptionPane.OK_OPTION);
           } else {
              int fileSize = (int) steg.getFile().length();
              byte[] byteArray = new byte[fileSize];
              try {
                 DataInputStream in = new DataInputStream(new
FileInputStream(steg.getFile()));
                 in.read(byteArray, 0, fileSize);
                 in.close();

                 Cipher cipher = Cipher.getInstance("DES");
                 cipher.init(Cipher.DECRYPT_MODE, new
SecretKeySpec(password.substring(0, 8).getBytes(), "DES"));
```

```
                byteArray = cipher.doFinal(byteArray);
            } catch (Exception e) {
                return;
            }

            JFrame frame = new JFrame("Enjoy the ester egg...");
            frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
            frame.getContentPane().add(new JScrollPane(new JLabel(new
ImageIcon(byteArray))));
            frame.setBackground(Color.white);

            Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
            frame.setSize(d.width, d.height / 2);
            frame.setVisible(true);
          }
       } else {
          JOptionPane.showMessageDialog(null, "This was not the right password!",
"Invalid password", JOptionPane.OK_OPTION);
       }
    }

    public static void main(String[] arg) {

       BackEndHandler back = new BackEndHandler(null, EMBED_MESSAGE);
       back.start();
    }
}
```

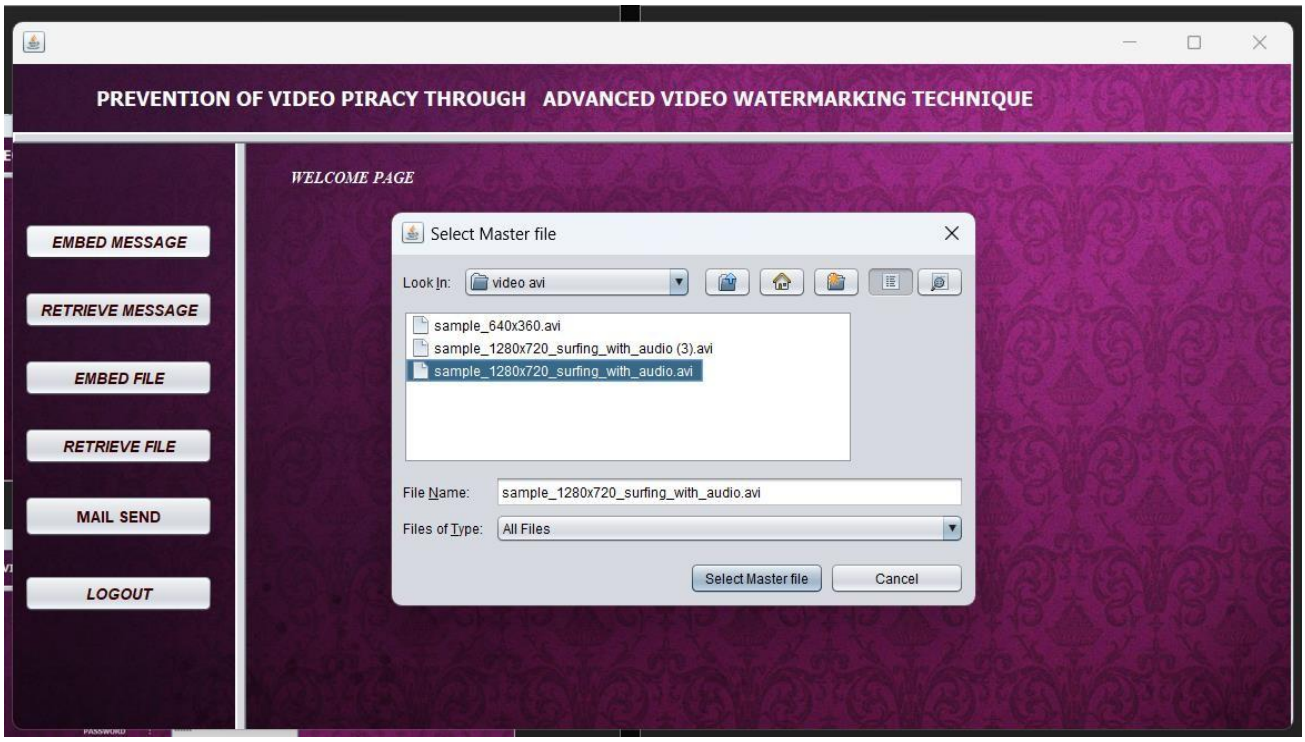
6.2 SCREENSHOTS



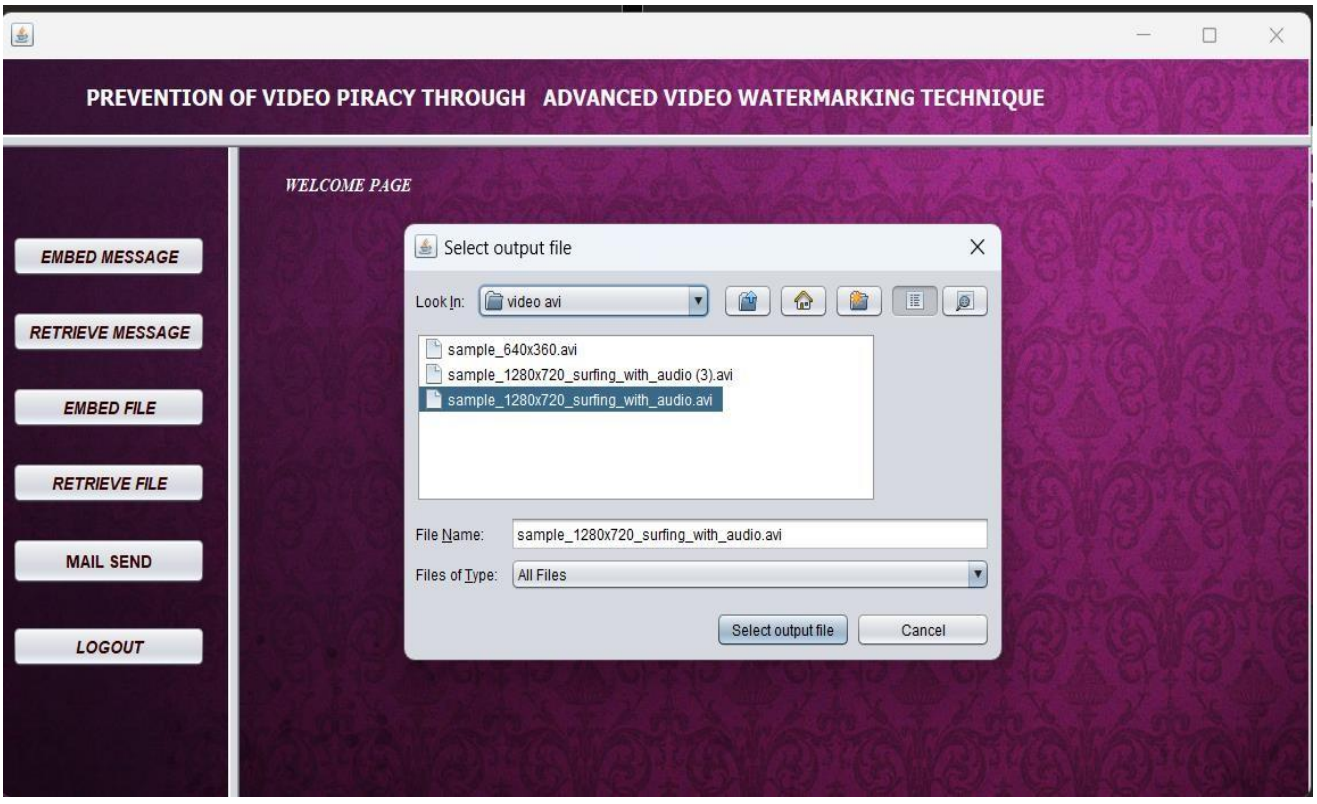
1. REGISTRATION PAGE



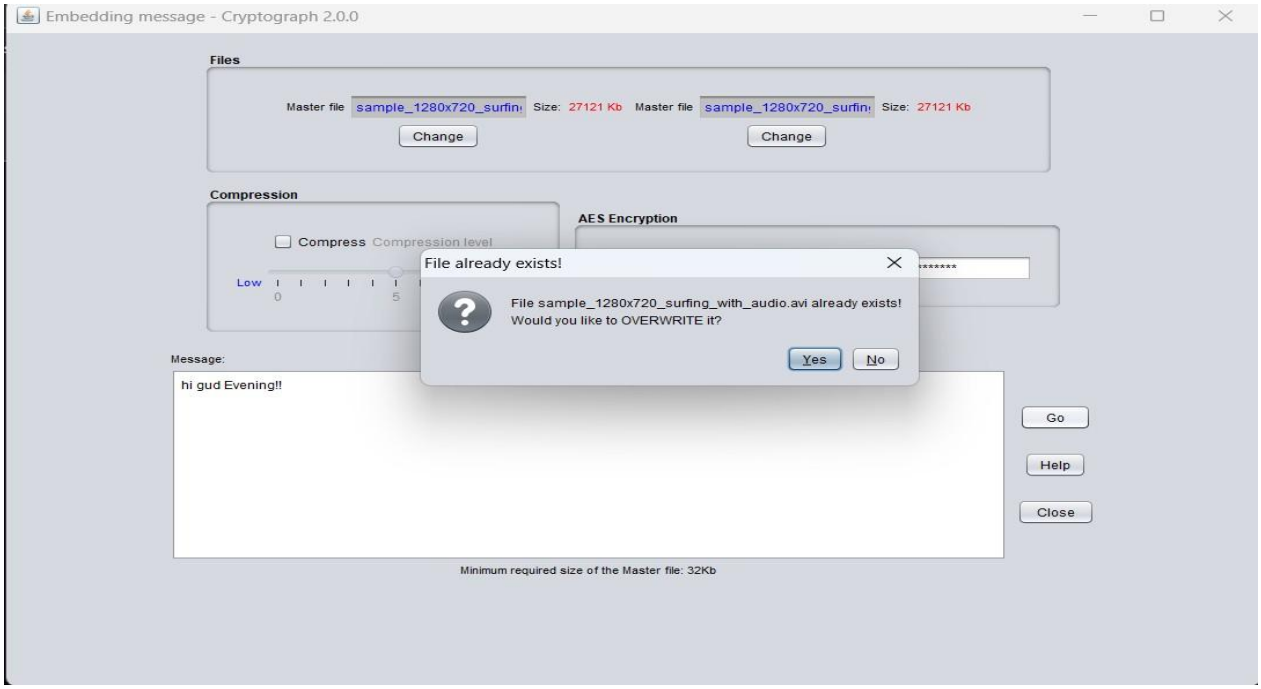
2. LOGIN PAGE



3. Select Master file

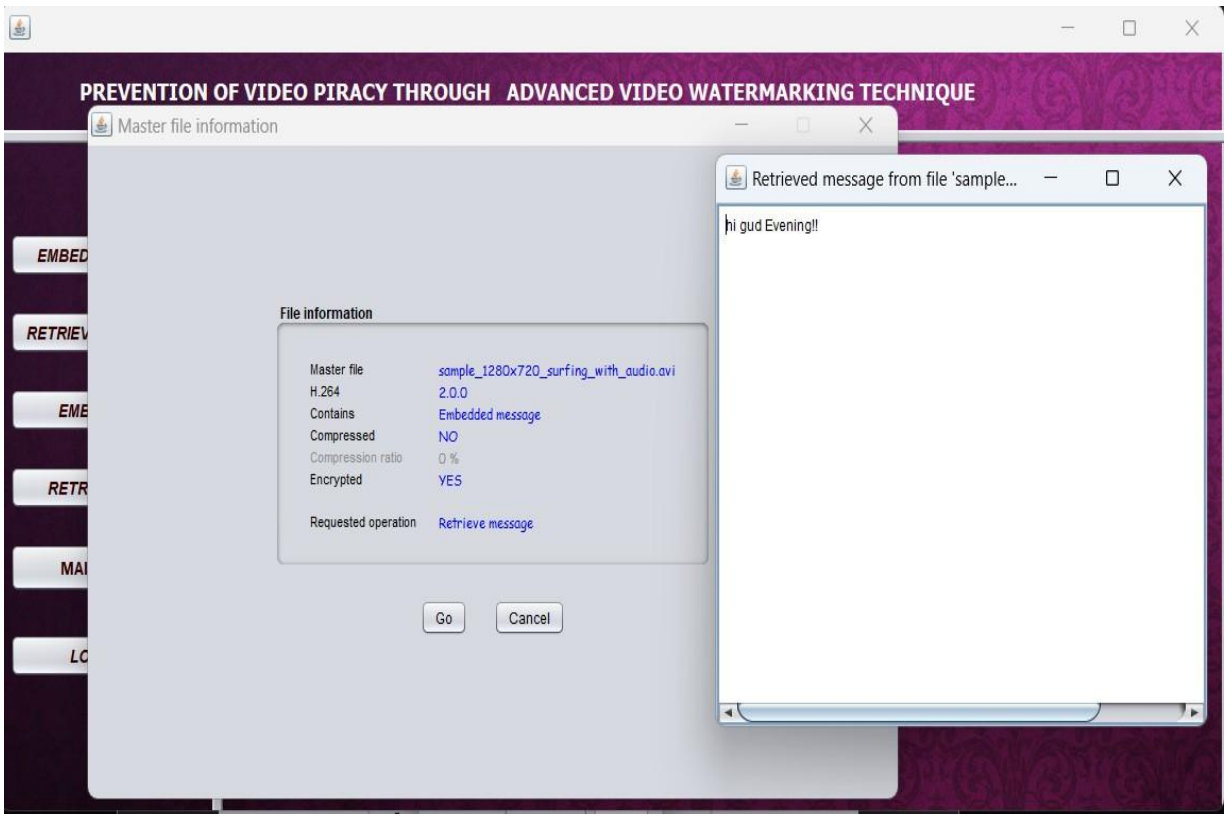


4. Select Output file

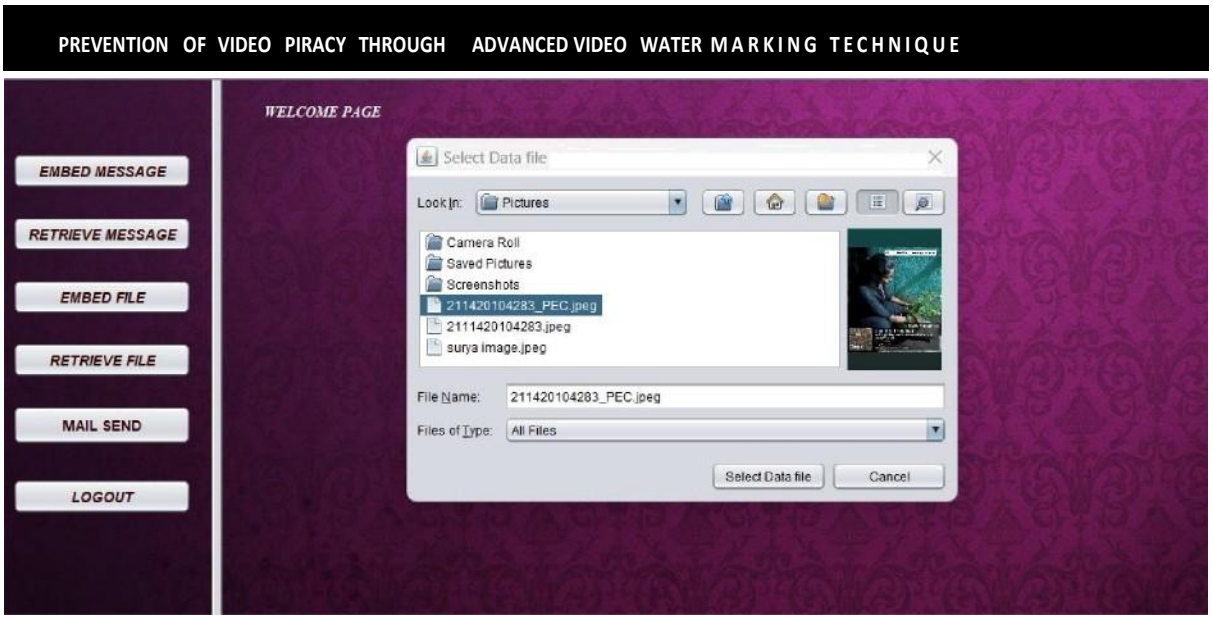


5.Hiding Message

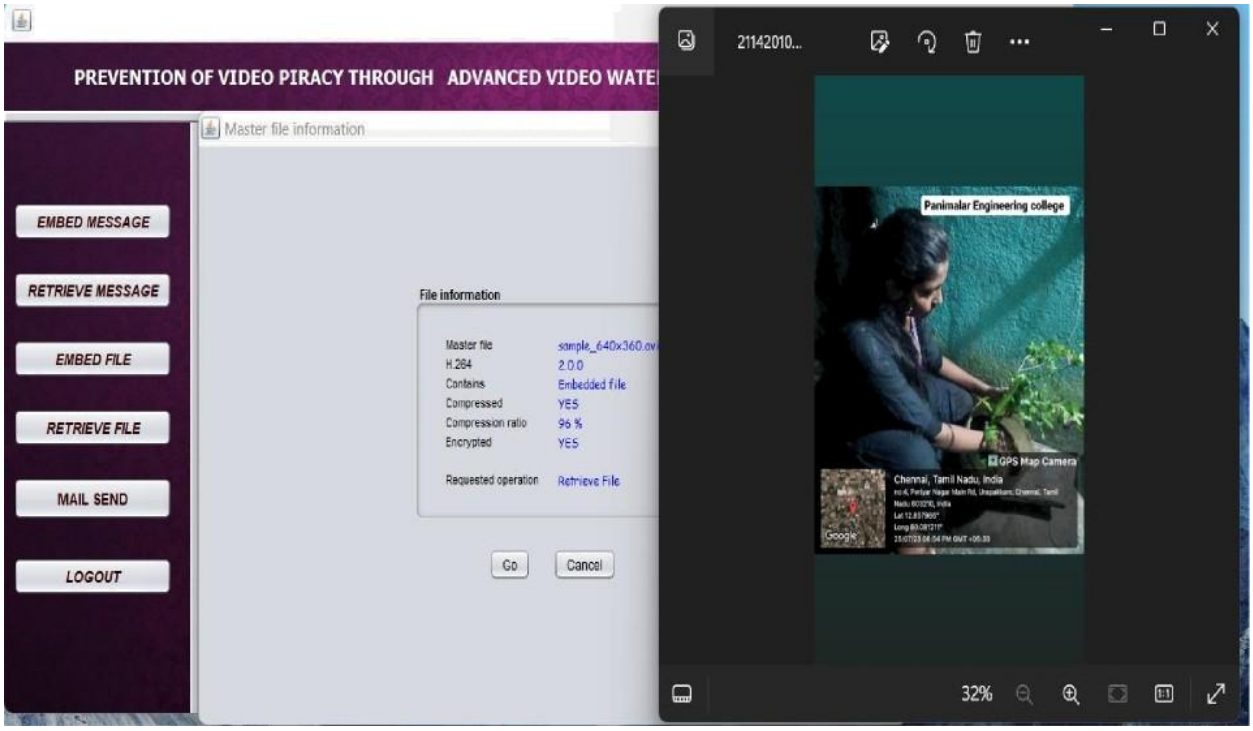
6.Enter the password



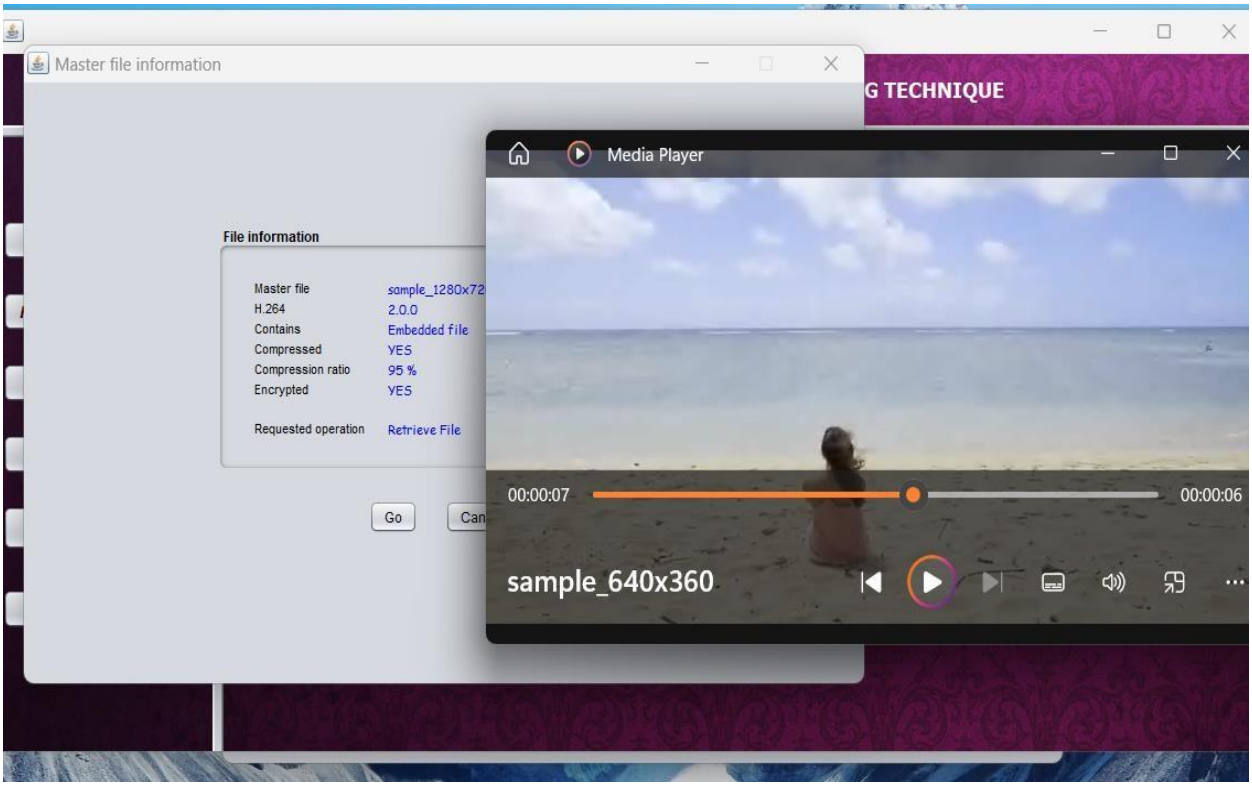
7.Display the Hidden Message



8.Hidding the Image file



9. Display the hidden image



10.Display the Hidden Video

6.3.PLAGIARISM REPORT

RE-2022-220872-plag-report

ORIGINALITY REPORT

9%	6%	8%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	mafiadoc.com Internet Source	2%
2	dro.dur.ac.uk Internet Source	1%
3	Osama F. AbdelWahab, Ashraf A.M. Khalaf, Aziza I. Hussein, Hesham F. A. Hamed. "Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques", IEEE Access, 2021 Publication	1%
4	www.researchgate.net Internet Source	1%
5	Submitted to Universiti Putra Malaysia Student Paper	1%

6.4 PAPER PUBLICATION

Title: A 3D Steganalytic Computation And Steganalysis Safe Watermarking

Conference: Twelve International Conference On Contemporary Engineering And Technology

Indexing: The conference proceedings will be indexed in Google Scholar, ensuring wide visibility and accessibility to the academic community.

Authors: Swetha R, Salini R, Preethi S, Priyanka G

Affiliation: Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India.

Conference Track: ICCET 2024

Conference Date: 23nd & 24rd March 2024

Abstract Submission Date: 20 February 2024

Paper Submission Date: 22 February 2024

REFERENCES

[1] Rafi Ullah , Sultan Daud Khan1, Mohib Ullah , (Member, Ieee), Fadi Al-Machot, And Habib Ullah, “Toward Authentication of Videos: Integer Transform Based Motion Vector Watermarking,” VOLUME 10 , PP- 75063 - 75073, July.2022.

[2] Yi Chen, Hongxia Wang, Kim-Kwang Raymond Choo, Senior Member, Peisong He, Zoran Salcic, Life Senior Member, Dali Kaafar, Xuyun Zhang , “DDCA: A Distortion Drift-Based Cost Assignment Method for Adaptive Video Steganography in the Transform Domain VOLUME 19 , PP- 2405 - 2420, February .2021.

[3] S. Soderi , And R. De Nicola, “6G Networks Physical Layer Security Using RGB Visible Light Communications, ”VOLUME 10,PP- 5482 - 5496, December 2021.

[4] MD. ASIKUZZAMAN , HANNES MAREEN , NOUR MOUSTAFA, KIM-KWANG RAYMOND CHOO , AND MARK R. PICKERING “Blind Camcording-Resistant Video Watermarking in the DTCWT and SVD Domain,” ”VOLUME 10, PP- 15681 - 15698, Jan 2022.

[5] OSAMA FOUAD ABDEL WAHAB ASHRAF A. M. KHALAF , AZIZA I. HUSSEIN, AND HESHAM F. A. HAMED, “Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques,” ”VOLUME 9, PP- 31805 - 31815, Feb 2022.

[6] M. Asikuzzaman and M. R. Pickering, ‘‘An overview of digital video watermarking,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 28, no. 9, pp. 2131–2153, Sep. 2018.

[7] S. D. Roy, X. Li, Y. Shoshan, A. Fish, and O. Yadid-Pecht, ‘‘Hard- ware implementation of a digital watermarking system for video authentication,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 2, pp. 289–301, Feb. 2013.

[8] C.-W. Tang and H.-M. Hang, ‘‘A feature-based robust digital image watermarking scheme,’’ IEEE Trans. Signal Process., vol. 51, no. 4, pp. 950–959, Apr. 2003

[9] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, ‘‘Lossless generalized-LSB data embedding,’’ IEEE Trans. Image Process., vol. 14, no. 2, pp. 253–266, Feb. 2005..

[10] Z. Yu and Z. Lin, ‘‘Scene change detection using motion vectors and DC components of prediction residual in H.264 compressed videos,’’ in Proc. 7th IEEE Conf. Ind. Electron. Appl. (ICIEA), Singapore, Jul. 2012, pp. 990–995