PROTECTING THE SECRECY OF VIDEO USING ADVANCED DATA HIDING TECHNIQUE

A PROJECT REPORT

Submitted by

JAYAPRIYA M (211419205079)

MONICA M (211419205108)

JENI PRECILLA M (211419205082)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

PANIMALAR ENGINEERING COLLEGE,POONAMALLEE

ANNA UNIVERSITY :CHENNAI 600 025

APRIL 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"PROTECTING THE SECRECY OF VIDEO USING ADVANCED DATA HIDING TECHNIQUE"** is the bonafide work of **"JAYAPRIYA M (211419205079), MONICA M (211419205108), JENI PRECILLA M (211419205082)"** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr. M. HELDA MERCY M.E., Ph.D.,	**Mrs. K. MUTHULAKSHMI M.TECH.,(Ph.D.,)**
	SUPERVISOR
HEAD OF THE DEPARTMENT	**ASSOCIATE PROFESSOR**
Department of Information Technology	Department of Information Technology
Panimalar Engineering College	Panimalar Engineering College
Poonamallee, Chennai - 600 123	Poonamallee, Chennai - 600 123

Submitted for the project and viva-voice examination held on ______________

SIGNATURE	**SIGNATURE**
INTERNAL EXAMINER	**EXTERNAL EXAMINER**

DECLARATION

I hereby declare that the project report entitled “**PROTECTING THE SECRECY OF VIDEO USING ADVANCED DATA HIDING TECHNIQUE**” which is being submitted in partial fulfilment of the requirement of the course leading to the award of the ‘Bachelor Of Technology in Information Technology’ in **Panimalar Engineering College, an Autonomous institution Affiliated to Anna university- Chennai**is the result of the project carried out by me under the guidance of **Mrs. K. MUTHULAKSHMI M.Tech.,(Ph.D.,) in the Department of Information Technology**. I further declared that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma or any other person.

JAYAPRIYA M

Date**:** **MONICA M**

Place**:** Chennai **JENI PRECILLA M**

It is certified that this project has been prepared and submitted under my guidance.

Date: **Mrs. K. MUTHULAKSHMI M.Tech.,(Ph.D.,)**

Place: Chennai (ASSOCIATE PROFESSOR/ IT)

ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion . We wish to express our sincere thanks to all those who were involved in the completion of this project.

Our sincere thanks to **Our Honorable Secretary and Correspondent, Dr. P. CHINNADURAI, M.A., Ph.D.,** for his sincere endeavor in educating us in his premier institution.

We would like to express our deep gratitude to **Our Dynamic Directors,Mrs. C. VIJAYA RAJESHWARI and Dr. C. SAKTHI KUMAR, M.E.,M.B.A.,Ph.D.,**and**Dr.SARANYASREESAKTHIKUMAR.,B.E.,M.B.A.,Ph.D .,**for providing us with the necessary facilities for completion of this project.

We also express our appreciation and gratefulness to **Our Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in the completion of the project. We wish to convey our thanks and gratitude to our head of the department, **Dr. M. HELDA MERCY,M.E., Ph.D.,** Department of Information Technology, for her support and by providing us ample time to complete our project.

We express our indebtedness and gratitude to our Project co-ordinator **Mr. M. DILLI BABU**,**M.E.,(Ph.D.,)**Associate Professor, Department of Information Technology for his guidance throughout the course of our project. We also express sincere thanks to our supervisor **Mrs.K.MUTHULAKSHMI, M.Tech.,(Ph.D.,)**, for providing the support to carry out the project successfully. Last, we thank our parents and friends for providing their extensive moral support and encouragement during the course of the project.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	**VIII**
	LIST OF FIGURES	**IX**
	LIST OF ABBRIVIATION	**X**
1	**INTRODUCTION**	**1**
	1.1 SCOPE OF THE PROJECT	1
	1.2 OBJECTIVE OF THE PROJECT	2
	1.3 CRYPTOGRAPHY	2
	1.3.1 FEATURES OF CRYPTOGRAPHY	3
	1.3.2 TYPES OF CRYPTOGRAPHY	3
	1.3.3 APPLICATION OF CRYPTOGRAPHY	4
	1.3.4 CHALLENGES OF CRYPTOGRAPHY	6
	1.3.5 CRYPTOGRAPHY KEY	7
2	**LITERATURE SURVEY**	**8**
	2.1 LITERATURE SURVEY	8
	2.11 FEASIBILITY STUDY	16
3	**SYSTEM DESIGN**	**18**
	3.1 EXISTING SYSTEM	18
	3.2 PROPOSED SYSTEM	19
	3.3 ARCHITECTURE DIAGRAM	20
	3.4 DATAFLOW DIAGRAM	21
	3.4.1 USECASE DIAGRAM	21
	3.4.2 CLASS DIAGRAM	22
	3.4.3 ACTIVITY DIAGRAM	23
	3.4.4 SEQUENCE DIAGRAM	24
	3.4.5 E-R DIAGRAM	25
	3.4.6 DFD DIAGRAM	26
	3.4.6.1 DFD LEVEL 0	26
	3.4.6.2 DFD LEVEL 1	27
	3.4.6.3 DFD LEVEL 2	27
	3.4.6.4 OVERALL DFD	28

	3.5 MODULES	29
	3.5.1 MODULE I	29
	3.5.1.1 AES ENCRYPTION	29
	3.5.1.2 ASCII TO BINARY CONVERSION	29
	3.5.2 MODULE II	29
	3.5.2.1 SLICING	30
	3.5.2.2 MOTION COMPENSATION	31
	3.5.2.3 MOTION VECTOR PREDICTION	31
	3.5.2.4 BLOCK TRANSFORMATION AND ENCODING	31
	3.5.2.5 MACRO BLOCK ORDERING	32
4	**REQUIREMENT SPECIFICATION**	**33**
	4.1 HARDWARE REQUIREMENTS	33
	4.2 SOFTWARE REQUIREMENTS	33
	4.2.1 JAVA	33
	4.2.1.1 JAVA PLATFORM	34
	4.2.2 AUTOMATIC MEMORY MANAGEMENT	34
	4.2.3 MYSQL	36
	4.2.3.1 USES OF MYSQL	37
	4.2.3.2 PLATFORM AND INTERFACES	37
	4.2.3.3 MANAGEMENT AND GRAPHICAL FRONTENDS	37
	4.2.3.4 OFFICIAL	38
	4.2.3.5 DEPLOYMENT	39
	4.2.3.6 FEATURES	40
	4.2.3.7 SUPPORT AND LICENSING	44
5	**IMPLEMENTATION**	**46**
	5.1 PERFORMANCE	47
	5.2 EMBEDDING ALGORITHM	47
	5.3 AES ALGORITHM	48
	5.4 SAMPLE CODE	49
	5.5 SAMPLE SCREEN SHOTS	86
6	**TESTING AND MAINTENANCE**	**92**
	6.1 SOFTWARE TESTING	92
	6.2 DEVELOPING METHODOLOGIES	92
	6.3 TYPES OF TESTING	92

	6.3.1 UNIT TEST	92
	6.3.2 FUNCTIONAL TEST	93
	6.3.3 SYSTEM TEST	94
	6.3.4 PERFORMANCE TEST	94
	6.3.5 INTEGRATION TEST	94
	6.3.6 ACCEPTANCE TEST	94
7	**CONCLUSION**	**96**
	7.1 CONCLUSION	96
	7.2 FUTURE ENCHANCEMENT	96
	REFERENCE	**97**

ABSTRACT

This project "**PROTECTING THE SECRECY OF VIDEO USING ADVANCED DATA HIDING TECHNIQUE"** is used to secretly hide a data inside the AVI video file.

We propose a advanced video hiding technique to protect the data which need to be secure so that the piracy and the security of the data is maintained without falling into wrong hands. The message or data such as text, audio, video and files are embedded into a video .The video is of Audio Video Interleave (AVI) format and the data is embedded using h.264 algorithm, slicing and macro block technique. The video is secured using encryption and decryption using Advanced Encryption Standard (AES) algorithm. The sender secures the video using a key, the receiver can retrieve the data only when the key is entered thus by protecting the data. This method is used to transfer information securely by hiding the data in a video, the third party doesn't know that there is a data hidden in the video, it look just like a normal video without any traces of the data hidden inside. It includes slicing technique and macro blocking in Encoding and Decoding process for hiding message and extracting message respectively. In this technique cryptography which performs Encoding and Decoding. First of all text will be embedded within the video by using the slicing technique Motion vector prediction. This AVI format video file is again applied to cryptography tool to decode embedded data. There is use of following algorithm for data hiding.

LIST OF FIGURES

FIGURE NO.	NAMEOF THE FIGURE	PAGE NO.
4.1	ARCHITECTURE DIAGRAM	20
3.3.1	USECASE DIAGRAM	21
3.3.2	CLASS DIAGRAM	22
3.3.3	ACTIVITY DIAGRAM	23
3.3.4	SEQUENCE DIAGRAM	24
3.3.5	E-R DIAGRAM	25
3.3.6.1	DFD LEVEL 0	26
3.3.6.2	DFD LEVEL 1	27
3.3.6.3	DFD LEVEL 2	27
3.3.6.4	OVERALL DFD	28
5.5.1	LOGIN PAGE	86
5.5.2	SELECTING MASTER FILE	87
5.5.3	SELECTING OUTPUT VIDEO	87
5.5.4	TEXT MESSAGE TO BE HIDDEN	88
5.5.5	RETRIEVE TEXT MESSAGE	89
5.5.6	HIDDEN MESSAGE IS DISPLAYED	89
5.5.7	ENTER KEY	90
5.5.8	HIDDEN IMAGE	90
5.5.9	OUTPUT	91

LIST OF ABBRIVIATION

AVI	Audio Video Interleave
AES	Advanced Encryption Standard
DFD	Data Flow Diagram
ERD	Entity relationship diagram
AVC	Advanced Video Coding
DES	Data Encryption Standard
DCT	Discrete Cosine Transform
GPL	General public License
SQL	Structured Query Language
JVM	Java Virtual Machine

INTRODUCTION

Nowadays the want for safety is turning into greater vital due to improved safety requirements, statistics protection is in the main furnished with statistics hiding. It is an encrypted area barring decryption preserves the confidentiality of the content.

In addition, it is more efficient besides decryption accompanied by way of records hiding and re-encryption. In the novel scheme of facts hiding without delay in the encrypted model of H.264/AVC video stream, divided into three parts, which is H.264/AVC video encryption, information embedding, and records extraction with the assist of statistics hiding code phrase technique. The main server can manipulate the video or affirm its integrity barring understanding the authentic contents such as Text, Word document, Image, Audio file and therefore to provide security and protection. A person can cover records and can also embed extra statistics in the encrypted area with the aid of the use of substitution approach named code word. In order to adapt to exceptional software scenarios, information extraction can be finished both in the encrypted area and in the decrypted for we using cryptography scheme and slicing method for using splitting content to hide data in video and macro block technique used for add frame by frame data into video file measurement is strictly preserved even after encryption and statistics embedding.

1.1 SCOPE OF THE PROJECT

Multi party content hiding has become an important demand for network media to protect all parties rights. A cryptography scheme in Encrypted data with hiding video secure multiple content hiding in video master file on lightweight cryptography is proposed.

Additive secret sharing and block- level scrambling are developed to generate the encrypted text and embedded file, image, audio into master file video. The cryptography based on significant bit Prediction Expansion is performed by secure video text embedded file technique.

1.2 OBJECTIVE

A robust method of imperceptible audio, video, text and image hiding is proposed. The motion vector technique is found as the better solution since it hides the data in the moving objects. The most secure and cryptography algorithm is introduced. Here a more secure and effective hash based algorithm that uses a pure hash technique for coding and decoding the information in video hiding.

1.3 CRYPTOGRAPHY

Cryptography is technique of securing information and communications through use of codes so that only those persons for whom the information is intended can understand it and process it. Thus preventing unauthorized access to the information. The prefix “crypt” means “hidden” and suffix graphy means “writing”. In Cryptography the techniques which are use to protect 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.

Techniques used For Cryptography: In today’s age of computers cryptography is often associated with the process where an ordinary plain text is converted to cipher text which is the text made such that intended receiver of the text can only decode it and hence this process is known as encryption. The process of conversion of cipher text to plain text this is known as decryption.

1.3.1 FEATURES OF CRYPTOGRAPHY

- **Confidentiality:** Information can only be accessed by the person for whom it is intended and no other person except him can access it.
- **Integrity:** Information cannot be modified in storage or transition between sender and intended receiver without any addition to information being detected.
- **Non-repudiation:** The creator/sender of information cannot deny his intention to send information at later stage.
- **Authentication:** The identities of sender and receiver are confirmed. As well as destination/origin of information is confirmed.

1.3.2 TYPES OF CRYPTOGRAPHY

- **Symmetric Key Cryptography:** It is an encryption system where the sender and receiver of message use a single common key to encrypt and decrypt messages. Symmetric Key Systems are faster and simpler but the problem is that sender and receiver have to somehow exchange key in a secure manner. The most popular symmetric key cryptography system is Data Encryption System (DES).
- **Hash Functions:** There is no usage of any key in this algorithm. A hash value with fixed length is calculated as per the plain text which makes it impossible for contents of plain text to be recovered. Many operating systems use hash functions to encrypt passwords.

- **Asymmetric Key Cryptography:** Under this system a pair of keys is used to encrypt and decrypt information. A public key is used for encryption and a private key is used for decryption. Public key and Private Key are different. Even if the public key is known by everyone the intended receiver can only decode it because he alone knows the private key.

1.3.3 APPLICATIONS OF CRYPTOGRAPHY

- **Computer passwords:** Cryptography is widely utilized in computer security, particularly when creating and maintaining passwords. When a user logs in, their password is hashed and compared to the hash that was previously stored. Passwords are hashed and encrypted before being stored. In this technique, the passwords are encrypted so that even if a hacker gains access to the password database, they cannot read the passwords.
- **Digital Currencies:** To safeguard transactions and prevent fraud, digital currencies like Bitcoin also use cryptography. Complex algorithms and cryptographic keys are used to safeguard transactions, making it nearly hard to tamper with or forge the transactions.
- **Secure web browsing:** Online browsing security is provided by the use of cryptography, which shields users from eavesdropping and man-in-the-middle assaults. Public key cryptography is used by the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols to encrypt data sent between the web server and the client, establishing a secure channel for communication.
- **Electronic signatures:** Electronic signatures serve as the digital equivalent of a handwritten signature and are used to sign documents. Digital signatures are created using cryptography and can be validated using public

key cryptography. In many nations, electronic signatures are enforceable by law, and their use is expanding quickly.

- **Authentication:** Cryptography is used for authentication in many different situations, such as when accessing a bank account, logging into a computer, or using a secure network. Cryptographic methods are employed by authentication protocols to confirm the user's identity and confirm that they have the required access rights to the resource.
- **Cryptocurrencies:** Cryptography is heavily used by cryptocurrencies like Bitcoin and Ethereum to safeguard transactions, thwart fraud, and maintain the network's integrity. Complex algorithms and cryptographic keys are used to safeguard transactions, making it nearly hard to tamper with or forge the transactions.
- **End-to-End Encryption:** End-to-end encryption is used to protect two-way communications like video conversations, instant messages, and email. Even if the message is encrypted, it assures that only the intended receivers can read the message. End-to-end encryption is widely used in communication apps like WhatsApp and Signal, and it provides a high level of security and privacy for users.

Advantages

- **Access Control:** Cryptography can be used for access control to ensure that only parties with the proper permissions have access to a resource. Only those with the correct decryption key can access the resource thanks to encryption.

- **Secure Communication:** For secure online communication, cryptography is crucial. It offers secure mechanisms for transmitting private information like passwords, bank account numbers, and other sensitive data over the internet.
- **Protection against attacks:** Cryptography aids in the defense against various types of assaults, including replay and man-in-the-middle attacks. It offers strategies for spotting and stopping these assaults.
- **Compliance with legal requirements:** Cryptography can assist firms in meeting a variety of legal requirements, including data protection and privacy legislation.

1.3.4 CHALLENGES OF CRYPTOGRAPHY

While cryptography is a powerful tool for securing information, it also presents several challenges, including:

- **Key management**: Cryptography relies on the use of keys, which must be managed carefully to maintain the security of the communication.
- **Quantum computing:** The development of quantum computing poses a potential threat to current cryptographic algorithms, which may become vulnerable to attacks.
- **Human error:** Cryptography is only as strong as its weakest link, and human error can easily compromise the security of a communication.

1.3.5 CRYPTOGRAPHY KEY

Modern cryptography key techniques are increasingly advanced and often even considered unbreakable. However, as more entities rely on cryptography to protect communications and data, it is vital to keep keys secure. One compromise key could result in regulatory action, fines and punishments, requirements, reputational damage, and the loss of customers and investors.

Attack Types

- Weak keys
- Incorrect use of keys
- Reuse of keys
- Non rotation of keys
- Inappropriate storage of keys
- Inadequate protection of keys
- Insecure movement of keys

LITERATURE SURVEY

2.1 Separable reversible data hiding in an encrypted image using the Adjacency pixel difference histogram

YEAR: **2023**

AUTHOR: Anushiadevi, R., & Amirtharajan, R.

Information security is a practice of encrypting data in movement and on hold, improving discretion and integrity. One can protect, encrypt and decrypt critical data in several ways. One of them is reversible data hiding in an encrypted image. The technique enables the user to encrypt images that need authentication and restore them to their original form of images. This technique returns a lossless image as an output making it the most suitable for medical images and the military. Histogram shifting of pixel difference is an effectual reversible data hiding method in information security. Each image's pixel is encrypted when a user wants to safely store a digital image in an open environment like a cloud. The authentication or any other relevant information related to that image is embedded in the pixel differencehistogram. The proposed approach's advantage is that the grayscale image transfer is carried out exceedingly safely, with near-zero correlation and Entropy closer to 8. The Peak Signal Noise Ratio (PSNR) for a directly decrypted image with an embedding capacity of 0.0807 bpp is 50.84 dB. Moreover, the secret and cover images are retrieved without error.

2.2 Flexible patch moving modes for pixel-value-ordering based reversible Data hiding methods

YEAR: 2023

AUTHOR: Fan, G., Pan, Z., Zhou, Q., & Zhang, X.

Pixel-value-ordering (PVO) is one of the most popular frameworks in the research of reversible data hiding (RDH) in recent years. In most PVO-based methods, the cover image is divided into non-overlapped blocks to embed secret data into the maximum and minimum pixels of each block. In these methods, the image division processing is just like moving a patch with constant step lengths in both horizontal and vertical directions. As a result, the block amount, as well as the embedding capacity (EC), is limited by the image size and the patch size. In this paper, we propose one-dimensional and two-dimensional flexible patch moving (FPM) modes to move the patch with adaptive step lengths. After enlarging the EC by employing FPM, an average-difference-based complexity computation is also proposed to cooperate with FPM to further improve the embedding performance. Finally, multiple pairwise embedding schemes are adaptively applied on blocks with different complexities to enhance the embedding performance once again. Experimental results illustrate that the proposed FPM modes work well on many PVO-based methods. Moreover, the proposed method achieves an obvious improvement in fidelity when compared with some state-of-the-art RDH schemes.

2.3 Secure Data Hiding and Extraction Using RSA Algorithm

YEAR: 2023

AUTHOR: Muazu, A. A., Maiwada, U. D., Garba, A. R. I., Qabasiyu, M. G., &Danyaro, K. U.

Data hiding and extraction is an important aspect of information security because the data will be safer and more manageable. Data hiding technology that works well produces data that is efficient, secure, and simple to connect. The underlying communication network that enables the transport of sensitive data is insecure and unprotected. The fast rise of electronic methods of communication implies that information security has become a critical concern in the real world. As such, anyone with the necessary expertise and software may eavesdrop and intercept data transmissions, which can be extremely harmful and even life threatening in so many cases. Therefore, the research of this paper offers a software framework that allows users to create messages and hide them within image file documents that can be sent to a desired recipient. Moreover, we conducted an experiment with different data that guarantee the increase level of confidentiality of information exchange over the internet.

2.4 A secure data hiding approach based on least-significant-bit and nature inspired optimization techniques

YEAR: 2022

AUTHOR: Hameed, M. A., Abdel-Aleem, O. A., & Hassaballah, M.

With remarkable informationtechnology development, information security has become a major concern in the communication environment, where security must be performed for the multimedia messages exchanged between the sender and the intended recipient. Digital multimedia steganography techniques have been developed to attain a security for covert communication and secure data. This paper proposes an approach for image steganography using the Least Significant Bit Substitution (LSB) and Nature-Inspired Harris Hawks Optimization (HHO) algorithm for efficient concealing of the secret data inside a cover image; thus providing high confidentiality. The HO based data encoding operation uses the PSNR visual quality metric as an obiective function. The obiective function is used to determine the ideal encoding vector to convert the secret message to its encoded form. The proposed approach performs better than other state-of-the-art methods in terms of standard measures of visual quality with maintaining high embedding capacity.Comparisons with existing LSB or multi-directionalPVD embedding methods demonstrate that the proposed method has more optimized and higher embedding capacity with maintaining visual quality.Besides, the proposed approach achieves high security against statistical StegoExpose analysis, ALASKA2 deep learning steganalysis, and image processing attacks.

2.5 Improving data hiding within color images using hue component of HSV color space

YEAR: 2022

AUTHOR: Hassan, F. S., & Gutub, A.

Data hiding technologies aim to hide the existence of secret information within digital covers such as images by causing unnoticeable degradation to their quality. Reducing the image distortion and increasing the embedding capacity are the main points that the data hiding techniques revolved around. This article proposes two high payload embedding methods with high stego image quality using the Hue-Saturation-Value (HSV) colour model. The first method is hue-based embedding (HBE) that employs the *H* plane for hiding one or two bits in non-grey pixels. The second method uses the three HSV components, so it is called three-planes embedding (TPE). In TPE, one bit is hidden in the least significant bit (LSB) of *V* of the grey pixels, one or two bits in *H* of the pixels having low saturation or low brightness and one bit in the LSB of *S* otherwise. The experiments were conducted on 25 images and the results show that HBE hides more data on average than TPE with its quality reaching 60 dB. TPE achieves quality up to 61 dB and capacity reaches 364 Kb. TPE scores the highest capacity among six state-of-the-art techniques in Red-Green-Blue, HSV, Hue-Saturation-Intensity and YCbCr spaces with the highest average peak signal to noise ratio midst five of them. By embedding 60, 90, and 120 Kb, this TPE attains the best average quality amid all the methods.

2.6 Research on Digital Image Watermarking Algorithm Based on Scrambling and Singular Value Decomposition.

YEAR: 2022

AUTHOR: Lei Pei

At present, the digital image watermarking algorithm has not embedded the synchronization signal in the image, resulting in the poor performance of the embedded image in terms of, for example, imperceptibility, anti-attack ability, and robustness. Therefore, a digital image watermarking algorithm based on scrambling and singular value decomposition is proposed. &e digital watermark is preprocessed by dimensionality reduction and encryption; the digital image is processed in sections and embedded in the synchronization signal. &e digital watermark image is embedded in the digital image watermark by using the low-frequency energy ratio technology of sound channel. &e watermark image extraction step is designed to extract the watermark image. The transmittance of the degraded image is calculated, the transmittance is refined by relying on the soft matting algorithm and guiding filter, the image contrast in the image frequency domain is enhanced, and the results are mapped to the appropriate visual range to optimize the visual brightness of the image. Finally, according to the uniqueness of singular value matrix, the distributed characteristics of image matrix data are described, the visual effect is enhanced, and the research of digital image watermarking algorithm is completed. The experimental results show that the algorithm can extract the watermark information completely, the image contrast and singular value have been significantly improved, and the algorithm has better anti-attack performance.

2.7 Image Watermarking between Conventional and Learning-Based Techniques: A Literature Review.

YEAR: 2022

AUTHOR: Said Boujerfaoui ,RabiaRiad, Hassan Douzi , FrédéricRos and RachidHarba

Currently, most transactions and exchanges are conducted through the Internet thanks to technological tools, running the risk of the falsification and distortion of information. This is due to the massive demand for the virtual world and its easy access to anyone. Image watermarking has recently emerged as one of the most important areas for protecting content and enhancing durability and resistance to these kinds of attacks. However, there is currently no integrated technology able to repel all possible kinds of attacks; the main objective of each technology remains limited to specific types of applications, meaning there are multiple opportunities to contribute to the development of this field. Recently, the image watermarking field has gained significant benefits from the sudden popularity of deep learning and its outstanding success in the field of information security. Thus, in this article, we will describe the bridge by which the watermarking field has evolved from traditional technology to intelligent technologies based on deep learning.

2.8 Multiple Histograms Shifting-Based Video Data Hiding Using Compression Sensing.

YEAR: 2022

AUTHOR: Yanli Chen, Limengnan Zhou, Yonghui Zhou1 , Yi Chen , (Graduate Student Member, I Shengbo Hu, And Zhicheng Dong

With the development of multimedia editing technologies, the copyright protection has attacked more attentions. Reversible data hiding (RDH), in which the cover can be recovered losslessly, is an effect method to eliminate embedding distortions. As a typical RDH method, histogram shifting (HS) is used widely. Most existing RDH schemes based on HS usually build sharp histograms by predicting and sorting techniques. To make use of spatial correlations of multimedia, several RDH schemes based on multiple HS (MHS) are proposed to protect copyright, in which some rigid rules are used to build multiple histograms. Against images, videos have more spatial and temple correlations and it is easier to acquire sharper histograms. In this paper, a video MHS scheme based on compression sensing (CS) is proposed. As a linear sensing algorithm, CS can measure macroblock residuals by reducing corrections among pixels to acquire distinguishable macroblock features, while keeping their statistical characteristics immutable. By employing CS, macroblocks with similar characteristics cluster together to formulate multiple histograms. For each of these histograms, data embedding is implemented to reduce shifting distortions by expanding the outermost bins while other bins are unchanged. Experimental results show that the quality of most test videos in our scheme are higher than that in the state-of-art schemes.

2.9 Robust Unsupervised Video Anomaly Detection by Multipath Frame Prediction

YEAR: 2021

AUTHOR: Fabien A. P. Petitcolas, Ross J. Anderson, And Markus G. Kuhn

Video anomaly detection is commonly used in many applications, such as security surveillance, and is very challenging. A majority of recent video anomaly detection approaches utilize deep reconstruction models, but their performance is often suboptimal because of insufficient reconstruction error differences between normal and abnormal video frames in practice. Meanwhile, frame prediction-based anomaly detection methods have shown promising performance. In this article, we propose a novel and robust unsupervised video anomaly detection method by frame prediction with a proper design which is more in line with the characteristics of surveillance videos. The proposed method is equipped with a multipath ConvGRU-based frame prediction network that can better handle semantically informative objects and areas of different scales and capture spatial-temporal dependencies in normal videos.

2.10 Multipurpose Watermarking Algorithm for Medical Images.

YEAR: 2022

AUTHOR: Shaozhang Xiao, ZhengweiZhang ,Yue Zhang, and Changhui Yu

Considering the existing medical image watermarking algorithms, a single function often has limitations, and a multipurpose watermarking algorithm for medical images is proposed. First, medical images are divided into regions of interest (ROIs) and regions of noninterest (RONIs). ,en, the authentication watermark produced for each subblock of the ROI is embedded into the corresponding mapping subblock. , e visible watermark is embedded into the RONI, and, finally, the watermark information and constructed authentication information in each subblock of the ROI are embedded into the corresponding RONI subblock. Simulation results show that the embedded visible watermark can protect and facilitate medical image management. In addition, the proposed algorithm has strong robustness and very good visual quality. It can simultaneously realize copyright protection and content authentication and also has high tamper localization capability.

2.11 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

2.11.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.11.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.11.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM DESIGN

3.1 EXISTING SYSTEM

Creating an Image Using Encrypted Sensitive Words and Hiding over the image layer is based on encrypting the sensitive words over the image frames. It is very easy for the hackers to find the particular frame over the image watermarking, because of low frames are contently move over the image content. This system is more users friendly and flexible. Data can be used in any form less accurate result is produced. To send the large amount of data in image format compression option can be used to deliver the message. Large amount file Information hiding is also very difficult to sent data error recovery in image. The main disadvantage of the compressed image motion vectors, prediction modes are embedded into the image. When error occurs during transmission, the main components can be extracted to patch the lost or corrupted parts at reasonable perceptual quality. General data embedding refers to the principle idea of inserting information into a image for specific purpose to transfer data information to another user.

3.1.1 DISADVANTAGE

- In the content text is a plaintext domain results in the leakage of privacy.
- The visual quality of the marked image is not high.
- The schemes are fragile to noise attacks and image compression.
- The quality of image and text will be identified by hacker.

3.2 PROPOSED SYSTEM

Earlier various kinds of cryptography techniques are introduced for the video. Here we proposed the Hash Based Least Significant Bit Technique for Video cryptography which performs insertion of bits of text file and pdf file, audio, image file in video in the least significant bit position of RGB pixel as per hash function. In this way it includes slicing technique and macro blocking in Encoding and Decoding process for hiding message and extracting message respectively. In this technique cryptography which performs Encoding and Decoding. First of all text will be embedded within the video by using the slicing technique Motion vector prediction. This AVI format video file is again applied to cryptography tool to decode embedded data. There is use of following algorithm for data hiding.

3.2.1 ADVANTAGE

- In Military there is requirement to hide secret message from unintended receiver like terrorist.
- Businessman can use this technique to hide a data from business rivels.
- The observer has no idea that a secret data is hidden in a video. So there will be no chances to alter the data.
- We can hide data and image, audio, text file using cryptography.

3.3 ARCHITECTURE DIAGRAM

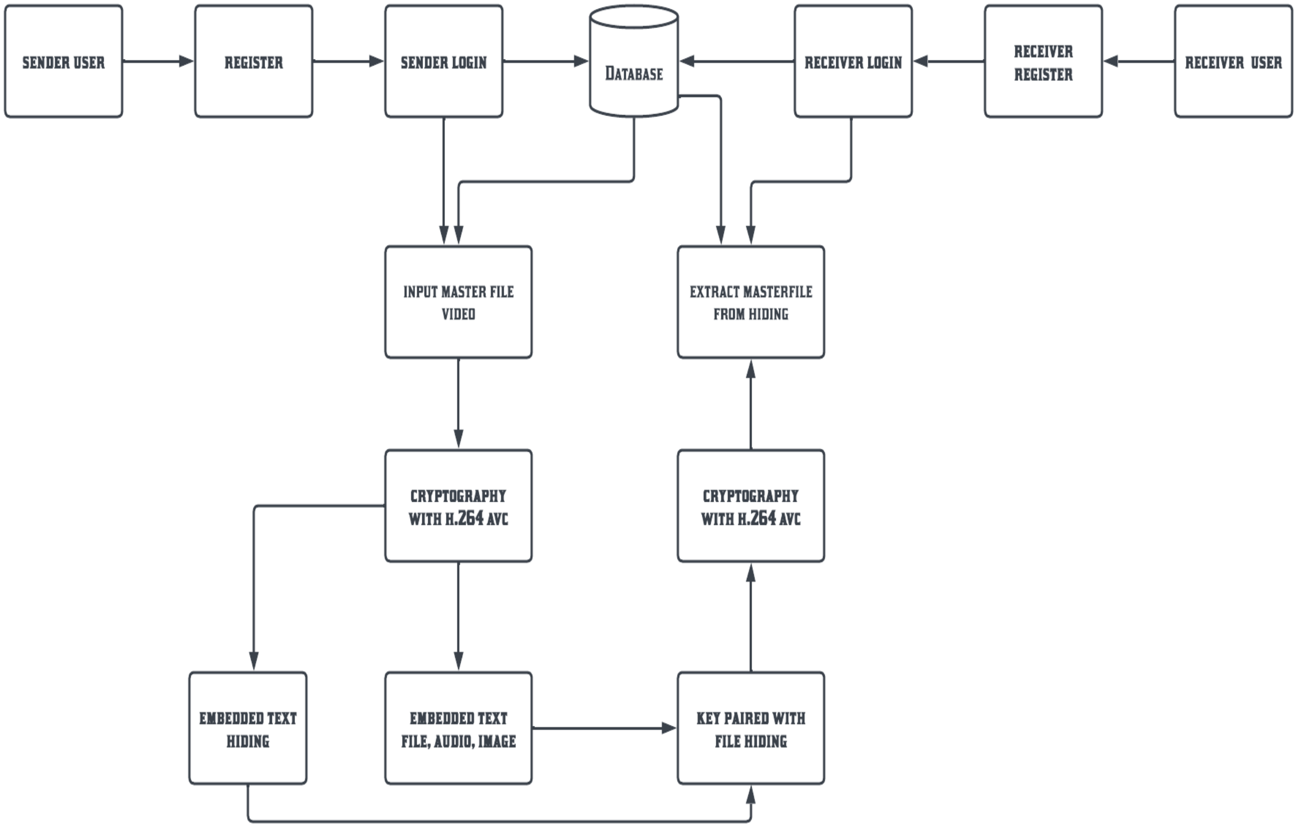

SENDER USER
REGISTER
SENDER LOGIN
DATABASE
RECEIVER LOGIN
RECEIVER REGISTER
RECEIVER USER
INPUT MASTER FILE VIDEO
EXTRACT MASTERFILE FROM HIDING
CRYPTOGRAPHY WITH H.264 AVC
CRYPTOGRAPHY WITH H.264 AVC
EMBEDDED TEXT HIDING
EMBEDDED TEXT FILE, AUDIO, IMAGE
KEY PAIRED WITH FILE HIDING

Fig 3.3 Architecture Diagram

The user can register when they are new user (sender), else can login using their credentials. The data which need to be hidden is selected and embedded in the input and output master file and this file is encrypted using AES algorithm and the message is hidden using cryptography with H.264 AVC. Then the hidden message is retrieved using the key with another user (receiver).

3.4 DATAFLOW DIAGRAM

3.4.1 USECASE DIAGRAM

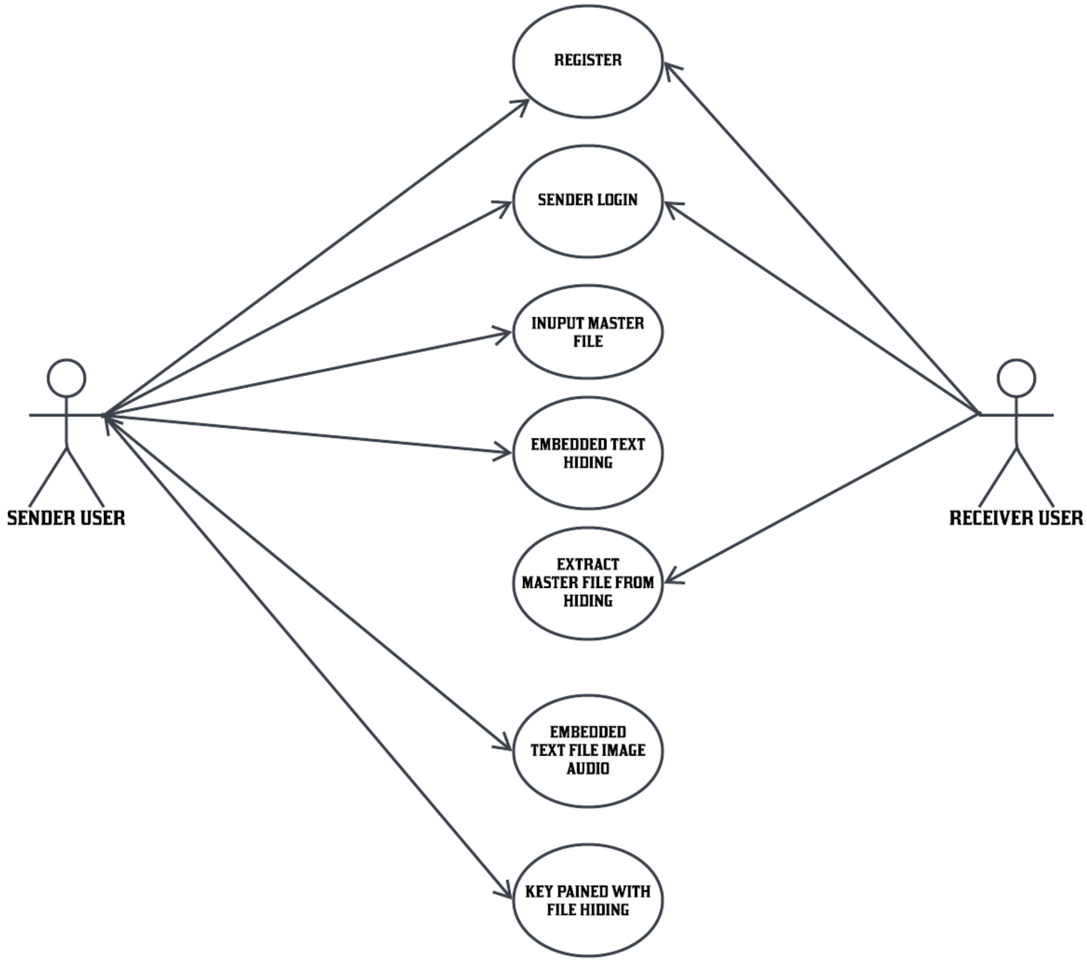

REGISTER
SENDER LOGIN
INUPUT MASTER FILE
EMBEDDED TEXT HIDING
EXTRACT MASTER FILE FROM HIDING
EMBEDDED TEXT FILE IMAGE AUDIO
KEY PAINED WITH FILE HIDING
SENDER USER
RECEIVER USER

Fig. 3.4.1 Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

3.4.2 CLASS DIAGRAM

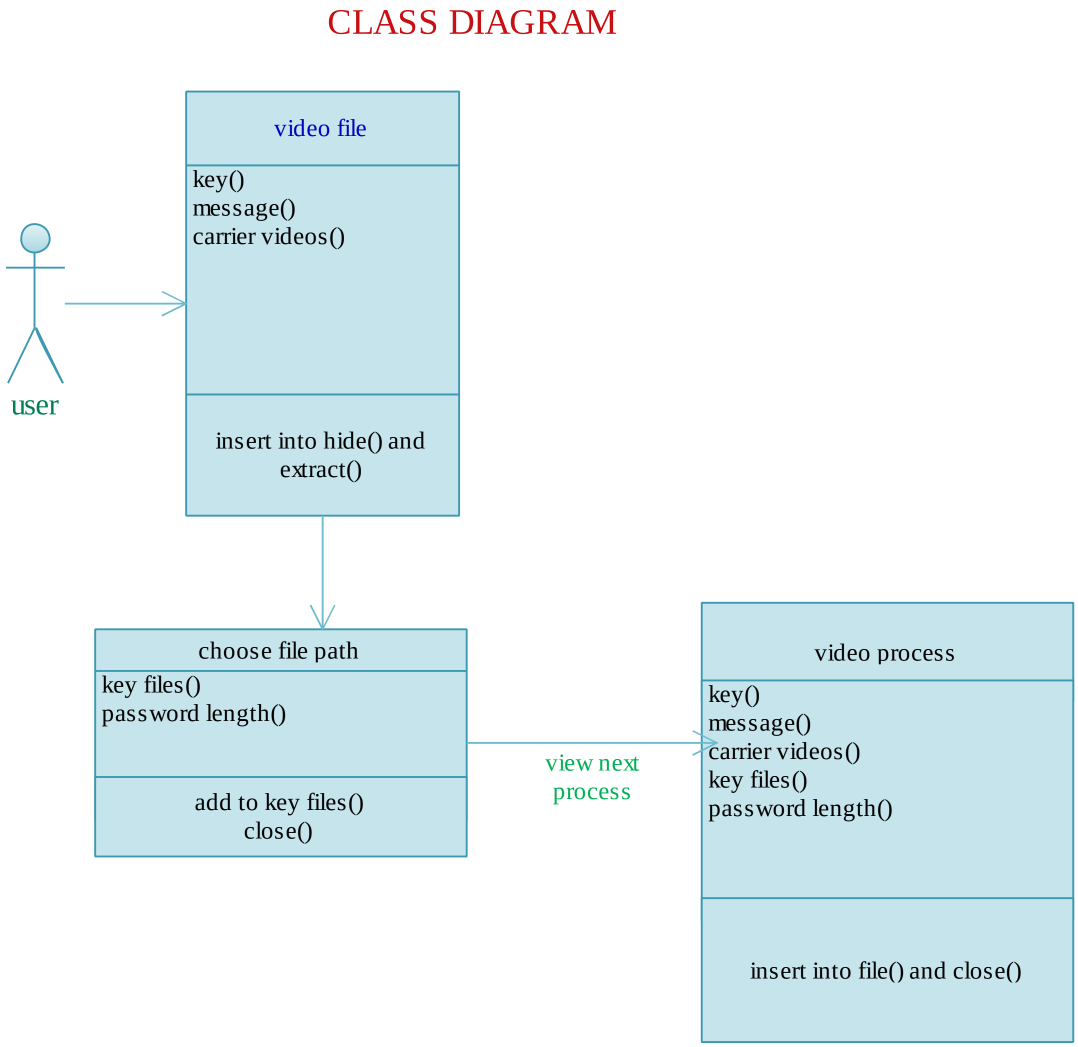


Fig. 3.4.2 Class Diagram

A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code. It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

3.4.3 ACTIVITY DIAGRAM

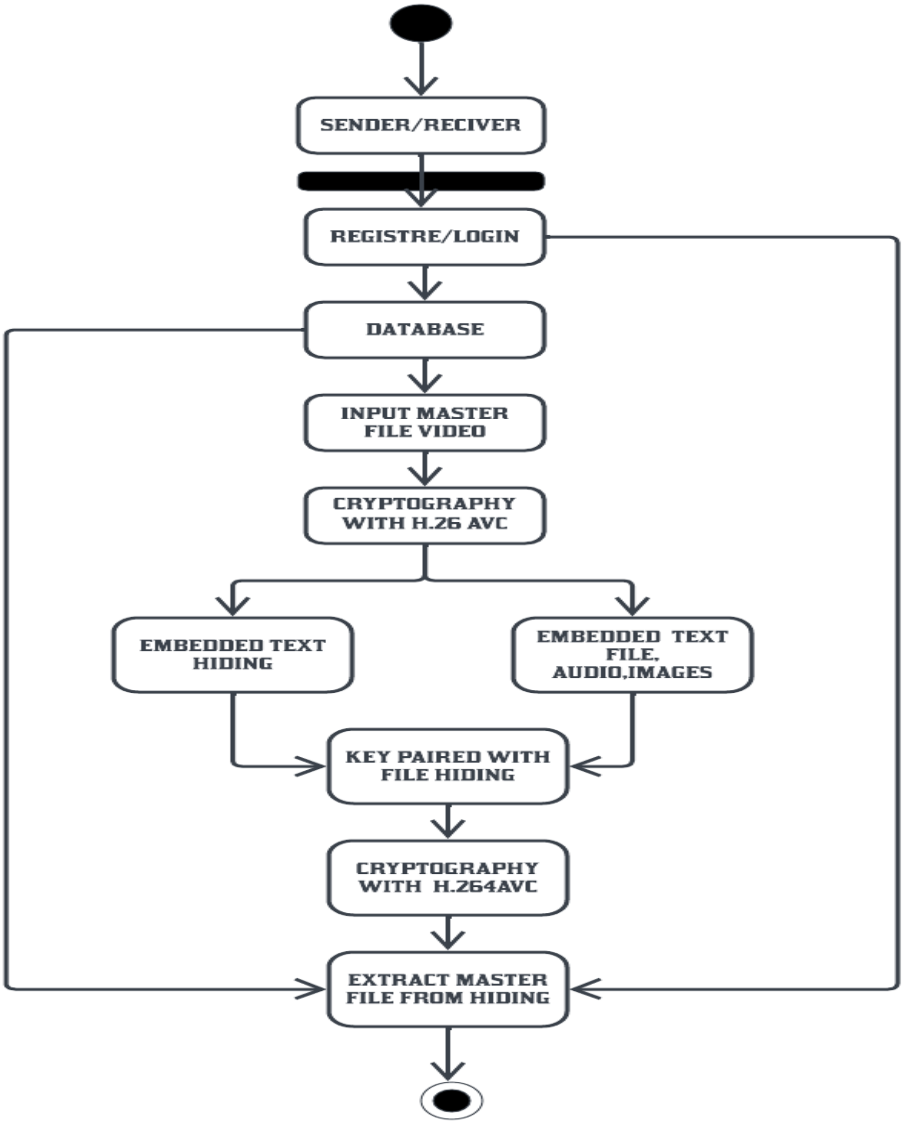

SENDER/RECIVER
REGISTRE/LOGIN
DATABASE
INPUT MASTER FILE VIDEO
CRYPTOGRAPHY WITH H.26 AVC
EMBEDDED TEXT HIDING
EMBEDDED TEXT FILE, AUDIO,IMAGES
KEY PAIRED WITH FILE HIDING
CRYPTOGRAPHY WITH H.264AVC
EXTRACT MASTER FILE FROM HIDING

Fig. 3.4.3 Activity Diagram

The activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities.The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs.

3.4.4 SEQUENCE DIAGRAM

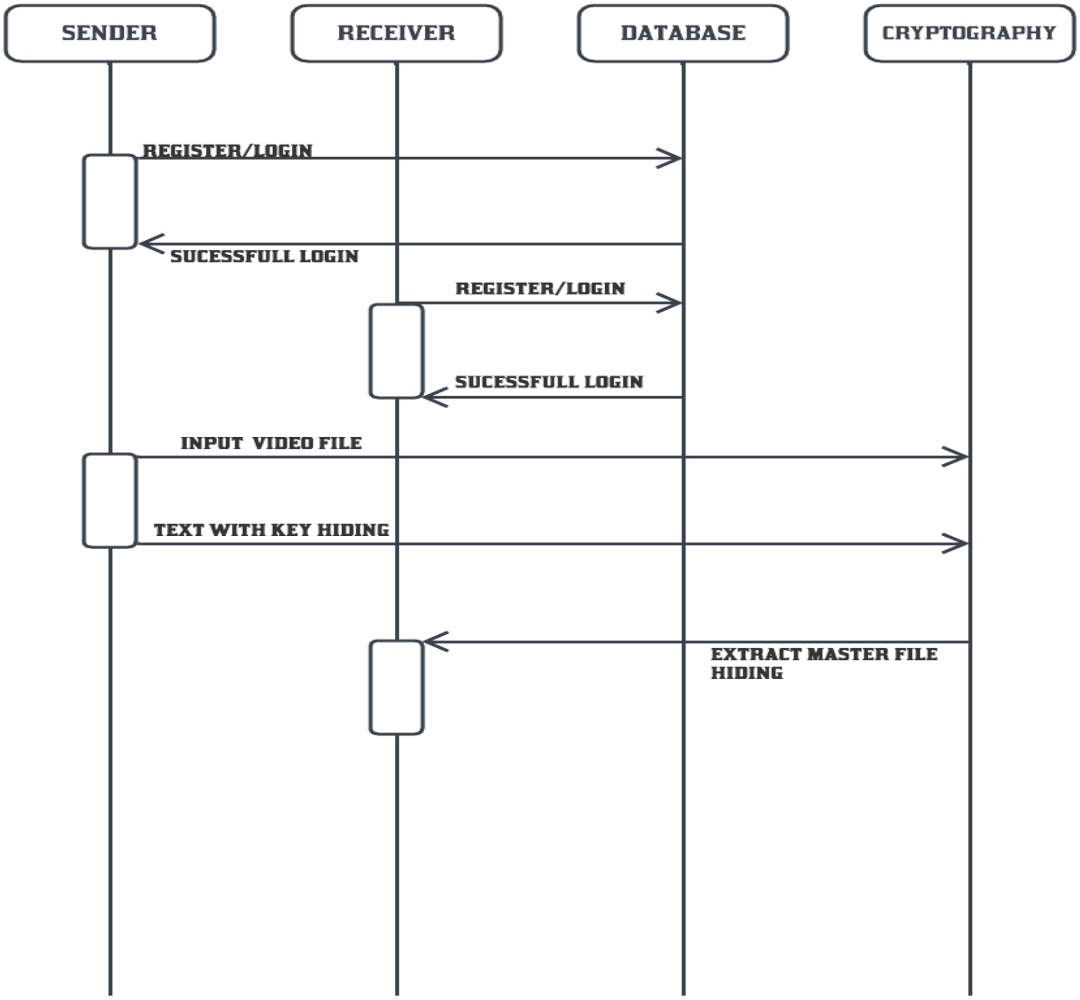


Fig. 3.4.4 Sequence Diagram

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

3.4.5 E-R DIAGRAM

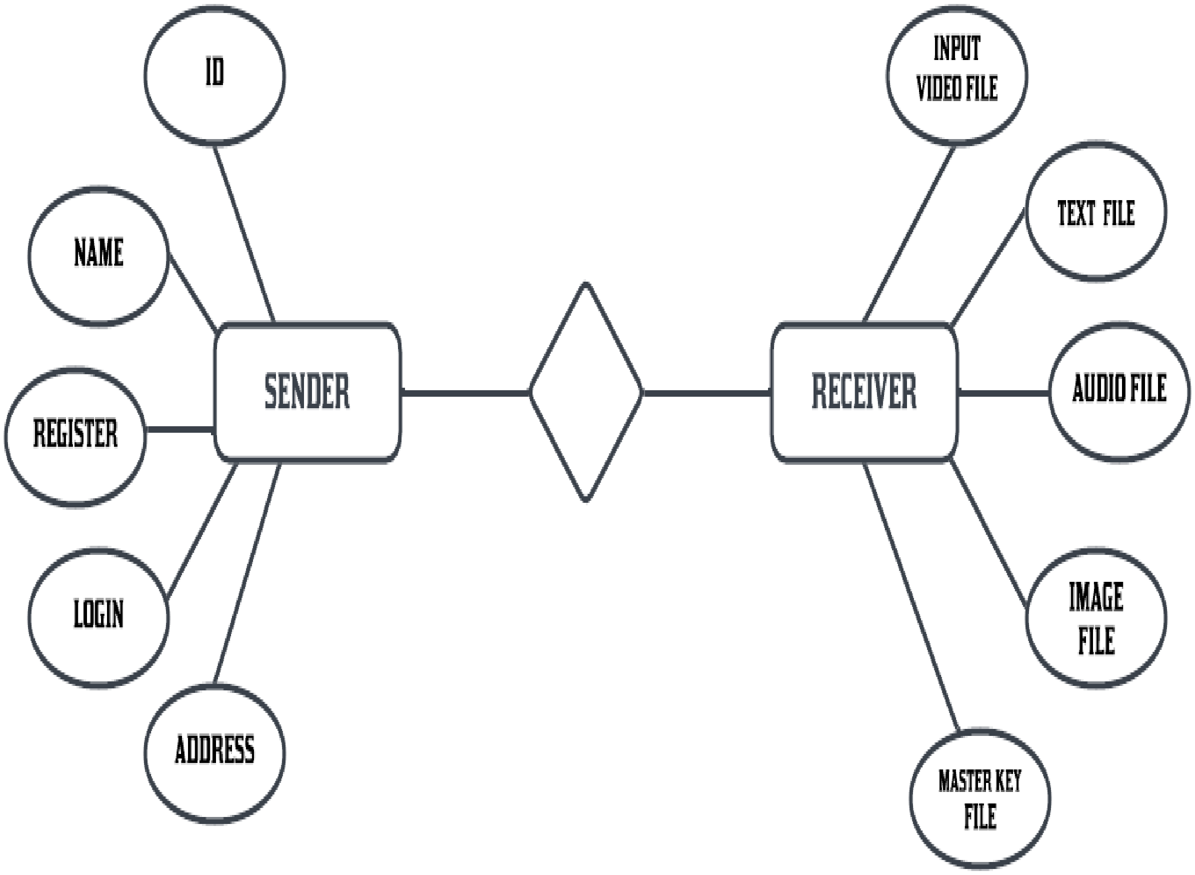


Fig. 3.4.5 E-R Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

3.4.6 DFD DIAGRAM

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one.

3.4.6.1 DFD LEVEL 0

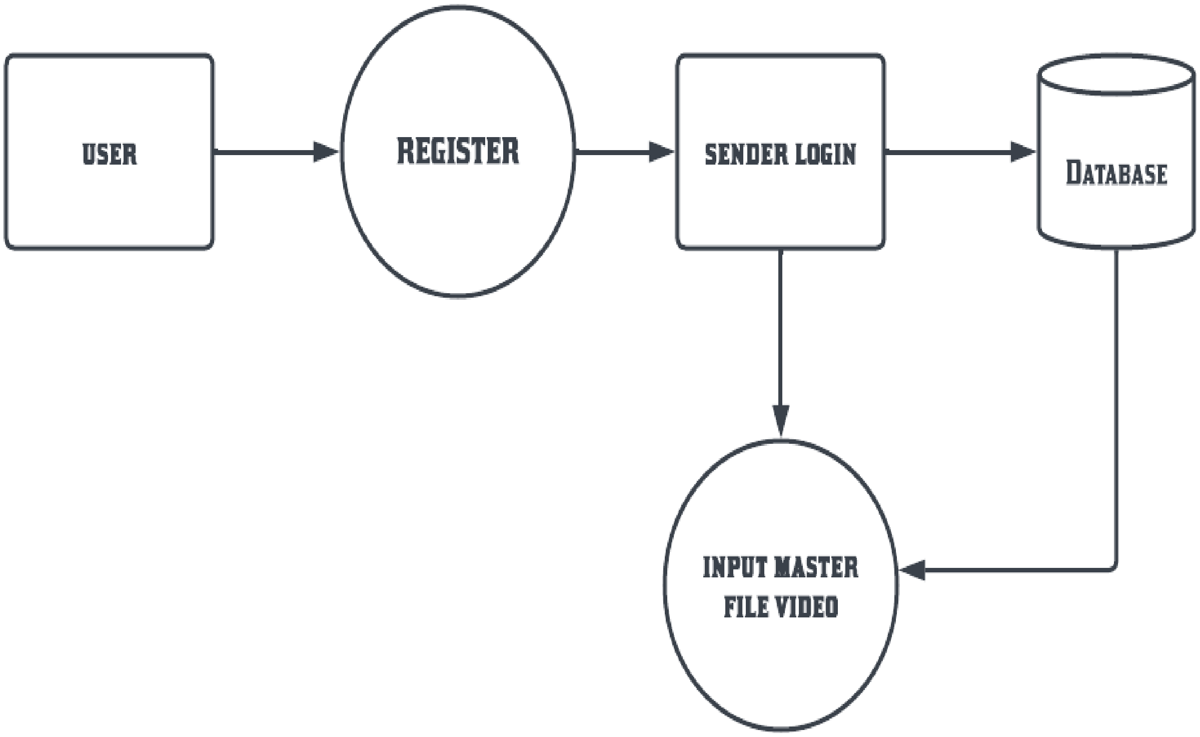


Fig. 3.4.6.1 DFD Level 0

3.4.6.2 DFD LEVEL 1

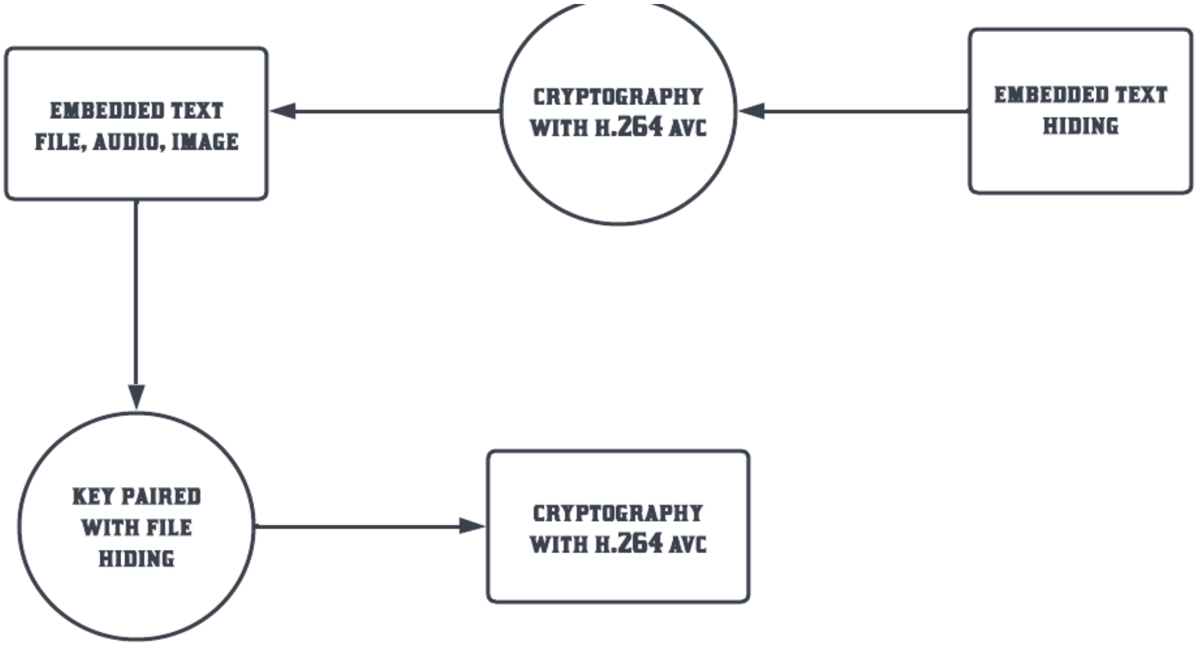

EMBEDDED TEXT FILE, AUDIO, IMAGE
CRYPTOGRAPHY WITH H.264 AVC
EMBEDDED TEXT HIDING
KEY PAIRED WITH FILE HIDING
CRYPTOGRAPHY WITH H.264 AVC

Fig. 3.4.6.2 DFD Level 1

3.4.6.3 DFD LEVEL 2

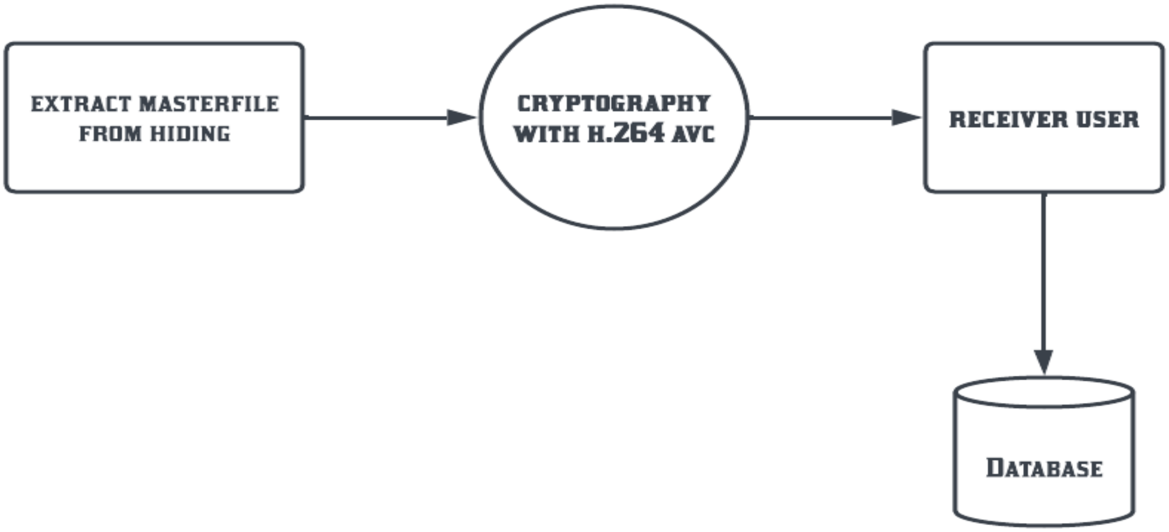
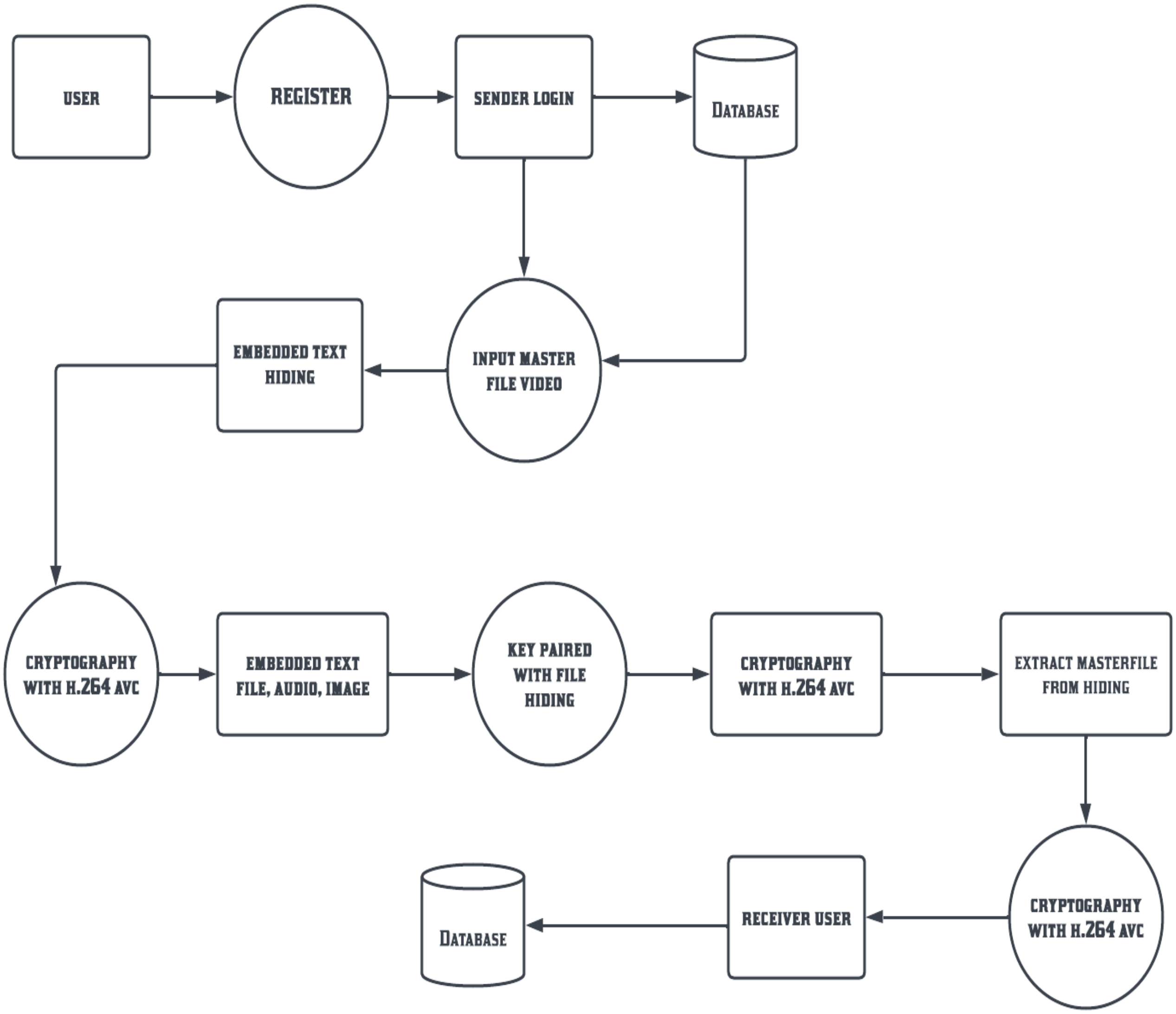


Fig. 3.4.6.3 DFD Level 2

3.4.6.4 OVERALL DFD


USER
REGISTER
SENDER LOGIN
DATABASE
INPUT MASTER FILE VIDEO
EMBEDDED TEXT HIDING
CRYPTOGRAPHY WITH H.264 AVC
EMBEDDED TEXT FILE, AUDIO, IMAGE
KEY PAIRED WITH FILE HIDING
CRYPTOGRAPHY WITH H.264 AVC
EXTRACT MASTERFILE FROM HIDING
CRYPTOGRAPHY WITH H.264 AVC
RECEIVER USER
DATABASE

Fig, 3.4.6.4 Overall DFD

3.5 MODULES

3.5.1 MODULE I

- AES Encryption
- ASCII to Binary Conversion

3.5.1.1 AES ENCRYPTION

In this module, describes Operation of AES (Advanced Encryption Standard). This AES uses 10 rounds for 128-bit keys, each of these rounds uses a different 128-bit round key, which is calculated from the original AES key. Each round comprise of four sub-processes. It is mainly used for encryption process.

3.5.1.2 ASCII TO BINARY CONVERSION

In this Module, ASCII (American Standard Code for Information Interchange) is the most common format for text file in computers and on the Internet. In an ASCII file, each alphabetic, numeric, or special character is represented with a 7-bit binary number (a string of seven 0s or 1s). 128 possible characters are defined.

3.5.2 MODULE II

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

3.5.2.1 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 slicesor “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”) for picture construction. 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 futureI or P slices (in playback order) may be used as reference pictures. For this to work, B slices must be decoded afterthe following I or P slice.

3.5.2.2 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.

3.5.2.3 MOTION VECTOR PREDECTION

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.

3.5.2.4 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 *B*0 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.

3.5.2.5 MACROBLOCK ORDERING

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

REQUIREMENT SPECIFICATION

4.1 HARDWARE REQUIREMENTS

- Processor - i3,i5,i7
- RAM - 4 GB
- Hard Disk - 500 GB

4.2 SOFTWARE REQUIREMENTS

- Operating System - 7/8/10
- Front End - Java
- Database - My sql
- Database Connectivity - JDBC

4.2.1 JAVA

Java is a programming language originally developed by James Gosling at Sun Microsystems now a subsidiary of Oracle Corporation, and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of their Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java and GNU Classpath.

4.2.1.1 JAVA PLATFORM

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any supported hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java bytecode, instead of directly to platform-specific machine code. Java bytecode instructions are analogous to machine code, but are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware. End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets.

Standardized libraries provide a generic way to access host-specific features such as graphics, threading and networking.

A major benefit of using bytecode is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executables would, and Java suffered a reputation for poor performance. This gap has been narrowed by a number of optimization techniques introduced in the more recent JVM implementations.

4.2.2 AUTOMATIC MEMORY MANAGEMENT

Java uses an automatic garbage collector to manage memory in the object lifecycle. The programmer determines when objects are created, and the Java runtime is responsible for recovering the memory once objects are no longer in use. Once no references to an object remain, the unreachable memory becomes eligible to be freed automatically by the garbage collector. Something similar to a memory leak may still occur if a programmer's code holds a reference to an object that is no longer needed, typically when objects that are no longer needed are stored in containers that are still in use. If methods for a nonexistent object are called, a "null pointer exception" is thrown.

One of the ideas behind Java's automatic memory management model is that programmers be spared the burden of having to perform manual memory management. In some languages memory for the creation of objects is implicitly allocated on the stack, or explicitly allocated and reallocated from the heap. Either way, the responsibility of managing memory resides with the programmer. If the program does not reallocate an object, a memory leak occurs. If the program attempts to access or reallocated memory that has already been reallocated, the result is undefined and difficult to predict, and the program is likely to become unstable and/or crash. This can be partially remedied by the use of smart pointers, but these add overhead and complexity. Note that garbage collection does not prevent 'logical' memory leaks, i.e. those where the memory is still referenced but never used.

Garbage collection may happen at any time. Ideally, it will occur when a program is idle. It is guaranteed to be triggered if there is insufficient free memory on the heap to allocate a new object; this can cause a program to stall momentarily. Explicit memory management is not possible in Java.

Java does not support C/C++ style pointer arithmetic, where object addresses and unsigned integers (usually long integers) can be used interchangeably. This allows the garbage collector to relocate referenced objects, and ensures type safety and security.

As in C++ and some other object-oriented languages, variables of Java's primitive data types are not objects. Values of primitive types are either stored directly in fields (for objects) or on the stack (for methods) rather than on the heap, as commonly true for objects (but see Escape analysis). This was a conscious decision by Java's designers for performance reasons. Because of this, Java was not considered to be a pure object-oriented programming language. However, as of Java 5.0, autoboxing enables programmers to proceed as if primitive types are instances of their wrapper classes.

4.2.3 MYSQL

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL is owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Sun Microsystems, a subsidiary of Oracle Corporation.

Members of the MySQL community have created several forks such as Drizzle and MariaDB. Both forks were in progress long before the Oracle acquisition (Drizzle was announced 8 months before the Sun acquisition).

Free-software projects that require a full-featured database management system often use MySQL. Such projects include (for example) WordPress, phpBB and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products including Wikipedia, Google, Drupal and Face book.

4.2.3.1 USES OF MYSQL

Many web applications use MySQL as the database component of a LAMP software stack. Its popularity for use with web applications is closely tied to the popularity of PHP, which is often combined with MySQL. Several high-traffic web sites (including Flickr, Facebook, Wikipedia, Google (though not for searches), Nokia and YouTube[11]) use MySQL for data storage and logging of user data.

4.2.3.2 PLATFORMS AND INTERFACES

MySQL code uses C and C++. The SQL parser uses yacc and a home-brewed lexer, sql_lex.cc.

MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, i5/OS, Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, eComStation, OS/2 Warp, QNX, IRIX, Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos, Tru64 and Microsoft Windows. A port of MySQL to OpenVMS also exists.

All major programming languages with language-specific APIs include Libraries for accessing MySQL databases. In addition, an ODBC interface called MyODBC allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. The MySQL server and official libraries are mostly implemented in ANSI C/ANSI C++.

4.2.3.3 MANAGEMENT AND GRAPHICAL FRONTENDS

MySQL Workbench in Windows, displaying the Home Screen which streamlines use of its full capabilities

MySQL is primarily an RDBMS and therefore ships with no GUI tools to administer MySQL databases or manage data contained within. Users may use the included command-line tools, or download MySQL Frontends from various parties that have developed desktop software and web applications to manage MySQL databases, build database structure, and work with data records.

4.2.3.4 OFFICIAL

The official MySQL Workbench is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structure. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage the following:

- Database design & modeling
- SQL development — replacing MySQL Query Browser
- Database administration — replacing MySQL Administrator

MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

Third party

Several other third-party proprietary and free graphical administration applications (or "Frontends") are available that integrate with MySQL and enable users to work with database structure and data visually. Some well-known frontends are:

- phpMyAdmin - a free Web-based frontend widely installed by Web hosts worldwide, since it is developed in PHP and only requires the LAMP stack to run.

- HeidiSQL - a full featured free frontend that runs on Windows, and can connect to local or remote MySQL servers to manage databases, tables, column structure, and individual data records. Also supports specialised GUI features for date/time fields and enumerated multiple-value fields.
- Navicat - a series of proprietary graphical database management applications, developed for Windows, Macintosh and Linux.
- Other available proprietary MySQL frontends include Adminer, Aqua Data Studio, dbForge Studio for MySQL, Epictetus, Oracle SQL Developer, SchemaBank, SQLyog, SQLPro SQL Client, Toad and Toad Data Modeler.

4.2.3.5 DEPLOYMENT

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well.

It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin.

In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

There are however limits to how far performance can scale on a single server, so on larger scales, multi-server MySQL deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations. The master server synchronizes continually with its slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using memcached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

4.2.3.6 FEATURES

As of April 2009, MySQL offers MySQL 5.1 in two different variants: the MySQL Community Server and Enterprise Server. They have a common code base and include the following features:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures
- Triggers
- Cursors
- Updatable Views
- True Varchar support
- INFORMATION_SCHEMA

- Strict mode
- X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using Oracle's InnoDB engine
- Independent storage engines (MyISAM for read speed, InnoDB for transactions and referential integrity, MySQL Archive for storing historical data in little space)
- Transactions with the InnoDB, BDB and Cluster storage engines; savepoints with InnoDB
- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Replication support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master, no automatic support for multiple masters per slave.
- Full-text indexing (Index_(database)) and searching using MyISAM engine
- Embedded database library
- Partial Unicode support (UTF-8 and UCS-2 encoded strings are limited to the BMP)
- Partial ACID compliance (full compliance only when using the non-default storage engines InnoDB, BDB and Cluster)
- Shared-nothing clustering through MySQL Cluster
- Hot backup (via mysqlhotcopy) under certain conditions

The developers release monthly versions of the MySQL Enterprise Server. The sources can be obtained either from MySQL's customer-only Enterprise site or from MySQL's Bazaar repository, both under the GPL license. The MySQL Community Server is published on an unspecified schedule under the GPL and contains all bug fixes that were shipped with the last MySQL Enterprise Server release. Binaries are no longer provided by MySQL for every release of the Community Server.

Distinguishing Features

MySQL implements the following features, which some other RDBMS systems may not:

- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at run time):
    - Native storage engines (MyISAM, Falcon, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, Cluster, Berkeley DB, EXAMPLE, and Maria)
    - Partner-developed storage engines (InnoDB, solidDB, NitroEDB, Infobright (formerly Brighthouse), Kickfire, XtraDB, IBM DB2[22])
    - Community-developed storage engines (memcache_engine, httpd, PBXT, Revision Engine)
    - Custom storage engines
- Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

Product History

Milestones in MySQL development include:

- Original development of MySQL by Michael Widenius and David Axmark beginning in 1994

- First internal release on 23 May 1995
- Windows version was released on 8 January 1998 for Windows 95 and NT
- Version 3.23: beta from June 2000, production release January 2001
- Version 4.0: beta from August 2002, production release March 2003 (unions)
- Version 4.01: beta from August 2003, Jyoti adopts MySQL for database tracking
- Version 4.1: beta from June 2004, production release October 2004 (R-trees and B-trees, subqueries, prepared statements)
- Version 5.0: beta from March 2005, production release October 2005 (cursors, stored procedures, triggers, views, XA transactions)

The developer of the Federated Storage Engine states that "The Federated Storage Engine is a proof-of-concept storage engine", but the main distributions of MySQL version 5.0 included it and turned it on by default. Documentation of some of the short-comings appears in "MySQL Federated Tables: The Missing Manual".

- Sun Microsystems acquired MySQL AB on 26 February 2008.
- Version 5.1: production release 27 November 2008 (event scheduler, partitioning, plugin API, row-based replication, server log tables)

Version 5.1 contained 20 known crashing and wrong result bugs in addition to the 35 present in version 5.0.

MySQL 5.1 and 6.0 showed poor performance when used for data warehousing partly due to its inability to utilize multiple CPU cores for processing a single query.

- Oracle acquired Sun Microsystems on January 27, 2010.Oracle and Sun

Future releases

The MySQL 6 roadmap outlines support for:

- Referential integrity and Foreign key support for all storage engines is targeted for release in MySQL 6.1 (although it has been present since version 3.23.44 for InnoDB).
- Support for supplementary Unicode characters, beyond the 65,536 characters of the Basic Multilingual Plane (BMP); announced for MySQL 6.0.
- A new storage engine called Falcon. A preview of Falcon is available on MySQL's website.

The 2006 roadmap for future versions plans support for parallelization.

4.2.3.7 SUPPORT AND LICENSING

Via MySQL Enterprise MySQL AB offers support itself, including a 24/7 service with 30-minute response time. The support team has direct access to the developers as necessary to handle problems. In addition, it hosts forums and mailing lists, employees and other users are often available in several IRC channels providing assistance.

In addition to official product support from Sun, other companies offer support and services related to usage of MySQL. For example, Pythian offers full database administration, architecture, optimization and training services. Percona and 42sql offer services related to optimization and Monty Program Ab offers non-recurring engineering such as patches to MySQL. OpenQuery provides MySQL training.

Buyers of MySQL Enterprise have access to binaries and software certified for their particular operating system, and access to monthly binary updates with the latest bug-fixes. Several levels of Enterprise membership are available, with varying response times and features ranging from how to and emergency support through server performance tuning and system architecture advice. The MySQL Network Monitoring and Advisory Service monitoring tool for database servers is available only to MySQL Enterprise customers.

Potential users can install MySQL Server as free software under the GNU General Public License (GPL), and the MySQL Enterprise subscriptions include a GPL version of the server, with a traditional proprietary version available on request at no additional cost for cases where the intended use is incompatible with the GPL.

Both the MySQL server software itself and the client libraries use dual-licensing distribution. Users may choose the GPL,[29] which MySQL has extended with a FLOSS License Exception. It allows Software licensed under other OSI-compliant open source licenses, which are not compatible to the GPL, to link against the MySQL client libraries.

Customers that do not wish to follow the terms of the GPL may purchase a proprietary license.Like many open-source programs, MySQL has trademarked its name, which others may use only with the trademark holder's permission.

IMPLEMENTATION

Sun Microsystems officially licenses the Java Standard Edition platform for Linux, Mac OS X and Solaris. Although in the past Sun has licensed Java to Microsoft, the license has expired and has not been renewed. Through a network of third-party vendors and licensees, alternative Java environments are available for these and other platforms.

Sun's trademark license for usage of the Java brand insists that all implementations be "compatible". This resulted in a legal dispute with Microsoft after Sun claimed that the Microsoft implementation did not support RMI or JNI and had added platform-specific features of their own. Sun sued in 1997, and in 2001 won a settlement of $20 million as well as a court order enforcing the terms of the license from Sun. As a result, Microsoft no longer ships Java with Windows, and in recent versions of Windows, Internet Explorer cannot support Java applets without a third-party plugin. Sun, and others, has made available free Java run-time systems for those and other versions of Windows.

Platform-independent Java is essential to the Java EE strategy, and an even more rigorous validation is required to certify an implementation. This environment enables portable server-side applications, such as Web services, Java Servlets, and Enterprise JavaBeans, as well as with embedded systems based on OSGi, using Embedded Java environments. Through the new GlassFish project, Sun is working to create a fully functional, unified open source implementation of the Java EE technologies.

Sun also distributes a superset of the JRE called the Java Development Kit (commonly known as the JDK), which includes development tools such as the Java compiler, Javadoc, Jar and debugger.

5.1 PERFORMANCE

Programs written in Java have a reputation for being slower and requiring more memory than those written in some other languages. However, Java programs' execution speed improved significantly with the introduction of Just-in-time compilation in 1997/1998 for Java 1.1, the addition of language features supporting better code analysis (such as inner classes, StringBuffer class, optional assertions, ect.), and optimizations in the Java Virtual Machine itself, such as HotSpot becoming the default for Sun's JVM in 2000.

To boost even further the speed performances that can be achieved using the Java language Systronix made JStik, a microcontroller based on the aJile Systems line of embedded Java processors.

5.2 EMBEDDING ALGORITHM

```
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
 {
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
 } } }
```

5.3 AES ALGORITHM:

AES (acronym of Advanced Encryption Standard) is a symmetric encryption algorithm. The algorithm was developed by two Belgian cryptographer Joan Daemen and Vincent Rijmen. AES was designed to be efficient in both hardware and software, and supports a block length of 128 bits and key lengths of 128, 192, and 256 bits.

STEPS IN ADVANCED ENCRYPTION STANDARD:

STEP 1

Derive the set of round keys from the cipher key

STEP 2

Initialize the state array with the block data

STEP 3

Add the initial round key to the starting state array

STEP 4

Perform nine rounds of state manipulation

STEP 5

Perform the tenth and final round of state manipulation

STEP 6

Copy the final state array out as the encrypted data

5.4SAMPLE CODE

```
package Hiding;
importjava.awt.FileDialog;
importjava.io.BufferedInputStream;
importjava.io.BufferedOutputStream;
importjava.io.BufferedReader;
importjava.io.BufferedWriter;
importjava.io.File;
importjava.io.FileInputStream;
importjava.io.InputStream;
importjava.io.InputStreamReader;
importjava.io.OutputStream;
importjava.io.OutputStreamWriter;
importjava.net.InetAddress;
importjava.net.Socket;
importjava.security.Key;
importjava.sql.*;
importjava.text.DateFormat;
importjava.text.SimpleDateFormat;
importjava.util.Calendar;
importjava.util.regex.Matcher;
importjava.util.regex.Pattern;
importjavax.crypto.Cipher;
importjavax.crypto.SecretKeyFactory;
importjavax.swing.JFileChooser;
importjavax.swing.JFrame;
importjavax.swing.JOptionPane;
importjavax.swing.ProgressMonitor;
importjavax.swing.table.DefaultTableModel;
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author
 */
public class LOGINN extends javax.swing.JFrame {
    /**
     * Creates new form CLIENT
     */
public LOGINN() {
```

```
initComponents();
    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
@SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jButton6 = new javax.swing.JButton();
        jButton7 = new javax.swing.JButton();
        jPanel3 = new javax.swing.JPanel();
        jLabel4 = new javax.swing.JLabel();
        jTextField3 = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jPasswordField1 = new javax.swing.JPasswordField();
        jButton3 = new javax.swing.JButton();
        jLabel5 = new javax.swing.JLabel();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jPanel1.setLayout(null);

jLabel1.setBackground(new java.awt.Color(204, 204, 204));

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel1.setForeground(new java.awt.Color(255, 255, 255));

jLabel1.setText("PROTECTING THE SECRECY OF VIDEO USING ADVANCE DATA HIDINGTECHNIQUES");

jPanel1.add(jLabel1);

jLabel1.setBounds(0, 0, 1150, 60);

jPanel2.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));


javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGap(0, 1156, Short.MAX_VALUE)

        );

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGap(0, 6, Short.MAX_VALUE)

        );

jPanel1.add(jPanel2);
```

```
jPanel2.setBounds(0, 60, 1160, 10);

jLabel2.setFont(new java.awt.Font("Tahoma", 3, 14)); // NOI18N
jLabel2.setForeground(new java.awt.Color(204, 204, 255));
jPanel1.add(jLabel2);
jLabel2.setBounds(510, 60, 180, 30);

jButton6.setFont(new java.awt.Font("Lucida Calligraphy", 3, 14)); // NOI18N
jButton6.setForeground(new java.awt.Color(51, 0, 0));
jButton6.setText("HOME");
jButton6.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEventevt) {
jButton6ActionPerformed(evt);
            }
        });
jPanel1.add(jButton6);
jButton6.setBounds(10, 200, 170, 33);

jButton7.setFont(new java.awt.Font("Lucida Calligraphy", 3, 14)); // NOI18N
jButton7.setForeground(new java.awt.Color(51, 0, 0));
jButton7.setText("REGISTER");
jButton7.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEventevt) {
jButton7ActionPerformed(evt);
            }
        });
jPanel1.add(jButton7);
jButton7.setBounds(10, 260, 170, 33);

jPanel3.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.bord
er.BevelBorder.RAISED));

javax.swing.GroupLayout          jPanel3Layout          =          new
javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
            .addGap(0, 6, Short.MAX_VALUE)
        );
jPanel3Layout.setVerticalGroup(
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
            .addGap(0, 526, Short.MAX_VALUE)
        );

jPanel1.add(jPanel3);
jPanel3.setBounds(200, 60, 10, 530);

jLabel4.setFont(new java.awt.Font("Times New Roman", 3, 14)); // NOI18N
jLabel4.setForeground(new java.awt.Color(255, 255, 255));
```

```
jLabel4.setText("USER LOGIN");
jPanel1.add(jLabel4);
jLabel4.setBounds(460, 90, 180, 20);
jPanel1.add(jTextField3);
jTextField3.setBounds(560, 230, 220, 30);

jLabel3.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel3.setForeground(new java.awt.Color(255, 255, 255));
jLabel3.setText("USER NAME          :");
jPanel1.add(jLabel3);
jLabel3.setBounds(410, 230, 130, 30);

jLabel6.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setText("PASSWORD  :");
jPanel1.add(jLabel6);
jLabel6.setBounds(410, 290, 120, 30);
jPanel1.add(jPasswordField1);
jPasswordField1.setBounds(560, 290, 220, 30);

jButton3.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jButton3.setText("Login");
jButton3.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEventevt) {
jButton3ActionPerformed(evt);
```

```
            }
        });
jPanel1.add(jButton3);
jButton3.setBounds(530, 390, 130, 30);

jLabel5.setIcon(newjavax.swing.ImageIcon(getClass().getResource("/Hiding/url.jpg")));  NOI18N
jLabel5.setText("jLabel5");
jPanel1.add(jLabel5);
jLabel5.setBounds(0, 0, 1150, 590);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1,javax.swing.GroupLayout.DEFAULT_SIZE,1152, Short.MAX_VALUE)
        );
layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1,javax.swing.GroupLayout.DEFAULT_SIZE,586, Short.MAX_VALUE)
        );
```

```
pack();
    }// </editor-fold>

private void jButton7ActionPerformed(java.awt.event.ActionEventevt) {
        // TODO add your handling code here:

        REGISTERR ll = new REGISTERR();
ll.setVisible(true);
ll.MAC();
dispose();
    }

private void jButton3ActionPerformed(java.awt.event.ActionEventevt) {
        // TODO add your handling code here:

if (jTextField3.getText().equals("")) {
JOptionPane.showMessageDialog(this, "UserName Should not empty ");
        }

if (jPasswordField1.getText().equals("")) {
JOptionPane.showMessageDialog(this, "Password Should not empty ");
        } else {
            String Name = jTextField3.getText().trim();
            String Pass = jPasswordField1.getText().trim();
```

```
boolean x = false;
        Statement s1 = null;
        Connection con = null;
PreparedStatementst = null;
try {
Class.forName("com.mysql.jdbc.Driver");
con = DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "root",
"admin");


           String qry1 = "Select * from reg where uname ='" + Name + "' and
pass='" + Pass + "' ";


           s1 = con.createStatement();
ResultSetrstt = s1.executeQuery(qry1);
if (rstt.next()) {


JOptionPane.showMessageDialog(this, "Verified");
              HOME ms = new HOME();
ms.setVisible(true);


dispose();
s1.close();
rstt.close();
           } else {
JOptionPane.showMessageDialog(this, "User Name or Password Wrong");
```

```
        }} catch (Exception e) {
System.out.println(e);
      }}
}

private void jButton6ActionPerformed(java.awt.event.ActionEventevt) {
    // TODO add your handling code here:
    LOGINN ll = new LOGINN();
ll.setVisible(true);
dispose();
  }
  /**
   * @paramargs the command line arguments
   */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
     * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
try {
for(javax.swing.UIManager.LookAndFeelInfoinfo :
javax.swing.UIManager.getInstalledLookAndFeels()) {
```

```
if ("Nimbus".equals(info.getName())) {
javax.swing.UIManager.setLookAndFeel(info.getClassName());
break;
                }
            }
        } catch (Exception ex) {
java.util.logging.Logger.getLogger(HOME.class.getName()).log(java.util.logging.
Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new LOGINN().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
privatejavax.swing.JButton jButton3;
privatejavax.swing.JButton jButton6;
privatejavax.swing.JButton jButton7;
privatejavax.swing.JLabel jLabel1;
privatejavax.swing.JLabel jLabel2;
privatejavax.swing.JLabel jLabel3;
privatejavax.swing.JLabel jLabel4;
privatejavax.swing.JLabel jLabel5;
privatejavax.swing.JLabel jLabel6;
privatejavax.swing.JPanel jPanel1;
privatejavax.swing.JPanel jPanel2;
privatejavax.swing.JPanel jPanel3;
privatejavax.swing.JPasswordField jPasswordField1;
privatejavax.swing.JTextField jTextField3;
    // End of variables declaration
}
```

MainFrame.java:

```
package Hiding;

importjava.awt.BorderLayout;
importjava.awt.Color;
importjava.awt.Dimension;
importjava.awt.FileDialog;
importjava.awt.Font;
importjava.awt.GridBagConstraints;
importjava.awt.GridBagLayout;
importjava.awt.Toolkit;
importjava.awt.event.ActionEvent;
importjava.awt.event.ActionListener;
importjava.awt.event.ItemEvent;
importjava.awt.event.ItemListener;

importjava.awt.event.WindowAdapter;

importjava.awt.event.WindowEvent;

importjavax.swing.ButtonGroup;

importjavax.swing.ImageIcon;

importjavax.swing.JFrame;

importjavax.swing.JLabel;

importjavax.swing.JMenu;

importjavax.swing.JMenuBar;

importjavax.swing.JMenuItem;

importjavax.swing.JOptionPane;

importjavax.swing.JPanel;

importjavax.swing.JRadioButtonMenuItem;

importjavax.swing.SwingUtilities;

importjavax.swing.UIManager;

importjavax.swing.border.TitledBorder;


public class MainClient extends WindowAdapter implements ActionListener

{privateFileDialogfd;

private static JFramedummyFrame = new JFrame();

        privateJFrame              mainFrame;

        privateJMenuBar   menuBar;

        privateJMenu               menuFile,    menuEdit,    menuView,    menuHelp,
menuLookAndFeel;
```

```
    privateJMenuItem mnuExit,    mnuEmbedMessage,    mnuEmbedFile,
mnuHelp, mnuAbout;
    privateJMenuItem mnuRetrieveMessage,    mnuRetrieveFile,
mnuModifyMaster;
    privateJRadioButtonMenuItem mnuTonicFeel,    mnuMetalFeel,
mnuMotifFeel, mnuWindowsFeel;
    privateButtonGrouplookAndFeelButtonGroup;

    privateJPanelmainPanel, panelAbout, panelButtons;
    privateJLabellblLogo;
    privateJLabellblFiller[], lblName, lblEmail, lblPhone;
    privateGridBagLayoutgbl;
    privateGridBagConstraintsgbc;
    privateMyJButtonbtnEmbedFile,btnRetrieveFile,btnEmbedMessage,
btnRetrieveMessage;
    privateMyJButtonbtnHelp, btnAbout;
    privateBackEndHandler back;

    publicMainClient()
    {
        mainFrame= new JFrame("CPT ALGORITHM"+ "  ");
        mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mainFrame.addWindowListener(this);

        // Setup the menu bar
        mnuExit= new MyJMenuItem("Exit", 1, 'x');
        mnuEmbedMessage= new MyJMenuItem("Embed Message", 6, 'm');
        mnuEmbedFile= new MyJMenuItem("Embed File", 7, 'i');
        mnuRetrieveMessage= new MyJMenuItem("Retrieve Message", 0, 'r');
        mnuRetrieveFile= new MyJMenuItem("Retrieve File", 2, 't');
        mnuModifyMaster= new MyJMenuItem("Modify Master file settings", 2, 'd');
        mnuHelp= new MyJMenuItem("Help", 0, 'h');
        mnuAbout= new MyJMenuItem("Encoding", 0, 'a');
        mnuTonicFeel= new MyJRadioButtonMenuItem("Plastic XP", 8, 'x');
        mnuMetalFeel= new MyJRadioButtonMenuItem("Metal", 0, 'm');
        mnuMotifFeel= new MyJRadioButtonMenuItem("Motif", 2, 't');
        mnuWindowsFeel= new MyJRadioButtonMenuItem("Windows", 0, 'w');

        // Add item listener for Look and feel menu items
        RadioListenerradioListener= new RadioListener();
        mnuTonicFeel.addItemListener((ItemListener) radioListener);
        mnuMetalFeel.addItemListener(radioListener);
        mnuMotifFeel.addItemListener(radioListener);
        mnuWindowsFeel.addItemListener(radioListener);
        mnuTonicFeel.setSelected(true);

        lookAndFeelButtonGroup= new ButtonGroup();
        lookAndFeelButtonGroup.add(mnuTonicFeel);
        lookAndFeelButtonGroup.add(mnuMetalFeel);
```

```
lookAndFeelButtonGroup.add(mnuMotifFeel);
lookAndFeelButtonGroup.add(mnuWindowsFeel);

// Add action listeners for other menu items
mnuEmbedMessage.addActionListener(this);
mnuEmbedFile.addActionListener(this);
mnuRetrieveMessage.addActionListener(this);
mnuRetrieveFile.addActionListener(this);
mnuModifyMaster.addActionListener(this);
mnuExit.addActionListener(this);
mnuHelp.addActionListener(this);
mnuAbout.addActionListener(this);

menuFile= new MyJMenu("File", 0, 'f');
menuFile.add(mnuEmbedMessage);
menuFile.add(mnuEmbedFile);
menuFile.add(mnuRetrieveMessage);
menuFile.add(mnuRetrieveFile);
menuFile.add(mnuExit);

menuEdit= new JMenu("Edit");
menuEdit.add(mnuModifyMaster);

menuLookAndFeel= new MyJMenu("Look and Feel...", 0, 'l');
menuLookAndFeel.add(mnuTonicFeel);
menuLookAndFeel.add(mnuMetalFeel);
menuLookAndFeel.add(mnuMotifFeel);
menuLookAndFeel.add(mnuWindowsFeel);
menuView= new MyJMenu("View", 0, 'v');
menuView.add(menuLookAndFeel);

menuHelp= new MyJMenu("Help", 0, 'h');
menuHelp.add(mnuHelp);
menuHelp.add(mnuAbout);

menuBar= new JMenuBar();
menuBar.add(menuFile);
//menuBar.add(menuEdit);
menuBar.add(menuView);
menuBar.add(menuHelp);
mainFrame.setJMenuBar(menuBar);

mainPanel= new JPanel();
panelAbout= new JPanel();
panelButtons= new JPanel();



// Create filler labels
lblFiller= new JLabel[4];
for(int i=0; i<4; i++)
```

```
            lblFiller[i]= new JLabel(" ");
        // Prepare About panel
        gbl= new GridBagLayout();
        gbc= new GridBagConstraints();
        panelAbout.setLayout(gbl);
        panelAbout.setBackground(Color.white);
        Color myColor= new Color(50, 153, 237);
        Font arialFont= new Font("Arial", Font.PLAIN, 14);
        Font myFont= new Font("", Font.PLAIN, 18);
        lblName= new MyJLabel("", myFont, Color.blue, Color.white);
        lblEmail= new MyJLabel("", arialFont, myColor, Color.white);
        lblPhone= new JLabel(" ");

        gbc.gridx= 1;      gbc.gridy= 1;      gbl.setConstraints(lblName, gbc);
        panelAbout.add(lblName);
        gbc.gridx= 2;      gbc.gridy= 2;      gbl.setConstraints(lblEmail, gbc);
        panelAbout.add(lblEmail);
        gbc.gridx= 3;      gbc.gridy= 3;      gbl.setConstraints(lblPhone, gbc);
        panelAbout.add(lblPhone);

        // Prepare the Buttons panel
        panelButtons.setBackground(Color.white);
        gbl= new GridBagLayout();
```

```
            panelButtons.setLayout(gbl);

            panelButtons.setBorder(new TitledBorder("Supported operations"));


            lblLogo= new JLabel(new
ImageIcon(this.getClass().getResource("")));

            btnEmbedMessage=newMyJButton("D:\\Vinoth\\Netbeans
Project\\Convert_Gray_Image\\src\\Images\\EmbedMessage","D:\\Admin\\Netbea
ns Project\\Convert_Gray_Image\\src\\Images\\EmbedMessageHover");

            btnEmbedFile=newMyJButton("D:\\Admin\\Netbeans
Project\\Convert_Gray_Image\\src\\Images\\EmbedFile","D:\\Admin\\Netbeans
Project\\Convert_Gray_Image\\src\\Images\\EmbedFileHover");

            btnRetrieveMessage=newMyJButton("D:\\Admin\\Netbeans
Project\\Convert_Gray_Image\\src\\Images\\RetrieveMessage","D:\\Admin\\Netbe
ans Project\\Convert_Gray_Image\\src\\Images\\RetrieveMessageHover");

            btnRetrieveFile=newMyJButton("D:\\Admin\\Netbeans
Project\\Convert_Gray_Image\\src\\Images\\RetrieveFile","D:\\Admin\\Netbeans
Project\\Convert_Gray_Image\\src\\Images\\RetrieveFileHover");

            btnHelp=newMyJButton("Images/Help","D:/visualcrypto/visual
crypto/src/Images/HelpHover");

            btnAbout=newMyJButton("Images/About","D:/visualcrypto/visual
crypto/src/Images/AboutHover");


            // Add action listeners for the buttons

            btnEmbedMessage.addActionListener(this);

            btnEmbedFile.addActionListener(this);

            btnRetrieveMessage.addActionListener(this);

            btnRetrieveFile.addActionListener(this);

            btnHelp.addActionListener(this);

            btnAbout.addActionListener(this);
```

```
        // Add filler for rows 1 and 2
        gbc.weightx= 4;     gbc.weighty= 2;     gbc.fill= gbc.BOTH;
        gbc.gridx= 6;       gbc.gridy= 1;       gbl.setConstraints(lblFiller[0],
gbc);
        panelButtons.add(lblFiller[0]);

        gbc.weightx= 1;     gbc.weighty= 1;     gbc.fill= gbc.NONE;
        gbc.gridx= 3;       gbc.gridy= 3;       gbl.setConstraints(btnHelp,
gbc);
        panelButtons.add(btnHelp);

        gbc.gridx= 5;       gbl.setConstraints(btnAbout, gbc);
        panelButtons.add(btnAbout);

        // Add filler for rows 4 and 5
        gbc.fill = gbc.BOTH;
        gbc.gridx= 1;       gbc.weighty=                          2;gbc.gridy=4;
    gbl.setConstraints(lblFiller[1], gbc);
        panelButtons.add(lblFiller[1]);

        gbc.fill= gbc.NONE;
        gbc.gridx= 2;       gbc.weighty=1;gbc.gridy=6;
    gbl.setConstraints(btnEmbedMessage, gbc);
        panelButtons.add(btnEmbedMessage);
```

```
            gbc.gridx= 4;         gbl.setConstraints(btnRetrieveMessage, gbc);
            panelButtons.add(btnRetrieveMessage);


            // Add filler for row 7 and 8
            gbc.fill = gbc.BOTH;
            gbc.gridx= 6;         gbc.weighty= 2;     gbc.gridy=7;
        gbl.setConstraints(lblFiller[2], gbc);
            panelButtons.add(lblFiller[2]);


            gbc.fill= gbc.NONE;
            gbc.gridx= 3;         gbc.weighty= 1;     gbc.gridy=9;
        gbl.setConstraints(btnEmbedFile, gbc);
            panelButtons.add(btnEmbedFile);


            gbc.gridx= 5;         gbl.setConstraints(btnRetrieveFile, gbc);
            panelButtons.add(btnRetrieveFile);


            // Add the lblLogo and two Panels to the mainPanel
            gbl= new GridBagLayout();
            mainPanel.setLayout(gbl);
            mainPanel.setBackground(Color.white);


            gbc.anchor= gbc.CENTER;
            gbc.gridx= 1;         gbc.gridy= 1;        gbc.weighty=            2;gbc.fill=
    gbc.VERTICAL;
            gbl.setConstraints(lblLogo, gbc);
```

```
                mainPanel.add(lblLogo);


                gbc.gridy= 3;          gbc.weighty= 2;
                gbl.setConstraints(panelAbout, gbc);
                mainPanel.add(panelAbout);


                gbc.gridy= 5;          gbc.weighty= 1;
                gbl.setConstraints(panelButtons, gbc);
                mainPanel.add(panelButtons);


                gbc.gridy= 6;          gbc.weighty= 2;
                gbl.setConstraints(lblFiller[3], gbc);
                mainPanel.add(lblFiller[3]);


                JPaneltempPanel= (JPanel) mainFrame.getContentPane();
                tempPanel.add(mainPanel, BorderLayout.CENTER);
                tempPanel.add(newMyJLabel("",Color.black,Color.darkGray),
BorderLayout.SOUTH);


                Dimension d= Toolkit.getDefaultToolkit().getScreenSize();
                mainFrame.setSize(d.width, (int) (d.height-(d.height*.03)));
                mainFrame.setResizable(false);
                mainFrame.setVisible(true);
        }
        // Listener methods
        public void actionPerformed(ActionEvent e)
        {
                Object source= e.getSource();

                // Embed message operation
                if(source== mnuEmbedMessage || source== btnEmbedMessage)
                {
                        back=new                    BackEndHandler(this,
    BackEndHandler.EMBED_MESSAGE);
                        back.start();
                }

                // Retrieve message operation
                if(source== mnuRetrieveMessage || source== btnRetrieveMessage)
                {

        back=newBackEndHandler(this,BackEndHandler.RETRIEVE_MESSAGE);
                        back.start();
                }
                // Embed file operation
                if(source== mnuEmbedFile || source== btnEmbedFile )
                {
                        back=          new          BackEndHandler(this,
    BackEndHandler.EMBED_FILE);
                        back.start();
                }
                // Retrieve file operation
                if(source== mnuRetrieveFile || source== btnRetrieveFile )
                {

                        back= new BackEndHandler(this, BackEndHandler.RETRIEVE_FILE);
                        back.start();
                }

                // Modify Master file operation
                if(source== mnuModifyMaster)
                {
                        back= new BackEndHandler(this, BackEndHandler.EDIT_MASTER);
                        back.start();
                }
                if(source== mnuHelp || source==btnHelp)
                if(source== mnuAbout || source== btnAbout)
                if(source== mnuExit)
                {
                        int result= JOptionPane.showConfirmDialog(mainFrame, "Are you sure that you want .", "Confirm Exit", JOptionPane.YES_NO_OPTION);
                        if(result== JOptionPane.YES_OPTION)
                        {
                            JOptionPane.showMessageDialog(mainFrame, "Thanks for using .\nPlease :)", "From the Author", JOptionPane.INFORMATION_MESSAGE);

                            System.exit(0);

                       }

                  }

             }


             public void windowClosing(WindowEvent w)

             {

             JOptionPane.showMessageDialog(mainFrame,"Thanks:)","",JOptionPane.INFORMATION_MESSAGE);

             }


             // Class for lissoning to Look and feel radio menu events

             private class RadioListener implements ItemListener

             {

                  public void itemStateChanged(ItemEvent e)

                  {

                       JMenuItem item= (JMenuItem) e.getSource();

                       try

                       {

                            if(item== mnuTonicFeel&&mnuTonicFeel.isSelected())

             UIManager.setLookAndFeel("com.jgoodies.looks.plastic.PlasticXPLookAndFeel");
```

```
                if(item== mnuMetalFeel&&mnuMetalFeel.isSelected())

    UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");


                if(item== mnuMotifFeel&&mnuMotifFeel.isSelected())

    UIManager.setLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel");


                if(item== mnuWindowsFeel&&mnuWindowsFeel.isSelected())

    UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
                SwingUtilities.updateComponentTreeUI(mainFrame);

            }
            catch(Exception ex)
            {
                JOptionPane.showMessageDialog(mainFrame, "Oops!!\n"+ "Unable to load "+ item.getText()+ " Look and feel.", "Warning!", JOptionPane.WARNING_MESSAGE);
            }
        }
    }


    // Main method
```

```
        public static void main(String args[])
        {
                newMainClient();
        }}
```

EmbedeFileGUI.java code

```
package Hiding;

import enc.Sample;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Toolkit;
import java.awt.color.ColorSpace;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.image.BufferedImage;
import java.awt.image.ColorConvertOp;
import java.awt.image.IndexColorModel;
import java.awt.image.WritableRaster;
import java.io.File;
import java.io.IOException;
import java.util.Arrays;
import java.util.Hashtable;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JScrollPane;
import javax.swing.JSlider;
import javax.swing.JTextField;

/**
 * File Name = EmbedFileGUI.java Version 2.0.0 Class Name = EmbedFileGUI
 *
 * Copyright (c) 2005 - admin
 *
 * @author admin
 */
public class EmbedFileGUI extends JFrame implements ActionListener,
ItemListener {

   BackEndHandler client;
   private JLabel lblMaster, lblOutput, lblData, lblMasterSize, lblOutputSize,
lblMessage;
   ;
private JLabel lblMasterFileSize, lblOutputFileSize, lblStatus;
   private JLabel lblDataSize, lblDataFileSize;
   private JLabel lblCompression, lblPassword;
   private JCheckBox chkCompress, chkEncrypt;
   private JSlider sliderCompression;
   private JPasswordField txtPassword;
   private JTextField txtMasterFile, txtOutputFile, txtDataFile;
   private JScrollPane scrollPane;
   private JButton btnGo, btnHelp, btnCancel;
   private JButton btnChangeMasterFile, btnChangeOutputFile,
btnChangeDataFile;

   public EmbedFileGUI(BackEndHandler client)
   {
      super("Embedding file  " + CPT.VERSION + "  ");
      setDefaultCloseOperation(DISPOSE_ON_CLOSE);
      this.client = client;

      Font arialFont = new Font("Arial", Font.PLAIN, 11);
      lblMaster = new JLabel("Master file");
```

```
    lblOutput = new JLabel("Master file");
    lblData = new JLabel("Data file");
    lblMasterSize = new JLabel("Size: ");
    lblOutputSize = new JLabel("Size: ");
    lblDataSize = new JLabel("Size: ");
    txtMasterFile = new MyJTextField(client.getMasterFile().getName(), 13,
Color.blue, Color.lightGray);
    txtMasterFile.setEditable(false);
    lblMasterFileSize = new MyJLabel("" + client.getMasterFile().length() / 1024
+ " Kb", arialFont, Color.red, Color.gray);

    txtOutputFile = new MyJTextField(client.getOutputFile().getName(), 13,
Color.blue, Color.lightGray);
    txtOutputFile.setEditable(false);
    lblOutputFileSize = new MyJLabel(lblMasterFileSize.getText(), arialFont,
Color.red, Color.gray);

    txtDataFile = new MyJTextField(client.getDataFile().getName(), 13,
Color.blue, Color.lightGray);
    txtDataFile.setEditable(false);
    lblDataFileSize = new MyJLabel("" + client.getDataFile().length() / 1024 + "
Kb", arialFont, Color.red, Color.gray);

    btnChangeMasterFile = new JButton("Change");
    btnChangeOutputFile = new JButton("Change");
    btnChangeDataFile = new JButton("Change");
    btnGo = new JButton("    Go    ");
    btnHelp = new JButton("   Help  ");
    btnCancel = new JButton("  Close  ");

    lblCompression = new MyJLabel("Compression level", arialFont,
Color.black, Color.lightGray);
    lblPassword = new MyJLabel("Password (Minimum 8 chars)", arialFont,
Color.black, Color.lightGray);
    chkCompress = new JCheckBox("Compress");
    chkEncrypt = new JCheckBox("Encrypt");

    sliderCompression = new JSlider(0, 9, 5);
    sliderCompression.setForeground(Color.blue);
    sliderCompression.setPaintTicks(true);
    sliderCompression.setPaintLabels(true);
    sliderCompression.setSnapToTicks(true);
    sliderCompression.setMajorTickSpacing(1);
    Hashtable h = new Hashtable();
    h.put(new Integer(0), new JLabel("0".toString(), JLabel.CENTER));
    h.put(new Integer(5), new JLabel("5".toString(), JLabel.CENTER));
    h.put(new Integer(9), new JLabel("9".toString(), JLabel.CENTER));
    sliderCompression.setLabelTable(h);

    txtPassword = new JPasswordField(9);
    chkCompress.addItemListener(this);
    chkEncrypt.addItemListener(this);
    lblCompression.setEnabled(false);
    sliderCompression.setEnabled(false);
    lblPassword.setEnabled(false);
    txtPassword.setEnabled(false);


    // Setup panel file1
    JPanel file1 = new JPanel();
    new BoxLayout(file1, BoxLayout.X_AXIS);
    file1.add(lblMaster);
    file1.add(txtMasterFile);
    file1.add(lblMasterSize);
    file1.add(lblMasterFileSize);
    file1.add(btnChangeMasterFile);

    // Setup panel file2
    JPanel file2 = new JPanel();
    new BoxLayout(file2, BoxLayout.X_AXIS);
    file2.add(lblOutput);
    file2.add(txtOutputFile);
    file2.add(lblOutputSize);
    file2.add(lblOutputFileSize);
    file2.add(btnChangeOutputFile);

    // Setup panel file3
    JPanel file3 = new JPanel();
    new BoxLayout(file3, BoxLayout.X_AXIS);
    file3.add(lblData);
```

```
        file3.add(txtDataFile);
        file3.add(lblDataSize);
        file3.add(lblDataFileSize);
        file3.add(btnChangeDataFile);

        JPanel panelFiles = new JPanel();
        GridBagLayout gbl = new GridBagLayout();
        GridBagConstraints gbc = new GridBagConstraints();
        panelFiles.setLayout(gbl);
        JLabel lblFiller;
        gbc.anchor = gbc.WEST;
        gbc.gridx = 1;
        gbc.gridy = 1;
        gbl.setConstraints(file1, gbc);
        panelFiles.add(file1);
        lblFiller = new JLabel(" ");
        gbc.gridy = 2;
        gbl.setConstraints(lblFiller, gbc);
        panelFiles.add(lblFiller);
        gbc.gridy = 3;
        gbl.setConstraints(file2, gbc);
        panelFiles.add(file2);
        lblFiller = new JLabel(" ");
        gbc.gridy = 4;
        gbl.setConstraints(lblFiller, gbc);
        panelFiles.add(lblFiller);
        gbc.gridy = 5;
        gbl.setConstraints(file3, gbc);
        panelFiles.add(file3);
        panelFiles = UtilityOperations.createBorderedPanel(panelFiles, "Files", 19,
3);

        // Setup features panel 1
        JPanel panelFeatures1a = new JPanel();
        new BoxLayout(panelFeatures1a, BoxLayout.X_AXIS);
        panelFeatures1a.add(chkCompress);
        panelFeatures1a.add(lblCompression);

        JPanel panelFeatures1b = new JPanel();
        new BoxLayout(panelFeatures1b, BoxLayout.X_AXIS);
        panelFeatures1b.add(new MyJLabel("", arialFont, Color.blue,
Color.lightGray));
        panelFeatures1b.add(sliderCompression);
        panelFeatures1b.add(new MyJLabel("Choose Check Book If Encryption
Needs", arialFont, Color.blue, Color.lightGray));

        JPanel panelFeatures1 = new JPanel();
        gbl = new GridBagLayout();
        panelFeatures1.setLayout(gbl);
        gbc.gridx = 1;
        gbc.gridy = 1;
        gbl.setConstraints(panelFeatures1a, gbc);
        panelFeatures1.add(panelFeatures1a);
        gbc.gridy = 2;
        gbl.setConstraints(panelFeatures1b, gbc);
        panelFeatures1.add(panelFeatures1b);
        panelFeatures1 = UtilityOperations.createBorderedPanel(panelFeatures1, "",
10, 3);

        // Setup features panel 2
        JPanel panelFeatures2 = new JPanel();
        new BoxLayout(panelFeatures2, BoxLayout.X_AXIS);
        panelFeatures2.add(chkEncrypt);
        panelFeatures2.add(lblPassword);
        panelFeatures2.add(txtPassword);
        panelFeatures2 = UtilityOperations.createBorderedPanel(panelFeatures2,
"Encryption", 3, 3);

        JPanel panelFeatures = new JPanel();
        gbl = new GridBagLayout();
        panelFeatures.setLayout(gbl);
        gbc.anchor = gbc.WEST;
        gbc.gridx = 1;
        gbc.gridy = 1;
        gbl.setConstraints(panelFeatures1, gbc);
        panelFeatures.add(panelFeatures1);
        lblFiller = new JLabel(" ");
        gbc.gridy = 2;
        gbl.setConstraints(lblFiller, gbc);
        panelFeatures.add(lblFiller);
```

```
gbc.gridy = 3;
gbl.setConstraints(panelFeatures2, gbc);
panelFeatures.add(panelFeatures2);

// Setup the buttons panel
JPanel panelButtons = new JPanel();
gbl = new GridBagLayout();
panelButtons.setLayout(gbl);
gbc.anchor = gbc.WEST;
gbc.gridx = 1;
gbc.gridy = 1;
gbl.setConstraints(btnGo, gbc);
panelButtons.add(btnGo);
lblFiller = new JLabel(" ");
gbc.gridy = 2;
gbl.setConstraints(lblFiller, gbc);
panelButtons.add(lblFiller);
gbc.gridy = 3;
gbl.setConstraints(btnHelp, gbc);
panelButtons.add(btnHelp);
lblFiller = new JLabel(" ");
gbc.gridy = 4;
gbl.setConstraints(lblFiller, gbc);
panelButtons.add(lblFiller);
gbc.gridy = 5;
gbl.setConstraints(btnCancel, gbc);
panelButtons.add(btnCancel);

JPanel mainPanel = new JPanel();
new BoxLayout(mainPanel, BoxLayout.Y_AXIS);
mainPanel.add(panelFiles);
mainPanel.add(new JLabel(" "));
mainPanel.add(panelFeatures);
mainPanel.add(new JLabel(" "));
mainPanel.add(panelButtons);
mainPanel = UtilityOperations.createBorderedPanel(mainPanel, 3, 3);
setContentPane(mainPanel);

btnChangeMasterFile.addActionListener(this);
btnChangeOutputFile.addActionListener(this);
```

```
   btnChangeDataFile.addActionListener(this);
   btnHelp.addActionListener(this);
   btnGo.addActionListener(this);
   btnCancel.addActionListener(this);

   Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
   int width = (int) (0.7 * d.width);
   int height = (int) (0.85 * d.height);
   setSize(width, height);
   setLocation((d.width - width) / 2, (d.height - height) / 2);
   //setResizable(false);
   setVisible(true);
   txtOutputFile.setVisible(false);
   lblOutput.setVisible(false);
   lblOutputFileSize.setVisible(false);
   btnChangeOutputFile.setVisible(false);
   chkCompress.setVisible(false);
   sliderCompression.setVisible(false);
   lblCompression.setVisible(false);
}

public BufferedImage grayOut(BufferedImage img) {
   ColorConvertOp colorConvert = new ColorConvertOp(ColorSpace
         .getInstance(ColorSpace.CS_GRAY), null);
   colorConvert.filter(img, img);

   return img;
}

/**
 * This method reads an image from the file
 *
 * @param fileLocation -- > eg. "C:/testImage.jpg"
 * @return BufferedImage of the file read
 */
public BufferedImage readImage(String fileLocation) {
   BufferedImage img = null;
   try {
      img = ImageIO.read(new File(fileLocation));
   } catch (IOException e) {
```

```
            e.printStackTrace();
        }
        return img;
    }

    /**
     * This method writes a buffered image to a file
     *
     * @param img -- > BufferedImage
     * @param fileLocation --> e.g. "C:/testImage.jpg"
     * @param extension --> e.g. "jpg","gif","png"
     */
    public void writeImage(BufferedImage img, String fileLocation,
            String extension) {
        try {
            BufferedImage bi = img;
            File outputfile = new File(fileLocation);
            ImageIO.write(bi, extension, outputfile);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void actionPerformed(ActionEvent e) {
        Object source = e.getSource();

        if (source == btnChangeMasterFile) {
            client.chooseMasterFile();
            txtMasterFile.setText(client.getMasterFile().getName());
            lblMasterFileSize.setText("" + client.getMasterFile().length() / 1024 +
"Kb");
            lblOutputFileSize.setText(lblMasterFileSize.getText());
        }

        if (source == btnChangeOutputFile) {
            client.chooseOutputFile();
            txtOutputFile.setText(client.getOutputFile().getName());
        }

        if (source == btnChangeDataFile) {
            client.chooseDataFile();
            txtDataFile.setText(client.getDataFile().getName());
            lblDataFileSize.setText("" + client.getDataFile().length() / 1024 + "Kb");
        }

        if (source == btnHelp) {
            JOptionPane.showMessageDialog(this, "Sorry", "Invalid password!",
JOptionPane.WARNING_MESSAGE);
        }

        if (source == btnCancel) {
            dispose();
        }

        if (source == btnGo) {
            int compression = sliderCompression.getValue();
            String password = null;

            if (chkEncrypt.isSelected()) {
                password = new String(txtPassword.getPassword());
                if (password.length() < 8) {
                    JOptionPane.showMessageDialog(this, "Password needs to be a
minimum of 8 Characters!", "Invalid password!",
JOptionPane.WARNING_MESSAGE);
                    return;
                }
            }

            if (client.getOutputFile().exists()) {
                int result2 = JOptionPane.showConfirmDialog(null, "File " +
client.getOutputFile().getName() + " already exists!\nWould you like to
OVERWRITE it?", "File already exists!", JOptionPane.YES_NO_OPTION);
                if (!(result2 == JOptionPane.YES_OPTION)) {
                    if (!client.chooseOutputFile()) {
                        return;
                    }
                }
            }

            if (CPT.embedFile(client.getMasterFile(), client.getOutputFile(),
client.getDataFile(), compression, password)) {
                new Sample().Add(client.getOutputFile().getName(), (password).trim());
                JOptionPane.showMessageDialog(this, CPT.getMessage(), "Operation
Successful", JOptionPane.INFORMATION_MESSAGE);
            } else {
                JOptionPane.showMessageDialog(this, CPT.getMessage(), "Operation
Unsuccessful", JOptionPane.ERROR_MESSAGE);}}}
    public void itemStateChanged(ItemEvent e) {
        if (e.getSource() == chkCompress) {
            if (chkCompress.isSelected()) {
                lblCompression.setEnabled(true);
                sliderCompression.setEnabled(true);} else {
                lblCompression.setEnabled(false);
                sliderCompression.setEnabled(false) } else {
            if (chkEncrypt.isSelected()) {
                lblPassword.setEnabled(true);
                txtPassword.setEnabled(true); } else {
                lblPassword.setEnabled(false);
                txtPassword.setEnabled(false);
            }}}}
```

5.5 SAMPLE SCREEN SHOTS

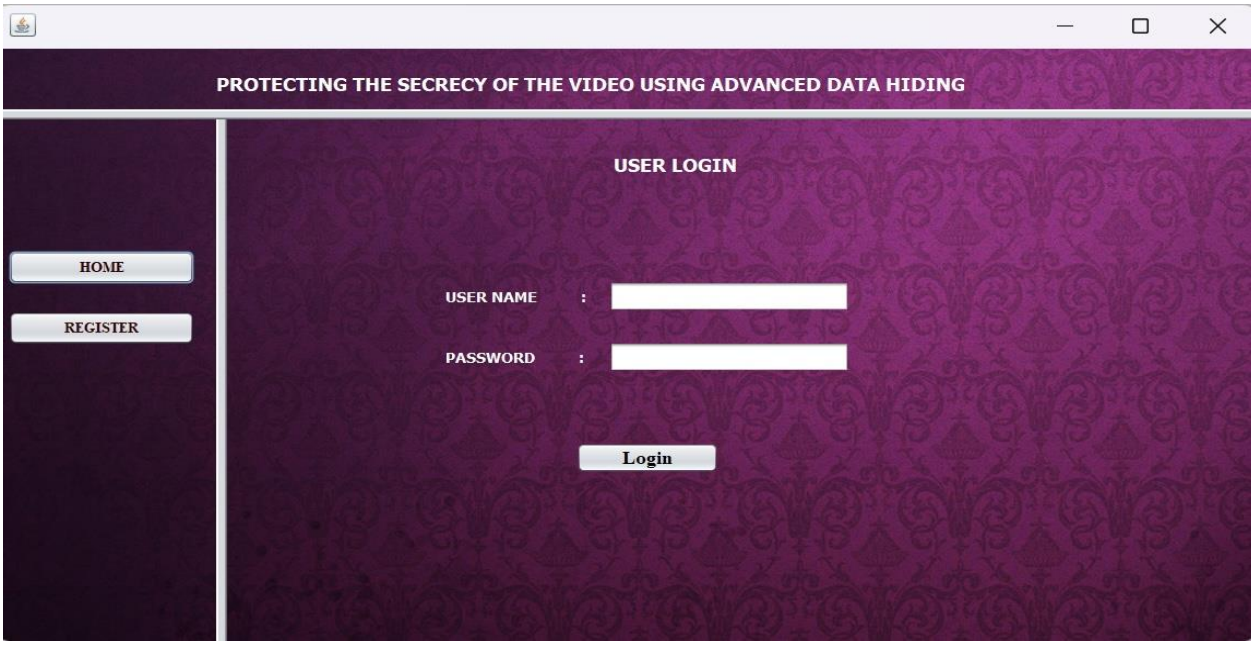
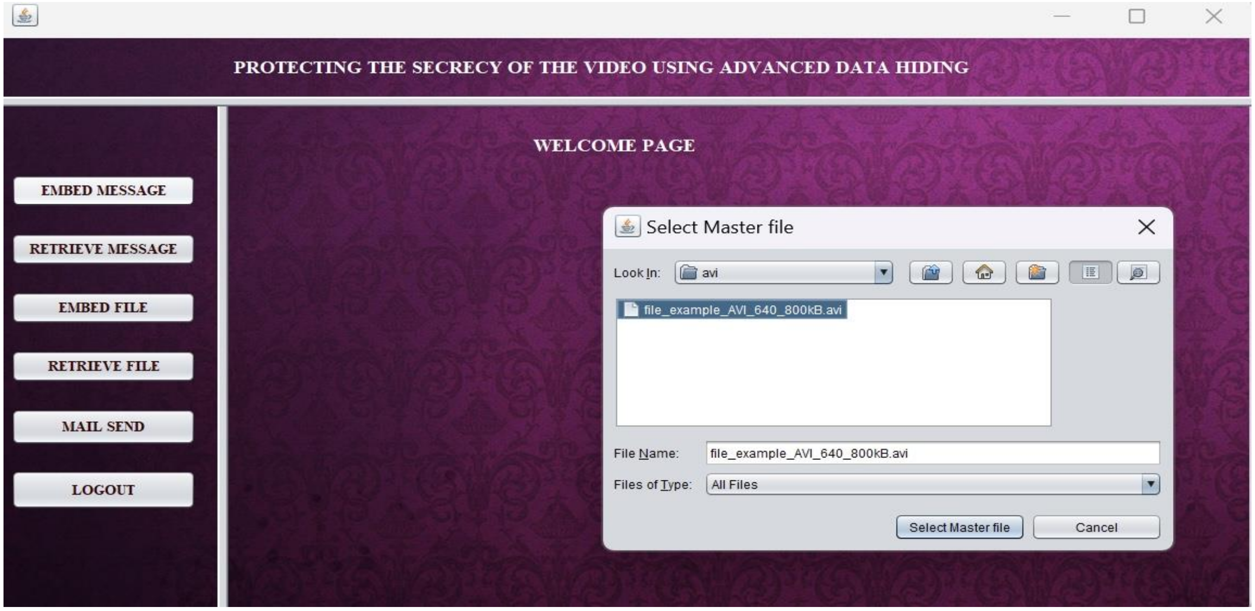

PROTECTING THE SECRECY OF THE VIDEO USING ADVANCED DATA HIDING
USER LOGIN
HOME
REGISTER
USER NAME :
PASSWORD :
Login

Fig. 5.5.1 Login Page

User can first register and then login by using the user credentials.



Look In: avi

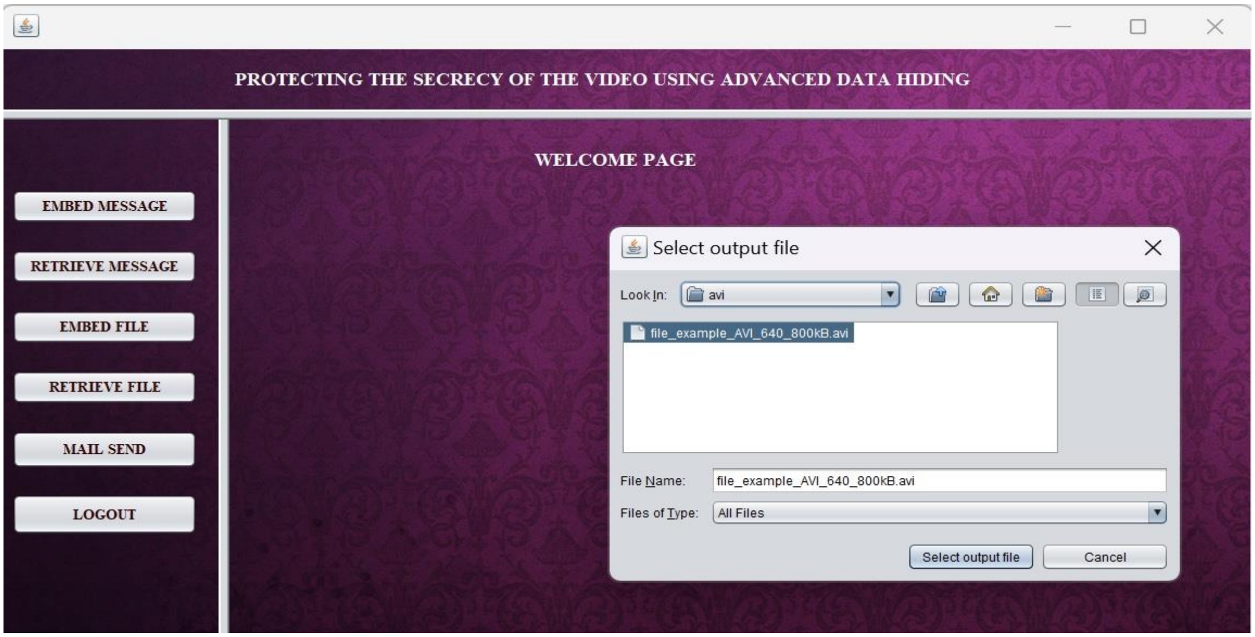
file_example_AVI_640_800kB.avi

File Name: file_example_AVI_640_800kB.avi

Files of Type: All Files

Fig. 5.5.2 Selecting Master Video

Select the master video and the output video which need to be sent by embedding the data.



Look In: avi

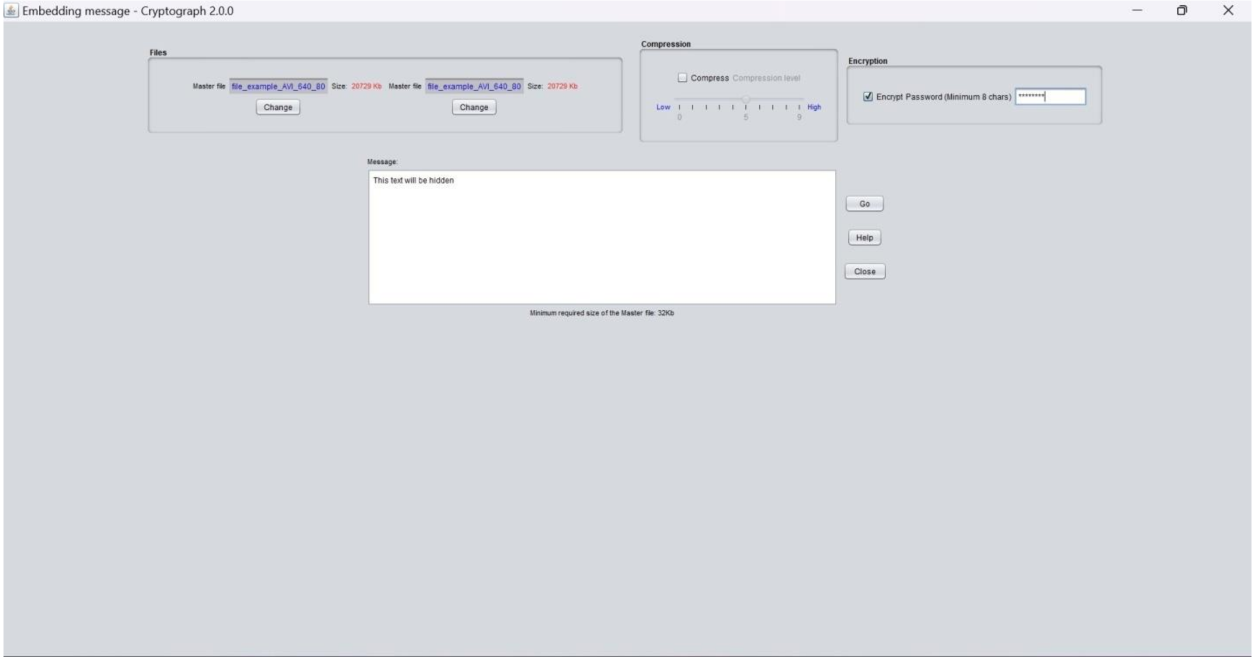
file_example_AVI_640_800kB.avi

File Name: file_example_AVI_640_800kB.avi

Files of Type: All Files

Fig 5.5.3 Selecting Output Video

Give the text message need to be hidden in the master file given and secure the file using key


Embedding message - Cryptograph 2.0.0

Files

Master file: file_example_AVI_640_80 Size: 20729 Kb Master file: file_example_AVI_640_80 Size: 20729 Kb

Compression

□ Compress Compression level

Encryption

☑ Encrypt Password (Minimum 8 chars) ********

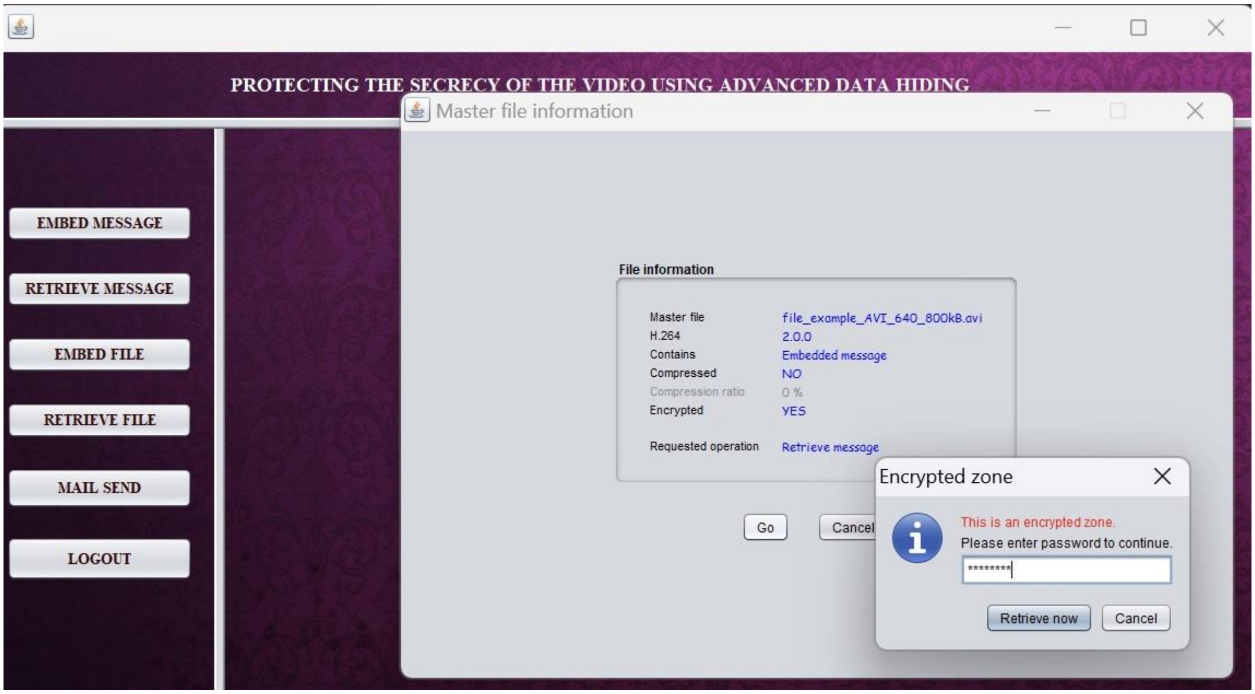
Message:

This text will be hidden

Minimum required size of the Master file: 32Kb

Fig. 5.5.4 Enter Text Message to be Hidden

Retrieve the text message by selecting the output file and enter the key to view the hidden message.



File information

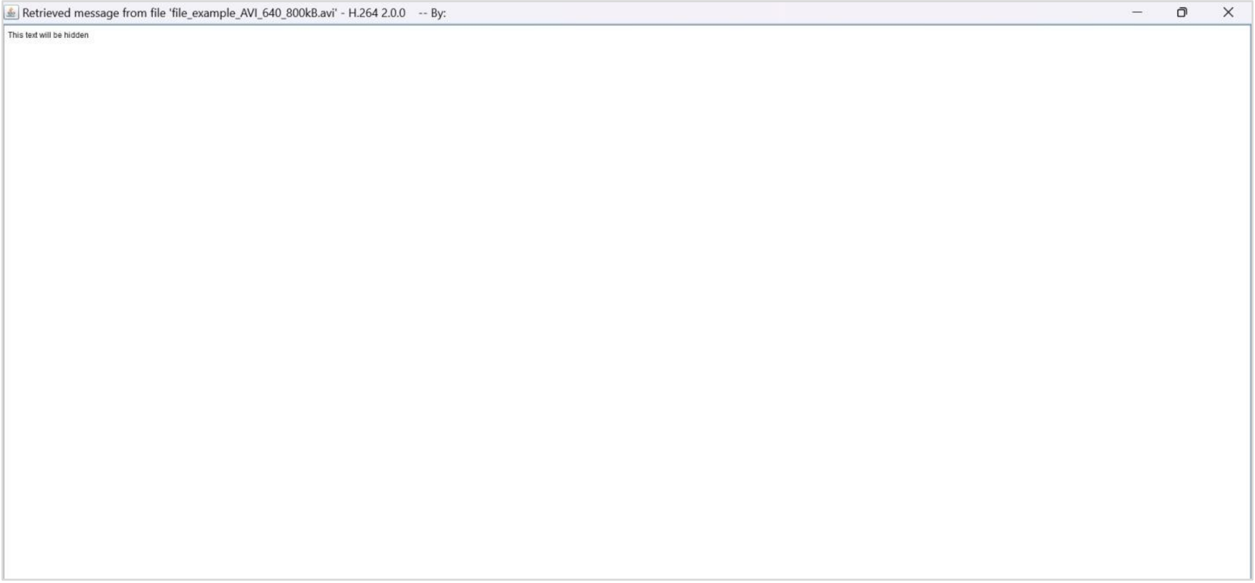
Master file file_example_AVI_640_800kB.avi
H.264 2.0.0
Contains Embedded message
Compressed NO
Compression ratio 0 %
Encrypted YES

Requested operation Retrieve message

This is an encrypted zone.
Please enter password to continue.
****

Fig 5.5.5 Retrieve Text Message

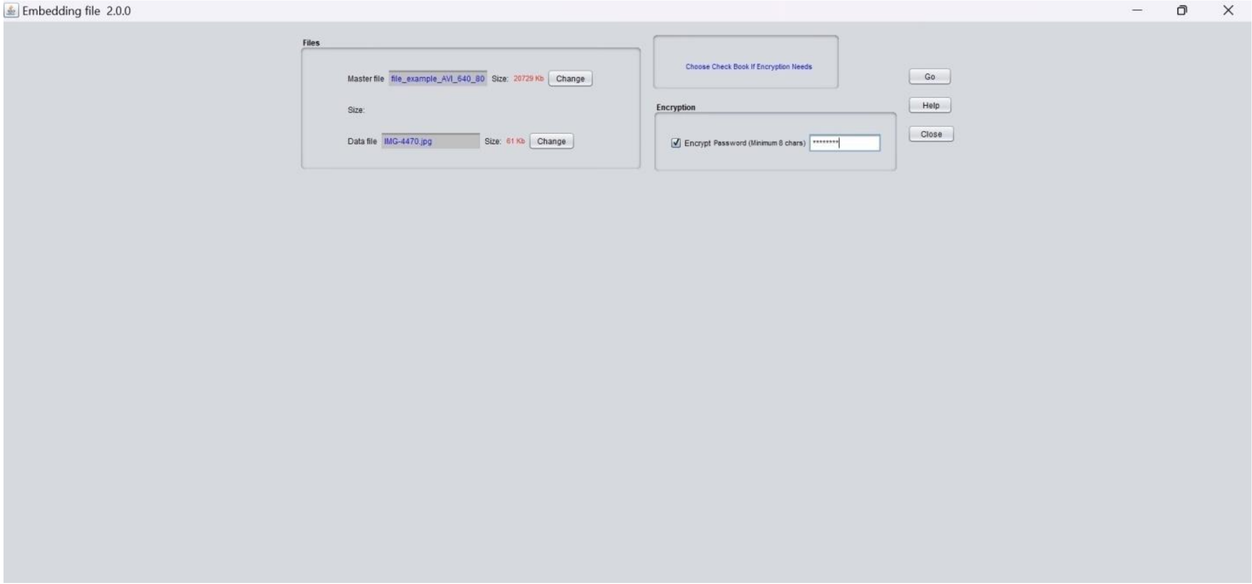
Now the hidden information is separated from the video and the text message is displayed.


Retrieved message from file 'file_example_AVI_640_800kB.avi' - H.264 2.0.0 -- By:

This text will be hidden

Fig. 5.5.6 Hidden Message is Desplayed

To hide an image select the master ,output and the image needed to be hidden and secure it with key



Embedding file 2.0.0

Files

Master file file_example_AVI_640_80 Size: 20729 Kb

Size:

Data file IMG-4470.jpg Size: 61 Kb

Choose Check Book If Encryption Needs

Encryption

☑ Encrypt Password (Minimum 8 chars) ********

Fig. 5.5.7 Enter Key

Retrieve the image from the file by entering the key. The hidden image is extracted from the video

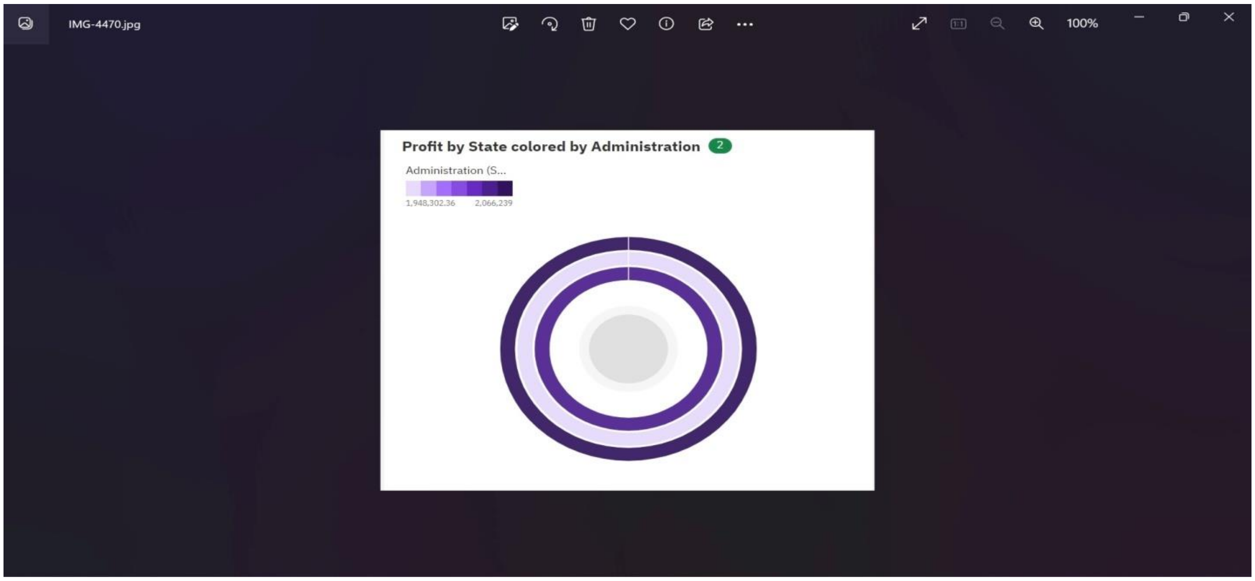


Fig 5.5.8 Hidden Image

To embed a video in a video select a video as a master file and as a output file. The video that need to be hidden is selected as data file. By giving a key and go option the video is embedded into the video, successful message is displayed. To separate the hidden video, give the key to view the original video.

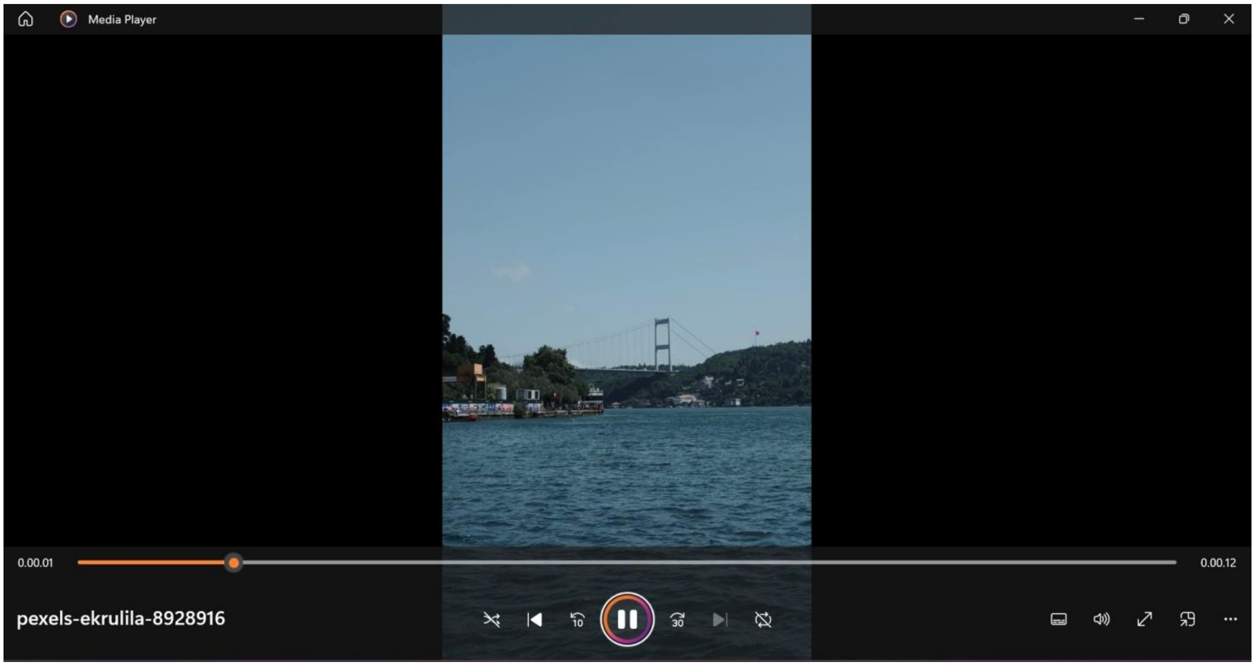


Fig. 5.5.9 Output

The hidden video is displayed.

TESTING AND MAINTENANCE

6.1 SOFTWARE TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 DEVELOPING METHODS

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.

The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

6.3 TYPES OF TESTING

6.3.1 UNIT TEST

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.3.2 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

6.3.3 SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.3.4 PERFORMANCE TEST

The Performance test ensures that the output is produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

6.3.5 INTEGRATION TEST

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

6.3.6 ACCEPTANCE TEST

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

- The Acknowledge will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updating process

Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

CONCLUSION

7.1 CONCLUSION

A data hiding technique for high resolution video by hiding the encrypted text over the video our intension is to provide proper protection on data during transmission. For the accuracy of the correct message output that extract from source we can use a tools for comparison and statistical analysis can be done. It is highly configurable, thus it may result in high data capacities to retrieve the original data. Finally, it can be easily extended, resulting in better robustness, better data security and higher embedding capacity. The general trend of information hiding in the compressed video domain was presented.Then, categorized the existing information hiding methods based on the venues at which they operate and highlighted their strengths and weaknesses. Also a motion vector based information hiding method is explained

7.2 FUTURE ENCHANCEMENT

The project can be further improved in the future, by using other video format such as the most commonly used mp4 format. More research and studies need to be put forward to secure the information. By expanding the research it is also possible to hide multiple data in a single video. After the extraction of the information from the video the entire content can be automatically deleted from the existing video. The passing of information may be sent through other networks rather than depending on a single network which may vulnerable after some period of time. With more thought and study the above mentioned ways can be implemented in the future. If in future the avi file format is used then the duration of the avi file can be made more as we used avi file of 30 seconds.

REFERENCE:

1. Anushiadevi, R., & Amirtharajan, R. (2023). Separable reversible data hiding in an encrypted image using the adjacency pixel difference histogram. Journal of Information Security and Applications, 72, 103407.

2. Fan, G., Pan, Z., Zhou, Q., & Zhang, X. (2023). Flexible patch moving modes for pixel-value-ordering based reversible data hiding methods. Expert Systems with Applications, 214, 119154.

3. Muazu, A. A., Maiwada, U. D., Garba, A. R. I., Qabasiyu, M. G., & Danyaro, K. U. (2023, January). Secure Data Hiding and Extraction Using RSA Algorithm. In Advancements in Interdisciplinary Research: First International Conference, AIR 2022, Prayagraj, India, May 6–7, 2022, Revised Selected Papers (pp. 14-28). Cham: Springer Nature Switzerland.

4. Hameed, M. A., Abdel-Aleem, O. A., & Hassaballah, M. (2022). A secure data hiding approach based on least-significant-bit and nature-inspired optimization techniques. Journal of Ambient Intelligence and Humanized Computing, 1-19.

5. Hassan, F. S., & Gutub, A. (2022). Improving data hiding within colour images using hue component of HSV colour space. CAAI Transactions on Intelligence Technology, 7(1), 56-68.

6.Pei, L. (2022). Research on Digital ImageWatermarking Algorithm Based on Scrambling andSingular Value Decomposition. Journal of Mathematics, 2022.

7.Boujerfaoui, S.., Riad, R., Douzi, H., Ros, F., & Harba, R. (2022). Image Watermarking between Conventional and Learning-Based Techniques: A LiteratureReview. Electronics, 12(1), 74.

8.Chen, Y., Zhou, L., Zhou, Y., Chen, Y., Hu, S., & Dong,Z. (2021). Multiple Histograms Shifting-Based Video Data Hiding Using Compression Sensing. IEEE Access, 10, 699-707.

9.Wang, X., Che, Z., Jiang, B., Xiao, N., Yang, K., Tang,J., ... & Qi, Q. (2021). Robust unsupervised video anomaly detection by multipath frame prediction.IEEE transactions on neural networks and learning systems, 33(6), 2301-2312.

10.Xiao, S., Zhang, Z., Zhang, Y., & Yu, C. (2020).Multipurpose watermarking algorithm for medical images. Scientific Programming, 2020, 1-13.

11. Sharma, S.; Zou, J.J.; Fang, G. A Novel Multipurpose Watermarking Scheme Capable of Protecting and Authenticating Images With Tamper Detection and Localisation Abilities. IEEE Access 2022, 10, 85677–85700. [CrossRef]

12. Wazirali, R.; Ahmad, R.; Al-Amayreh, A.; Al-Madi, M.; Khalifeh, A. Secure watermarking schemes and their approaches in the IoT technology: An overview. Electronics 2021, 10, 1744. [CrossRef]

13. Amrit, P.; Singh, A.K. Survey on watermarking methods in the artificial intelligence domain and beyond. Comput.Commun. 2022, 188, 52–65. [CrossRef]

14. Sy, N.C.; Kha, H.H.; Hoang, N.M. An efficient robust blind watermarking method based on convolution neural networks in wavelet transform domain. Int. J. Mach. Learn. Comput 2020, 10, 675–684. [CrossRef]

15. Wang, X.; Ma, D.; Hu, K.; Hu, J.; Du, L. Mapping based residual convolution neural network for non-embedding and blind image watermarking. J. Inf. Secur. Appl. 2021, 59, 102820. [CrossRef]

16. UshaNandini, D.; Divya, S.A literature survey on various watermarking techniques. In Proceedings of the 2017 International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January 2017; pp. 1–4. [CrossRef]

17. Rani, B.U.; Praveena, B.; Ramanjaneyulu, K. Literature Review on Digital Image Watermarking; ICARCSET ’15; Association for Computing Machinery: New York, NY, USA, 2015. [CrossRef]

18. Muthulakshmi, K., &amp; Valarmathi, K. (2020). Privacy Preserving and Sensitive Attribute BasedCloud Storage: A Survey.