# Encryption, Watermarking and Steganography in Application to Biometrics

**Hitha Meka**

**Problem Report submitted to the**
**College of Engineering and Mineral Resources**
**at West Virginia University**
**in partial fulfillment of the requirements**
**for the degree of**

**Master of Science**

**in**

**Electrical Engineering**

**Natalia A. Schmid, Ph.D., Chair**
**Bojan Cukic, Ph.D.**
**Donald A. Adjeroh, Ph.D.**
**Xin Li, Ph.D.**

**Lane Department of Computer Science and Electrical Engineering**

**Morgantown, West Virginia**
**2007**

**Keywords: Security, Encryption, Watermarking, Steganography, Iris**

# ABSTRACT

## Encryption, Watermarking and Steganography in Application to Biometrics

**Hitha Meka**

With the rapid growth of networked systems, digital data is subject to illegal copying, forgery and unauthorized distribution. To provide protection and privacy to the biometric data, when transmitted through an unsecured communication channel, techniques such as encryption, watermarking and steganography have been studied which can be used to achieve security. Encryption makes the sensitive data meaningless to an interceptor while any changes to the sensitive data can be detected by using watermarking techniques. Steganography is used to hide sensitive data in an unsuspicious image. Least significant bit method, which is a spatial domain method, and comparison of mid-band DCT coefficients, which is a frequency domain method, have been implemented. The techniques were evaluated for their robustness against signal processing attacks such as Gaussian noise, rotation and compression and results have been presented. Two steganographic methods which belong to spatial domain, one method which hides text in an image and another method which hides image in an image have been implemented and their robustness against signal processing attacks have been examined and results have been presented. Finally, the security achieved by combining two of the above techniques (encryption and watermarking or steganography) has been studied.

This thesis is dedicated to my family,
Rajendra Prasad Meka, Usha Rani Meka, Siva Naga Chaitanya Meka
and
In the loving memory of my grandfather, Mikkilineni Basavaiah
who have made this possible.

# Acknowledgments

I would like to thank my advisor, Dr. Natalia A. Schmid, for her guidance throughout my research. I appreciate her help, time and support in guiding me towards completing my work.

I am grateful to Dr.Bojan Cukic, Dr. Donald A. Adjeroh and Dr. Xin Li for being on my committee and guiding me.

I thank my roommates and friends who have supported me and helped me to get through difficult times.

Finally, I thank my parents and brother for their immense support and love. All this wouldn’t have been possible without their support.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 What is biometrics?

Traditional techniques for identification and verification are based on **'What you know'** such as passwords, PINS which can be forgotten and **'What you have'** such as tokens, ID which can be stolen or lost. Biometric identification is based on physiological or behavioral characteristics that provide information about **'Who you are'**. These characteristics are hard to forget or lose.

Biometrics is an automatic recognition of individuals based on their physiological and/or behavioral characteristics. Based on requirements of an application, a biometric system can operate in identification or verification mode. For an individual, using a biometric to be identified or verified, a copy of the individual's signature has to be stored in a database at the enrollment stage. Figure 1.1 presents the block-diagram of the enrollment stage.

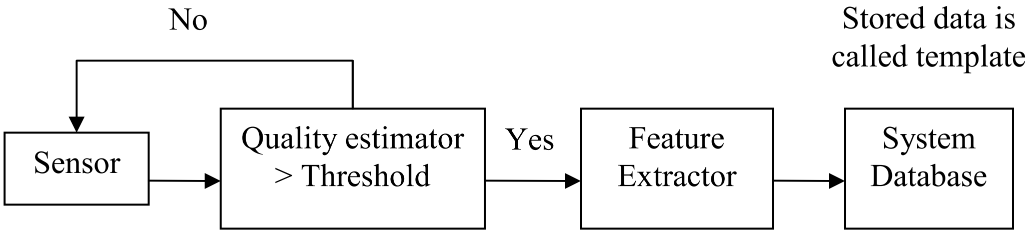


Figure 1.1: Enrollment stage [1].

In the Verification mode, the biometric system validates the identity of the person by comparing a query template with his/her own biometric template stored in the system database. It is a one-to-one comparison. Figure 1.2 describes the steps of the Verification stage.

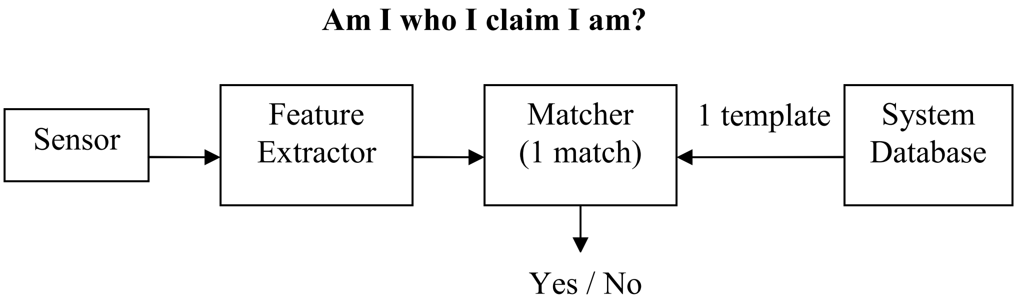


Figure 1.2: Verification mode [1]

In Identification mode, the biometric system recognizes an individual by searching the templates of all users in the database for a match. It is 1-to-n comparisons. Figure 1.3 presents the block-diagram of the identification stage.

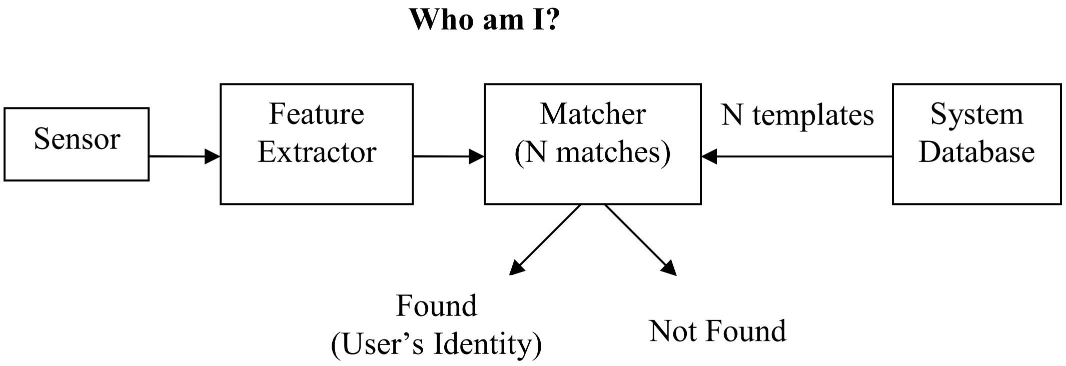


Figure 1.3: Identification mode [1]

## 1.2 Commonly used biometrics

Most commonly used biometrics are fingerprints, iris and face.

### Fingerprints

A fingerprint is a pattern of a series of ridges and valleys on the surface of the fingertip, which are formed during the first seven months of fetal development. Fingerprint recognition is highly accepted and the performance is sufficiently accurate for small and medium scale systems with low cost. Fingerprints are unique, even identical twins have different fingerprints. A small number of populations may exhibit identification which is not satisfactory because of genetic factors, aging, cuts and bruises on their fingers.

## Iris

The iris is the annular region of the eye bounded by the pupil and the sclera. The visible iris texture is formed during the fetal development and stabilizes during the first two years. In the past 2 decades, iris biometrics was not as highly accepted as the fingerprints, since iris biometrics is relatively new, while fingerprint was used in forensic applications more than 150 years. Also, there is a concern that iris carries some medical information about health of an individual. Apart from poor acceptance, the performance of iris system is accurate for large scale systems and cost effective. As fingerprints, iris of identical twins are unique and even the iris of left eye is different from the right eye.

## Face

Face is the most common biometric characteristic used by humans for recognition. Face recognition is highly accepted with moderate performance. This system has many restrictions on the templates obtained based on the illumination, pose and expression. If there is a slight difference in any one or more of the above three properties, automatic recognition becomes challenging.

## Other Biometrics

Hand geometry uses the measurements of the hand which are not unique when used for large scale systems. Gait is the way one walks. It does not remain invariant due to change in body weight, inebriety, etc. Signature is the way a person signs his/her name which changes over a period of time due to many reasons. There are other biometrics such as keystroke, voice, palm print DNA, thermal image of face which are not commonly used.

Biometrics are hard to forge but biometric signatures can be stolen. They are unique identifiers, but they are not secrets. Biometric-based identification has many advantages when compared to traditional identification techniques, but the problem of ensuring the security and integrity of the biometric data is critical. If biometric data is stolen, it remains stolen for the whole life. Biometrics can be used as a powerful identifier only if the connection between the readers to the verifier is secure.

## 1.3 Attacks on biometric systems

The use of biometrics for access control applications is increasing rapidly. For security of critical applications, the biometric systems should withstand different types of attacks. If the system is constantly supervised, then it would be difficult for the hackers to attack the system, but if the system is unsupervised or remote, then the hackers will have ample of time to deceive the system.

The hackers can attack the biometric system in eight different ways. Figure 1.4 displays a block-diagram indicating where various attacks can occur.

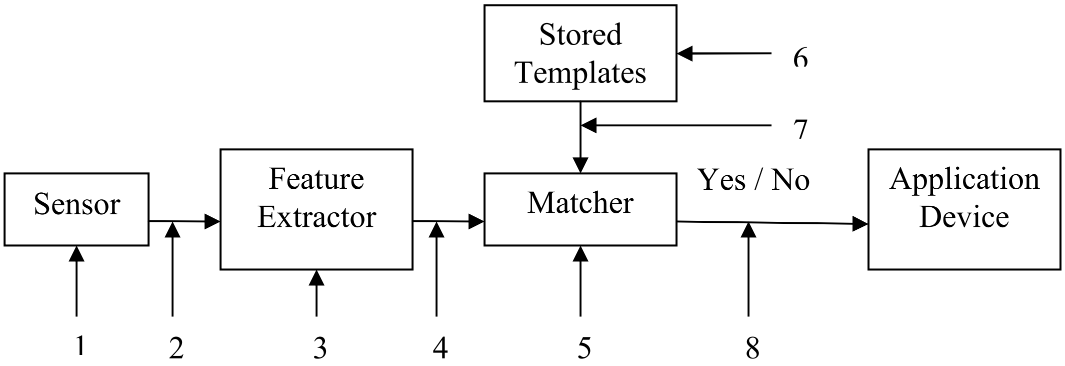


Figure 1.4: Attacks on a generic biometric system [3]

**First attack:** Attack on the sensor. Sensor can be overridden by presenting fake biometrics. Like a fake finger, face mask or a copy of signature.

**Second attack:** Attack on the channel between the sensor and the feature extractor. Biometrics which was submitted can be resubmitted or replayed by bypassing the sensor. Like an old copy of fingerprint or face image.

**Third attack:** Attack on the feature extractor. Feature extractor can be override by attacking it and forcing it to produce feature values selected by the hacker.

**Fourth attack:** Attack on the channel between the feature extractor and the matcher. Features extracted by the extractor can be replaced by a different feature set. This type of attack is difficult because the feature extractor and matcher are not separate. This attack is possible only if the matcher is remote and the features extracted have to be sent to the matcher for matching purpose.

**Fifth attack:** Attack on the matcher. Matcher can be overridden by attacking it and forcing it to produce high or low matching score irrespective of the input.

**Sixth attack:** Attack on the stored database. The database can be local or remote. Templates which are stored at the time of enrollment can be attacked by modifying one or more templates in the database. This could result in fraudulent authorization of an individual or a denial of service.

**Seventh attack:** Attack on the channel between the system's database and the matcher. Respective template is selected and sent through a channel to the matcher for identification. This template can be changed accordingly by the hacker.

**Eighth attack:** Attack on the channel between the matcher and application device. The decision whether the user can access the application device can be changed by the hacker accordingly.

These attacks threaten the security of biometric systems. To overcome this problem, the security of the system has to be increased. This can be achieved by using techniques such as encryption, watermarking and steganography on the data that has to be stored in a system's database or transmitted through an unsecured communication channel.

## Encryption

Encryption is a process of perturbing or modifying information to make it unreadable without special knowledge, also called scrambling. Encrypted data appears to be meaningless to an interceptor who tries to get the sensitive data. For identification and verification, the encrypted templates are decrypted and matched with the template obtained online. Encryption does not provide security when the templates are decrypted for matching.

## Watermarking

In digital watermarking, the proprietary information is embedded in the host data to protect the intellectual property rights of that data. It provides protection against illegal utilization of biometric data. There are two types of watermarking which are based on domain of application. In spatial domain, the pixel values in the image are changed. In spectro-transform domain, watermark signal is added to the host image.

### Steganography

Steganography means secret communication in Greek. Steganography is hiding important data in an unsuspected carrier image. Fingerprint minutiae are extracted and are hidden in insignificant pixels of the cover image like a face image, iris image or an image which is unsuspected. And this cover image is transmitted through an unsecured communication channel. Even if the hacker intercepts the cover image, he/she will not know that fingerprint minutiae are hidden in the cover image.

Combination of two techniques provides more security rather than a single technique. Encryption can be combined with watermarking or steganography, combining the advantages of the above techniques provides more security.

## 1.4 Motivation

To transmit sensitive data through an unsecured communication channel where an interceptor can hack the data, certain techniques can be used to protect the data. By encrypting the data before transmitting it, even if an unauthorized user intercepts the channel and gets the sensitive data, he/she will not be able to read the data without decrypting it using proper decryption algorithm and key. Using encryption, security and privacy of the sensitive data is ensured [6]. Digital files are subjected to illegal copying, forgery and unauthorized distribution [18]. Digital watermarking techniques can be used to detect where and by how much a digital file has been tampered. Protecting the copyrights of the owner of the digital file can be ensured by using watermarking [17]. Secret data to be transmitted can be hidden in an unsuspicious carrier and can be transmitted. Hiding the very occurrence of the communication itself along with the secret data can be achieved using steganography [29]. Combination of two techniques such as encryption and watermarking or encryption and steganography increases the security, protection and privacy of the sensitive data.

## 1.5 Contributions

Global survey has been conducted for protecting the biometric data from an interceptor. Proving security and privacy have been studied for secured transactions by using encryption, watermarking and steganographic techniques.

Watermarking in spatial and frequency domain has been implemented and the robustness of the techniques is evaluated by signal processing attacks. If the watermarked image withstands the attacks by providing means to recover the embedded watermark completely, then the technique is considered robust.

Spatial domain steganography which hides text in an image and image in another image has also been implemented and the robustness is evaluated by attacking the stego image with signal processing techniques.

Combination of the above techniques has also been studied to provide more security to the biometric template which can be attacked in a biometric system as seen in Section 1.3.

## 1.6 Organization of the report

Chapter 2 explains the basic encryption and decryption system. The reason for using encryption, its applications and also explains briefly the different techniques of encryption. It also evaluates the encryption techniques applied to fingerprint, iris and face biometrics.

Chapter 3 presents the basic watermark embedding and extracting system along with the purpose of watermarking and its uses in different fields. Different types of watermarking methods have been explained briefly. It also evaluates spatial domain and fragile invisible watermarking techniques on fingerprints. Digital watermarking technique is explained on a multi-modal system using face and iris templates. Finally the technique which was implemented in spatial domain and frequency domain has been explained continued with the results obtained for both the techniques. Robustness of the technique to signal processing attacks is evaluated.

Chapter 4 gives a basic steganographic embedding and extracting system, its purpose and applications have been mentioned. Various types of steganographic methods and attacks on the steganographic system have been explained along with the explanation of the spatial domain technique used with face biometrics. Finally explaining the two spatial domain methods implemented, one hiding the sensitive text in an image and other hiding an image in another image. Presented the results for both the techniques and evaluated the robustness to signal processing attacks.

Chapter 5 explains the techniques used to increase the security level by combining two of the above techniques. It also concludes the work and provides suggestions for any future work.

# Chapter 2

# Encryption

## 2.1 What is encryption?

Encryption also called scrambling is a process of perturbing information to make it unreadable without special knowledge [4]. The original data is called plaintext. Encryption converts the plaintext into ciphertext. Cipher is an algorithm for performing encryption and decryption, which is a sequence of defined steps that are followed as a procedure [9]. A key is selected for encrypting a message or data by a cipher. Figure 2.1 shows the encryption process.

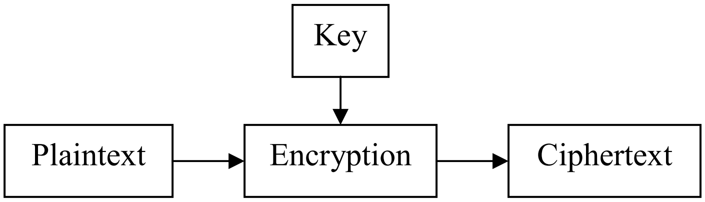


Figure 2.1: Encryption process

Ciphertext contains all the information of the plaintext message, but it is not in a format readable by a human or computer without decrypting it. It is a random data.

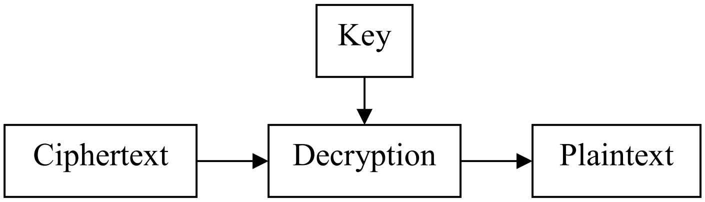


Figure 2.2: Decryption process

Plaintext is obtained by deciphering the ciphertext using a key. Key used for decryption can be the same key used for encryption or a different key based on the cipher followed for encrypting the plaintext. The decryption process is shown in Figure 2.2.

## 2.2 Purpose of encryption (or) Why encryption?

In order to send sensitive data, the communication must be secured. By using encryption, security and privacy of the data can be ensured [6]. Even if an unauthorized user intercepts the communication channel and gets the sensitive data, he/she will not be able to read it without decrypting it using proper cipher and key.

## 2.3 Applications

Encryption was used to protect communication for centuries. In early days, encryption was not used by the public but was used for military purpose by the government. In the mid 1970s, new reliable encryption methods were developed. In present days, encryption is widely used in providing secure voice and data communication and protecting stored information [8]. It is used in secure electronic transactions such as e-mails, internet e-commerce and marketing. It is also used in banking, i.e., bank automatic teller machines (ATM). Other applications are in health care, mobile telephone networks and in wireless communication as they are easy to tap [7, 10].

## 2.4 Types of ciphers

Ciphers are sub-divided into classical, rotor machines and modern ciphers. (See Figure 2.3 for details) Ciphers used in olden days are classical ciphers which are different from modern ciphers in the way they operate.

### 2.4.1 Classical cipher

This cipher was used in olden days and is not used now. It operates on alphabets (A –Z), which is implemented by hand or by simple mechanical devices. This cipher is more prone to ciphertext attacks and decryption can be done by having the knowledge of frequency analysis. Classical ciphers are sub-divided into substitution cipher and transposition cipher [6].

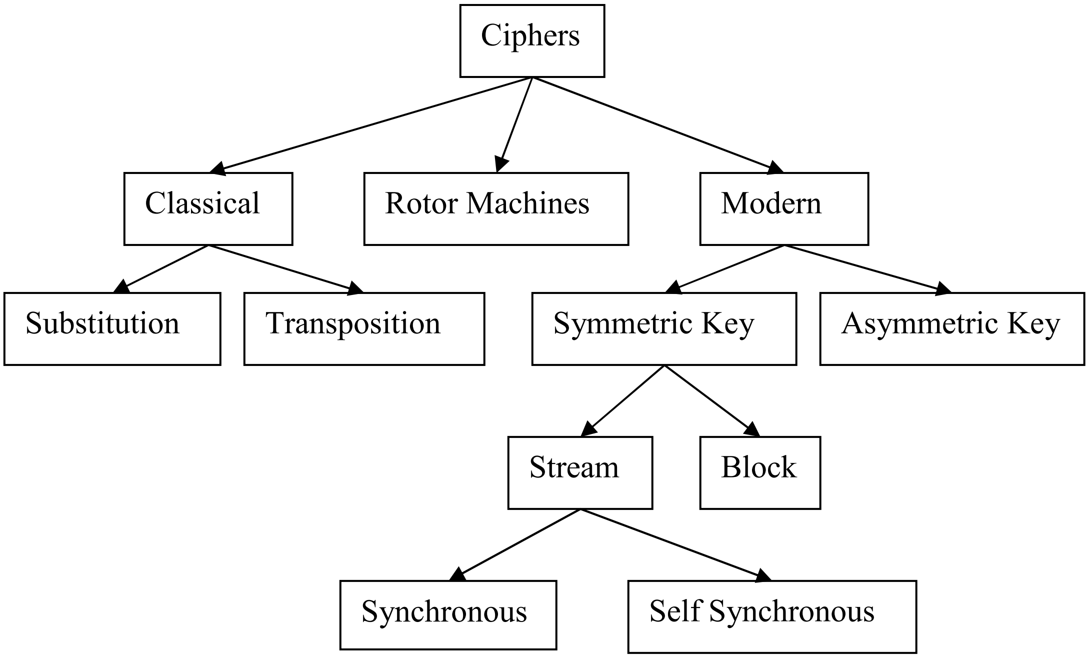


Figure 2.3: Types of ciphers [6]

### 2.4.1.1 Substitution cipher

In this cipher, plaintext letter or a set of plaintext letters are substituted with ciphertext by following a method which is regular. There are different types of substitution ciphers, based on whether they operate on single letter at a time which is called **simple substitution cipher** or on a set of letters at a time which is called **poly-graphic substitution cipher**. In **mono-alphabetic substitution cipher,** fixed substitutions are done through the entire message. In **poly-alphabetic substitution cipher,** different substitutions are done at different times in a message. In **homo-phonic substitution cipher,** plaintext letter is mapped to more number of ciphertext. In **one-time pad substitution**, plaintext letter is combined with the key character at that position using XOR. Decryption is done by performing reverse order of substitution. Substitution ciphers are not very strong and can be easily broken by having the knowledge of frequency analysis as classical ciphers [6].

### 2.4.1.2 Transposition cipher

In transposition cipher, the orders of the plaintext letters are changed. There are many different types of transposition cipher such as rail fence cipher, route cipher, columnar transposition, double transposition, myszkowski transposition, disrupted transposition.

If frequency distribution of ciphertext is nearly same as the frequency distribution of plaintext, then the cipher used is transposition. As transposition does not affect the frequency of letters because, the ciphertext contains all the letters of plaintext but in a different order. By moving letters of ciphertext and looking for anagrams of English words, transposition can be broken. If an anagram is found, information about the pattern used is revealed. This is called **anagramming**.

In order to increase the strength of transposition and substitution, which are weak when used individually, they can be combined for a stronger cipher. By doing so, the weakness of one method is overcome by the other. The pattern used can not be revealed by replacing high frequency ciphertext letters with high frequency plaintext letters as transposition is used. Anagramming of transposition fails as substitution is also used [6].

### 2.4.2 Rotor machines

Rotor machine is an electro-mechanical device. These machines were used widely in 1930-1950s. The main component of rotor machine is a group of wheels, which rotates. These wheels are equipped with an array of electrical contacts on both their sides. Substitution of letters is implemented by the wiring between the contacts Letters are scrambled in some assumed complex pattern. After encrypting each letter, the position of the rotor is advanced. Thus the substitution changes for every letter after encrypting it. A rotor machine produces a complex poly-alphabetic substitution cipher. Enigma machine is a famous example [6].

### 2.4.3 Modern cipher

Modern ciphers are classified based on their operation and whether they use one or two keys. This cipher is sub-divided into symmetric which uses one key and asymmetric which uses two keys for encryption and decryption.

### 2.4.3.1 Symmetric key

It is also called as single-key, private-key and one-key. A key is shared secretly between two parties which want to communicate. A single key is used for both encryption and decryption [5]. There are two types of symmetric key ciphers, stream ciphers and block ciphers. Popular examples of symmetric key ciphers are DES, AES, TDES. There are few disadvantages with this type of encryption:

- It needs a common key, which has to be kept secret during the distribution and also when it is being in use.
- Receiver cannot verify that the message was not altered and the message was sent by the authorized sender.
- If ‘n’ parties want to communicate, then n(n-1)/2 keys are needed by the system. Key management is difficult as the key must be kept secret.

### Stream cipher

They are also known as state ciphers. These ciphers encrypt one bit of plaintext at a time. Transformations of each bit or byte vary during encryption. Based on the key, a pseudo-random key stream is generated. An internal state is considered to generate successive elements of a key stream. Internal state is updated in two ways:

1) It is updated independent of plaintext and ciphertext.
2) It is updated based on the previous ciphertext.

The method used for updating the internal state sub-divides stream cipher into synchronous and self-synchronous stream cipher [6].

### Synchronous stream cipher

A pseudo-random digit is generated independent of the plaintext and ciphertext. These digits are combined with plaintext for encryption and are combined with ciphertext for decryption. Sender and receiver should be synchronized for correct decryption. During transmission, if any digits are added or deleted, the synchronization is lost. If a single digit is corrupted only that single digit in the plaintext is affected, the error does not propagate [6].

### Self-Synchronous stream cipher

It is also known as asynchronous stream ciphers. Key stream is computed using N previous ciphertext digits. The receiver synchronizes with the key stream generator after receiving N ciphertext digits. Advantage with this cipher is, it can be easily recovered even if there is addition or deletion of a digit from the message stream.

Stream ciphers are fast and easy to implement with less hardware complexity, but are less secure [6].

### Block cipher

This cipher encrypts fixed-length of bits as a single unit. There are many modes of operation. One of the modes of operation uses padding, which allows plaintext of any length to be encrypted. Block ciphers are slow with more hardware complexity, but are more secure compared to stream cipher. DES, AES and Triple DES are popular examples of block ciphers.

### Data Encryption Standard (DES) [11]

DES was selected as an official Federal Information Processing Standard (FIPS) for United States in 1976. As DES is a block cipher, its block size is 64 bits and key size is 64 bits, in which only 56 bits are used by the algorithm and the remaining 8 bits are used for checking parity which will be discarded after use. There are 16 identical stages which are called rounds.

At each round, 64 bit plaintext is divided into two 32 bit blocks and each block is processed alternatively, which is called as Feistel scheme shown in Figure 2.4. One 32 bit block is transformed with the key by using F-function (Feistel - function) described in Figure 2.5. This block is combined with the other 32 bit block using X-OR bit-by-bit. The results obtained at each stage are swapped for the next stage.

The F-function has four stages:

**Expansion:** Block with 32 bits is expanded to 48 bits by duplicating few bits.

**Key Mixing:** The expanded block is combined with the subkey by X-OR operation. Sixteen 48 bits subkeys, one for each stage are derived from the main key. 56 bits from the main key are taken and divided into two blocks of 28 bits each. Each block is shifted

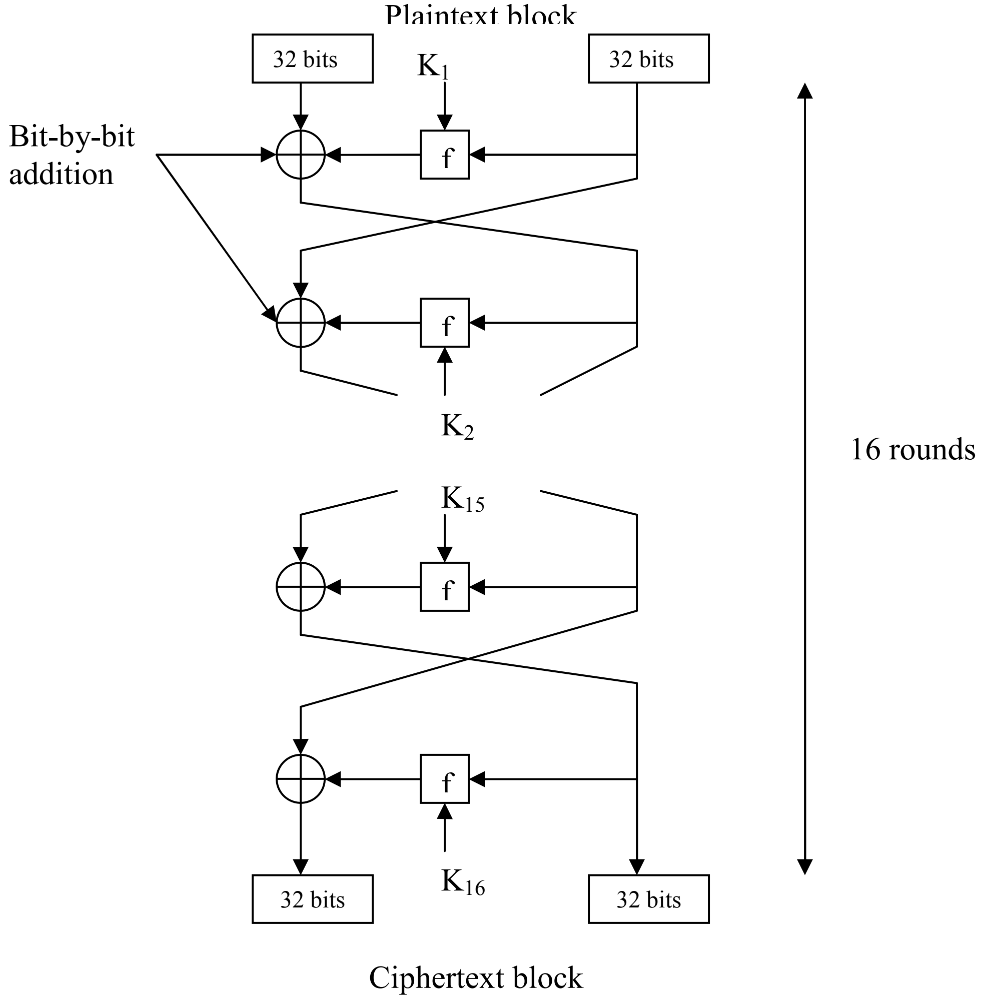


Figure 2.4: Feistel scheme [11]

to the left by one bit for each stage and 24 bits from each block are selected to make one 48 bit subkey, which is called key scheduling.

**Substitution (S-box):** Each block of 48 bits is divided into eight 6 bit blocks. These 6 input bits are replaced by 4 bits of output by a non-linear transformation. The output of this stage is 32 bits.

**Permutation (P-box):** 32 bits from the substitution stage are rearranged according to a permutation which is the same for all the 16 stages.

DES was less secured and was slow. Research was performed to design a block cipher which was more secure. DES was reused as Triple DES (TDES) which was relatively more secured but was very slow.

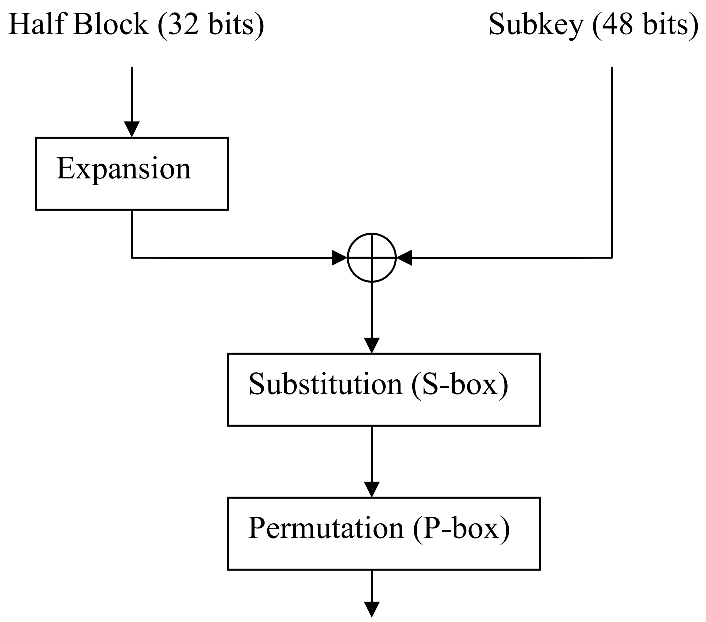


Figure 2.5: Feistel function (F-function) [6]

## **Triple Data Encryption Standard (TDES)** [6]

TDES is using DES three times. There are two types of TDES, one which uses two different keys 2TDES and three different keys 3TDES. (See Figure 2.6)

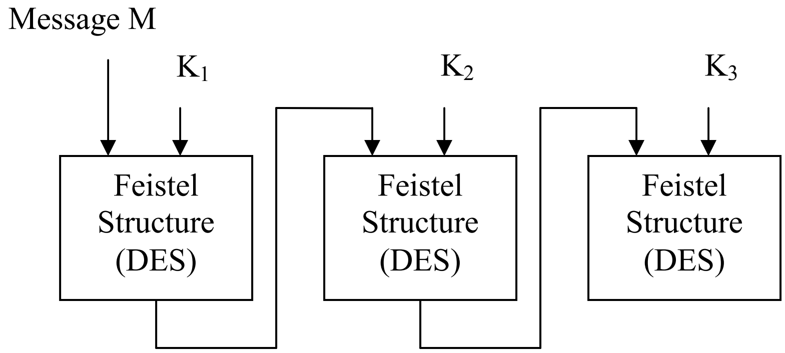


Figure 2.6: TDES structure [6]

**TDES operation:**

$$DES\ (K_3;\ DES\ (K_2;\ DES\ (K_1;\ M)))$$

where M is the message that has to be encrypted and $K_1$, $K_2$, $K_3$ are DES keys.

2TDES with two different keys where $K_1 = K_3$ has a key size of 128 bits, out of which only 112 bits are used and the others are used as parity bits. 3TDES has key size of 192 bits, out of which only 168 bits are used.

TDES is less used as it is being replaced by Advanced Encryption Standard (AES). TDES has slow performance in software but AES is 6 times faster when compared to DES.

### **Advanced Encryption Standard (AES)** [6]

AES is faster and easy to implement in hardware and software. It uses less memory. Its block size is 128 bits and key size of 128, 192 or 256 bits can be used.

AES operates on 4X4 array of bytes which is called a state. There are four stages.

**AddRoundKey:** Round key is obtained from the main key using key scheduling. This round key is combined with each byte of the state using X-OR.

**SubBytes:** Each byte is replaced by another byte following a non-linear substitution. This stage provides the required non-linearity in the ciphertext.

**ShiftRows:** It is a transposition step. This step shifts the bytes in each row by a varying offset in a cyclic order. The first row is not changed.

**MixColumns:** It is a mixing operation in which all four bytes of a column of a state are combined using a linear transformation.

ShiftRows and MixColumns provide the required diffusion in the ciphertext.

### 2.4.3.2 Asymmetric key

It is also called as public key encryption. It uses two keys which are related mathematically to each other but not derivable from each other, private key and public key [5]. Private key has to be kept secret and public key can be published. Key management is easy as it involves less number of keys when compared to symmetric key encryption which requires many keys if more number of parties want to communicate. There are two types of operation. Figure 2.7 and 2.8 present a block-diagram of the encryption/decryption process with public key.

1) Public key encryption in which a message encrypted with receiver’s public key can be decrypted only by the receiver with his/her private key. This ensures **confidentiality**.

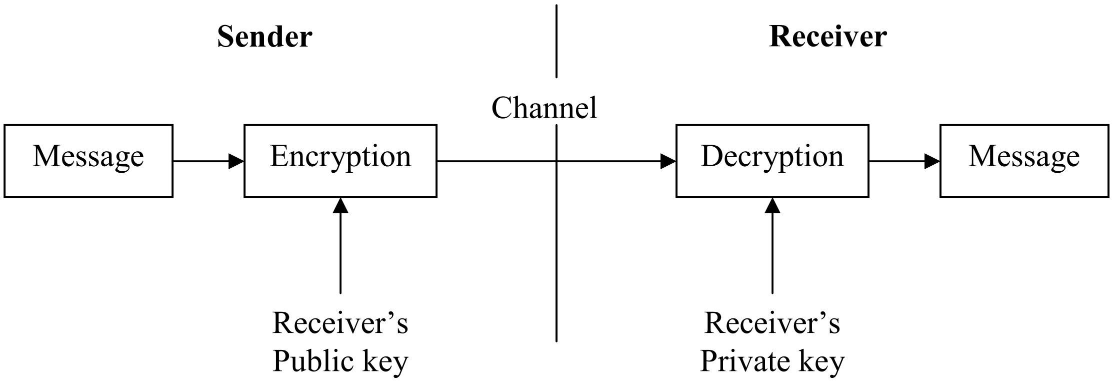


Figure 2.7: Public key encryption for confidentiality

2) Digital signatures in which a message encrypted with sender's private key can be decrypted by anyone who has sender's public key. This proves that the message was not tampered and was sent only by the sender, which ensures **authenticity**.

There are few disadvantages with this type of encryption [5]:

- It is costly and needs more computation when compared to symmetric key encryption.
- It is 100 to 1000 times slower than the symmetric key encryption. As it takes more time, it should not be used for encrypting bulk messages.

RSA is the most popularly used asymmetric key encryption algorithm till date.

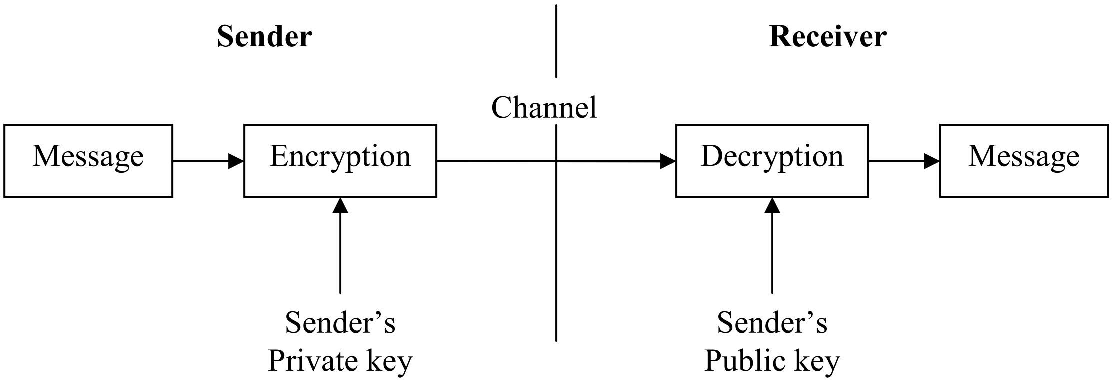


Figure 2.8: Public key encryption for authenticity

## **RSA (Rivest, Shamir and Adleman)** [11]

This algorithm is called RSA as it was developed by Ron Rivest, Adi Shamir and Leonard Adleman. This algorithm assumes that the plaintext P and ciphertext C are integers which are represented by the data in the plaintext and ciphertext blocks. It also assumes that P and C are in the interval [0, n], where ‘n’ is the product of two large prime numbers ‘a’ and ‘b’ which are selected randomly.

$$\phi(n) = (a-1)*(b-1),$$

Choosing ‘s’ such that it lies in the interval [1, Ø (n)] and has no factors with Ø (n). This is public key exponent. Then choosing ‘r’ such that

$$r*s = 1 + x*\phi(n),$$

for some integer ‘x’. This is private key exponent.

**Encryption**

Receiver sends his/her public key (n and s) to the sender and keeps private key as a secret. When the sender wants to send the plaintext P, he/she computes ciphertext using receiver’s public key:

$$C = P^{s} \bmod n,$$

where C is transmitted to the receiver.

**Decryption**

At the receiver’s end, P can be recovered from the ciphertext by using receiver’s private key.

$$P = C^{r} \bmod n$$

If ‘n’ is chosen to be very large, then factorizing will be computationally difficult.

Asymmetric key encryption is slow as it involves overhead for calculating the relation between the public key and private key. So it can not be used for bulky messages, but it is much secured compared to symmetric key encryption. Symmetric key encryption’s security lies in transmitting and handling the keys safely. It is fast and can be used for bulky messages. To overcome the disadvantages of the individual encryption methods, they are combined together. Asymmetric key encryption is used for transmitting the symmetric keys securely between the parties. This is done by encrypting the secret key using the public key of the receiver, which can be decrypted only by the receiver using his/her private key. After obtaining the secret key, symmetric encryption is carried out between the two parties. This way, the advantages of both the methods are combined and disadvantage of one method is overcome by the other. This can be used for applications which require high security.

## 2.5 Evaluation

In this section, encryption technique applied to biometric data is explained.

### 2.5.1 Securing Data and Financial Transaction [12]

The main problem of security arises when the transactions are carried on a remote system. As most of the data transfers and financial transactions are performed electronically, a highly secure system is necessary. An unauthorized access can cause great damage.

Traditional security schemes such as ‘what they have?’ (ID card, tokens) and ‘what they know?’ (Password, PIN)), can not guarantee for a positive identification in a remote system. PINs, passwords, IDs, encryption key can be stolen, lost or can be exchanged between the users. A hacker can break any encryption technique by stealing or discovering the private key of an authorized user and can decrypt the sensitive data easily.

Biometrics can be used for positive identification along with encryption techniques for secure communication. The user places his/her finger on the sensor, which is a small optical device that produces a fingerprint image in a digital form. An algorithm is applied to convert the digital image into useful data for identification, which is stored as template in the database. Templates are encrypted when they have to be transmitted between the database and the matching module for matching purpose.

There are two types of architectures for verification of a user who intends to get an access to a device or database which is kept secured. They are central and distributed architectures. The use of these architectures depends on the application.

**Central Architecture**

The user places his/her finger on the sensor, the output of the sensor is the digital image of the user’s fingerprint. Features are extracted and sent to the matcher. The stored template at the time of enrollment is sent to the matcher. Live template and stored template are matched. Figure 2.9 illustrates how recognition system is performed.

If the user’s identity is confirmed, authorization or access is granted. Other information of that particular user such as credit card information or medical records may be transferred if needed. This architecture is used when other information needs to be stored along with the template. The hacker can intercept the data between the feature extractor and the matcher or matcher and the database, if the matcher and database are remote.

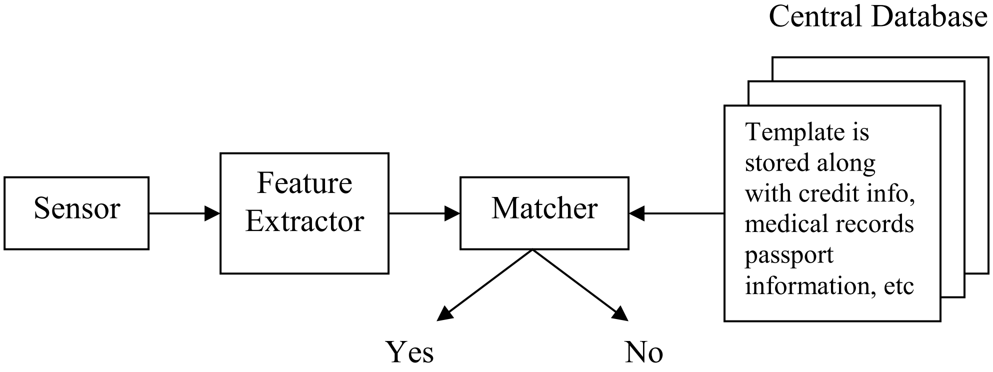

Central Database
Sensor
Feature Extractor
Matcher
Template is stored along with credit info, medical records passport information, etc
Yes
No

Figure 2.9: Central architecture system

**Distributed Architecture**

In this architecture, the template is stored on a medium at the time of enrollment, which is carried by the user. This system is used when there are many users. The user places his/her finger on the sensor and also scans the card through a scanner. Features are extracted from the digital fingerprint and are sent to the matcher.

The stored template is obtained when the card is scanned and is sent to the matcher. Live template and stored template are matched. Authorization or access is granted if the user is the true owner of the card. Information such as social security number, passport information, credit information, medical records can be stored on the card. Figure 2.10 illustrates how recognition system is performed in distributed architecture.

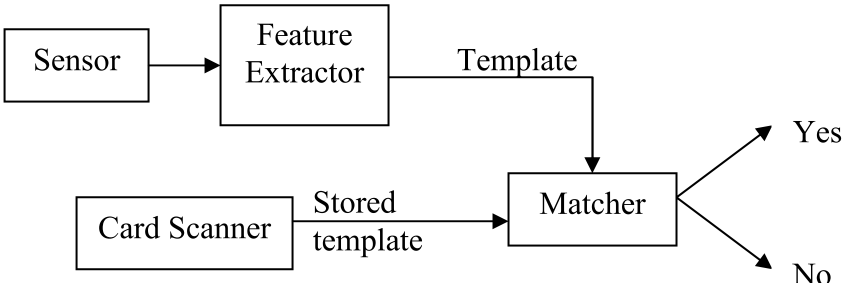


Figure 2.10: Distributed architecture system

Encryption is used when distributed architecture is implemented in the system. Central architecture needs encryption techniques when the **verification terminal is**

**remote or unsupervised**. By using encryption for communication purpose, it prevents hackers from emulating the existence of a device from another location. It provides the integrity but not the authenticity of the user as the hacker can get the access of the private key used for encryption. So fingerprint verification and the key are required for integrity and authenticity. Fingerprint verification provides positive identification of the holder of the key and the key provides authenticity of the biometric verification terminal.

**Database Security**

New Secure Service Option by Oracle uses central database architecture for their Oracle 7TM database and Identix TouchSafe IITM is used as encrypted verification terminal. (See Figure 2.11).

The verification terminal consists of fingerprint reader and a compatible printed circuit board which is installed in the client's PC. There is a connection cable between the sensor and the client's PC. Encryption key is stored in non-volatile memory on the printed circuit board. There is a database which is accessed only by authorized users. For secured communication between the database and the client's PC, transmitted data is encrypted. At the enrollment stage, the user's logon name is stored along with the template of that user.

At the time of verification, the user logs on and places the finger on the verification terminal. The user's stored template in the database is sent to the verification terminal where both are matched. If they match, the user is given access, else the user is rejected.

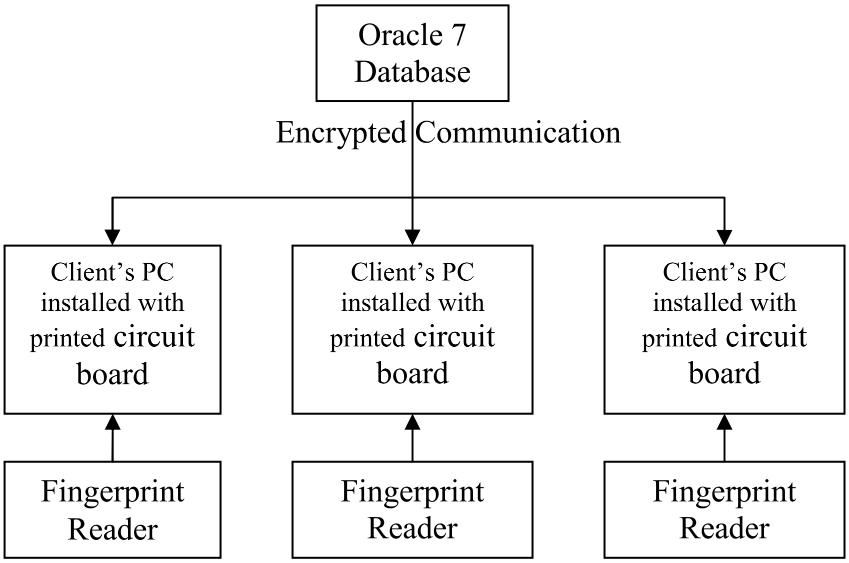

Oracle 7 Database
Encrypted Communication
Client's PC installed with printed circuit board
Client's PC installed with printed circuit board
Client's PC installed with printed circuit board
Fingerprint Reader
Fingerprint Reader
Fingerprint Reader

Figure 2.11: New Secure Service Option system [12]

This scheme gives access only to authorized users.

**Financial Transactions**

Using biometrics can prevent fraud in the bank card industry. Biometrically encoded debit or credit cards can be used to eliminate this fraud. If this card was stolen or lost, it can not be used by another person. And also the imposter who is trying to use the lost or stolen card would be caught at the first attempt.

## 2.5.2 Biometric Authentication for E-Commerce Transaction [13]

E-Commerce industry is growing enormously because of ease of use and its efficiency. But it has few security issues which need to be addressed to prevent card industry fraud and invasions of privacy data. In order to achieve this, biometric attributes have to be used for identification and verification for protecting customers from fraudulent attacks. **A web-based architecture which uses encrypted iris patterns for authentication of a customer for e-commerce transactions has been developed**. Iris is used for authentication, as it is unique and does not change with time. Iris image is acquired, processed and features are extracted. These features are encrypted and stored.

**Iris Authentication System**

The biometric card authentication system consists of iris image feature extractor and RSA encryption unit at the client side and a verification unit along with the database consisting of credit/debit card details at the server side for authentication.

At the enrollment stage, the encrypted iris of the user is stored along with the credit/debit card number in the issuing company's database. Client's PC is equipped with web camera to capture the image of iris at the time of transaction. The client's PC has software which preprocesses, normalizes, enhances and extracts the features of the captured iris image, using Principal Component Analysis (PCA). PCA is used for finding patterns in the data. By using those patterns, the image is compressed without losing much information. Iris image whose size is m x n pixels after enhancement stage can be converted to a matrix of m x n size. Eigen vector and eigen values for this matrix are calculated and eigen vectors whose eigen values are large are considered. Eigen vectors whose eigen values are small are rejected as they have less information. This data set is encrypted by using RSA algorithm and is transmitted to the e-commerce site along with credit/debit card details by the client’s system. Figure 2.12 displays a web-based architecture involving biometric-based authentication.

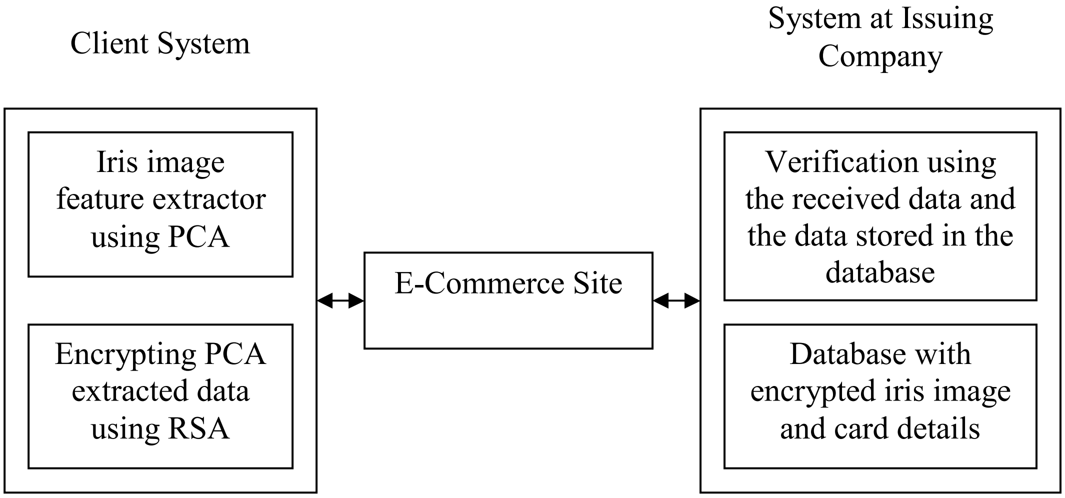

Client System
System at Issuing Company
Iris image feature extractor using PCA
Encrypting PCA extracted data using RSA
E-Commerce Site
Verification using the received data and the data stored in the database
Database with encrypted iris image and card details

Figure 2.12: Web-based architecture for biometric authentication system [13]

E-Commerce site will transfer these details to the issuing company for verification. The software at the issuing company verifies the identity of the client with the details stored in its database. It sends its confirmation to the e-commerce site to continue the transaction. As iris image is encrypted before transmitting it to the e-commerce site, the biometrics of the user is protected and only the issuing company can access it.

### 2.5.3 Cancelable Biometric Filter for Face Recognition [14]

Cancelable biometric template is an important aspect for improving biometric authentication system. Consider a biometric template stored on a card for authentication purpose. When the card is lost or stolen, then the biometrics is lost for the rest of the life. In order to ensure cancelability and protect the biometrics from hackers, the biometric templates must be encrypted. Even if the biometric template is lost or stolen, a different encrypted biometric template can be generated from the original biometrics.

**System Architecture**

At the enrollment stage, few images of the user are taken. These images are convolved using a random convolution kernel. The user selects a PIN which is used as a seed to random number generator. Random number generator is used to generate random convolution kernel. All the convolved images of the user are used to produce a single biometric filter which is stored on the card. Figure 2.13 shows the enrollment stage.

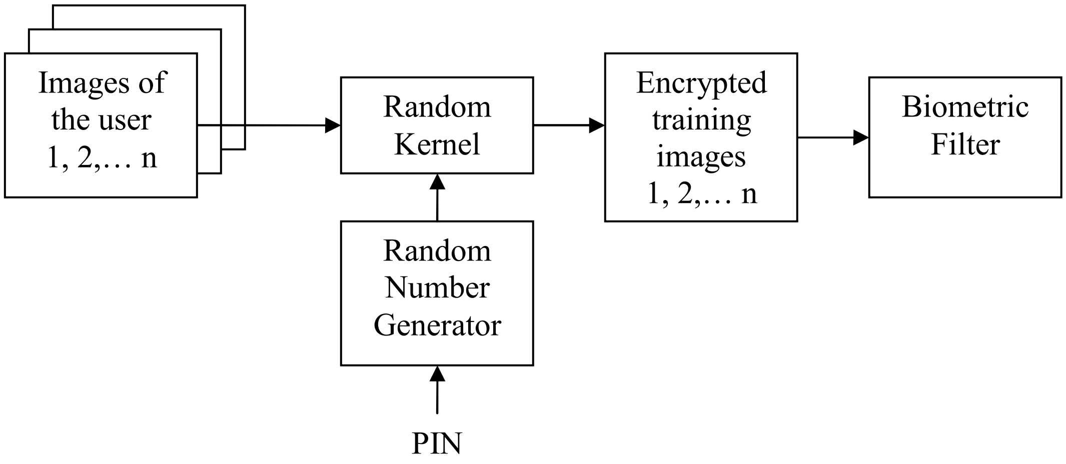

Images of the user 1, 2,… n
Random Kernel
Encrypted training images 1, 2,… n
Biometric Filter
Random Number Generator
PIN

Figure 2.13: Enrollment stage [14]

Inverse Fourier transform of the encrypted biometric filter will not produce the any images which can look like a face as the images are convolved using random kernels.

If the card is lost or stolen, the enrollment stage produces different random convolution kernels using a different PIN. Thus it produces different encrypted biometric filter which is re-issued on a different card. Attacker who stole the card can reconstruct the original filter by applying de-convolution, which is extremely difficult without the knowledge of the user's PIN or the random convolution kernel used.

At the verification stage, the user will present his/her card and also the PIN. By using the PIN as a seed for random number generator, it will produce the random convolution kernel. This random convolution kernel is used to convolve with the test face images presented by the user. The convolved test image is cross-correlated with the stored encrypted biometric filter on the card. The correlation outputs are used to determine whether the user is genuine or imposter. The metric used to determine this is peak-to-side lobe ratio (PSR). PSR is (peak – mean)/ standard deviation. The main advantage of this method is that the whole verification process is carried out in encrypted domain. Figure 2.14 shows the verification stage.

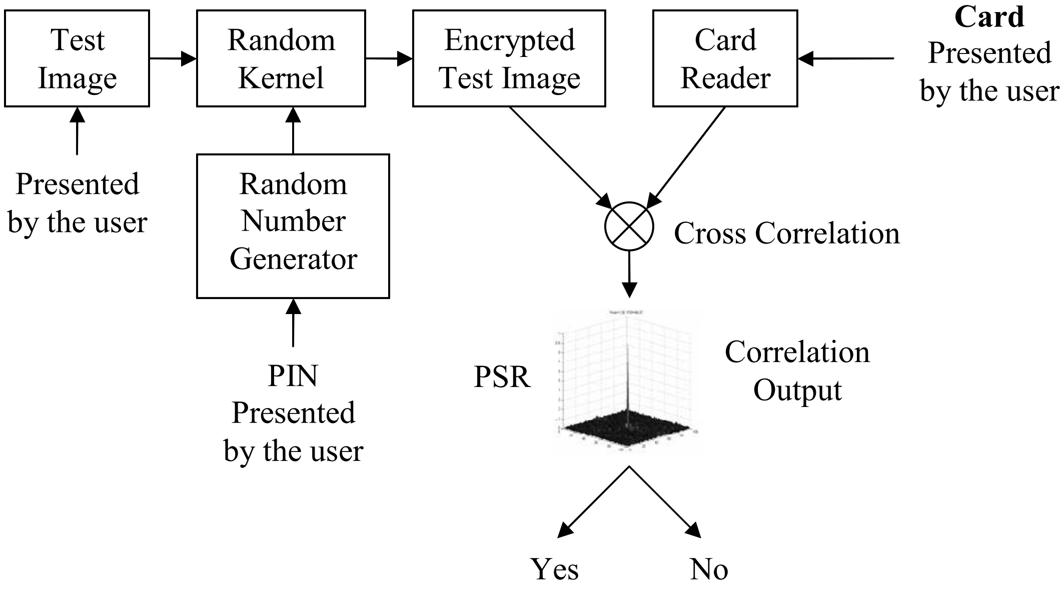


Figure 2.14: Verification stage [14]

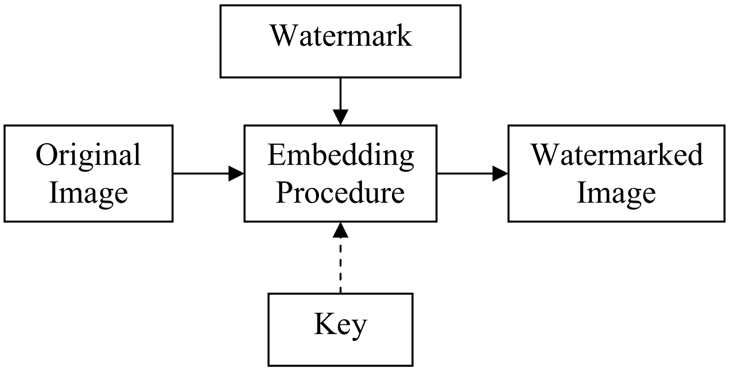
Using correlation filter provides shift invariance. The biometric authentication performance produces same results with or without encryption and also when the images are convolved with different random kernels. This is an important aspect in providing cancelable biometric templates as it gives us flexibility in choosing different random kernels for convolution when re-issuing the card with different biometric template from the same biometrics. Using different encryption technique can prevent the shift invariance property of correlation filters.

# Chapter 3

# Digital Watermarking

## 3.1 What is digital watermarking?

The process of embedding information into another object or signal is called watermarking [15]. A digital watermarking is a technique for permanently embedding an identification code into digital data such as audio, video or images. Identification code contains information related to copyright protection and data authentication [16]. If the owner of a digital file wants to protect the copyrights of his/her file, they can do so by using digital watermarking techniques. Figure 3.1 shows the watermark embedding process.

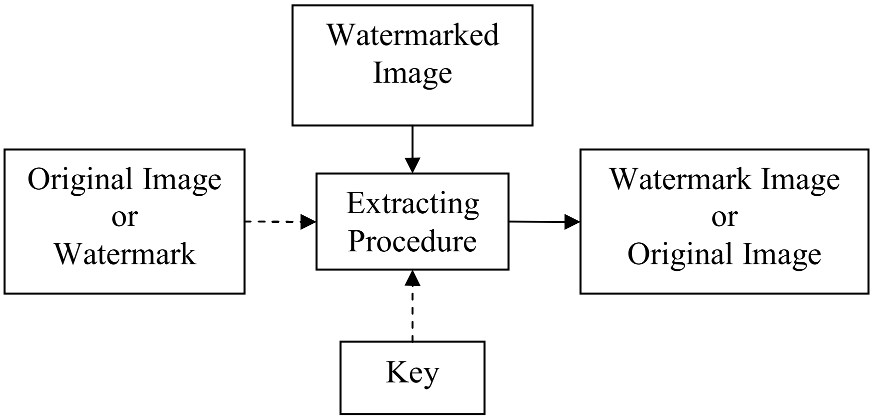


Dashed line is used to indicate that it is optional

Figure 3.1: Watermark embedding procedure [17]

Information about the original image such as author, creator, owner, distributor or authorized consumer is called watermark which is embedded into the original image by the embedding procedure. Using a key is optional, it is used to increase the security of the system. Key is used as a seed for a pseudo random number generator. This random number generator produces positions where the watermark can be embedded. Even if the watermarked image is altered, the positions where watermark is embedded can be known [21]. The output is a watermarked image.

In the extracting procedure, the watermarked image is used to extract the watermark or the original image. In the embedding procedure, if a key is used, then the same key is required at the extracting stage. At this stage, either the watermark or the original image can be provided to the system. The watermark extracting process is shown in Figure 3.2.


Watermarked Image
Original Image or Watermark
Extracting Procedure
Watermark Image or Original Image
Key

Dashed line is used to indicate that it is optional

Figure 3.2: Watermark extracting procedure [21]

If the original image is provided, then the watermark can be recovered or if a watermark is presented, then the original image can be obtained [21]. Few extracting procedures extract the watermark without using the original image or the watermark embedded.

If the watermarked image is copied and distributed, the watermark is also distributed along with the image which can be detected.

## 3.2 Purpose of watermarking (or) Why watermarking?

With the rapid growth of networked multimedia systems, the digital files are subjected to illegal copying, forgery and unauthorized distribution [18]. To overcome the above problems and to determine where and by how much a digital file has been changed, can be achieved by using digital watermarking techniques. These techniques can be used for protecting the copyrights of the owner of the digital file [17].

## 3.3 Applications

Digital watermarking is used for copyright protection, data integrity and data authentication of digital files. For copyright protection, the information about the owner is embedded as a watermark to the digital file. This prevents others from claiming falsely about the ownership of the file. By embedding the watermark, it also prevents from illegal copying of the digital file, which ensures data integrity. For data authentication, where and by how much a file has been changed can be detected by embedding a watermark into the digital file [19].

## 3.4 Types of watermarks

Ideal watermarking system has the following properties [22]:

**Perceptibility:** A digital watermark which is embedded must be perceptually invisible so as to prevent distracting or distorting the original or cover image [20].

**Statistically invisible:** The imposter should not be able to detect the embedded watermark, so that he/she can erase or destroy it [18].

**Embedding and Extracting:** Embedding procedure should be simple and computationally less complex, so that the extracting process has less computation. If the embedding and extracting procedures are computationally less, it involves less cost and it can be used widely. Extraction should be accurate [22].

**Robustness:** The watermark should be resistant to signal processing attacks such as rotation, filtering, adding noise, compression, etc [22]. Accuracy in extracting and robustness are achieved by embedding the watermark bits several times at different locations in the original image [23].

**Unambiguous:** Identification of the true owner should be accurate based on the retrieval of the watermark [22].

**Capacity and Speed:** An ideal watermarking system should be able to embed any amount of information ranging from 1 bit to a whole image [20]. The procedure for detecting or embedding the watermark should be less complex so that it can be used in biometric systems [22].

Watermarks are sub-divided into many types based on different aspects.

## 3.4.1 Division Based on human perception

This is sub-divided into visible watermarks and invisible watermarks.

### 3.4.1.1 Visible watermarks

These watermarks can be seen clearly by the viewer and can also identify the logo or the owner. Visible watermarking technique changes the original signal. The watermarked signal is different from the original signal [6]. (See Figure 3.3)

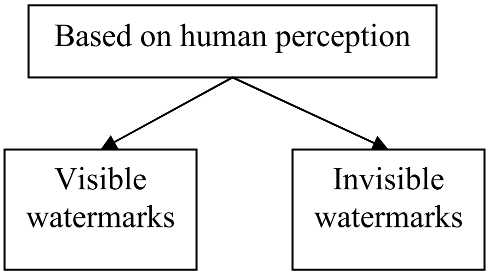


Figure 3.3: Classification of watermarks – Type I

Visible watermark embedding algorithms are less computationally complex. The watermarked image can not with stand the signal processing attacks, like the watermark can be cropped from the watermarked image [24]. Figure 3.4 shows visible watermarked image.

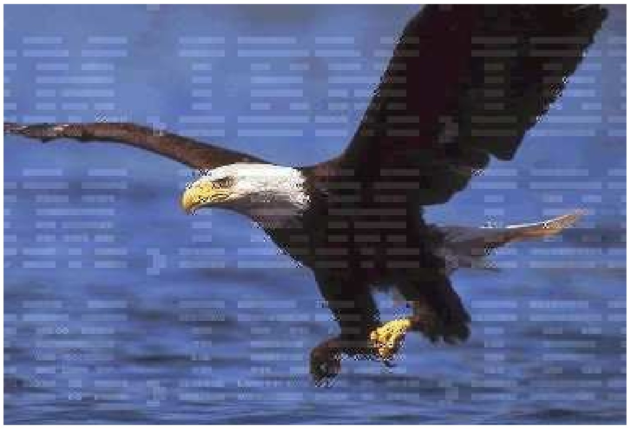


Figure 3.4: Visible watermarked image [21]

Spreading the watermark throughout the image is a best option, but the quality of the image is degraded which prevents the image from being used in medical applications [24].

### 3.4.1.2 Invisible watermarks

These watermarks can not be seen by the viewer. The output signal does not change much when compared to the original signal. Figure 3.5 shows invisible watermarked image.

Figure 3.5: Invisible watermarked image [21]

The watermarked signal is almost similar to the original signal [6]. As the watermark is invisible, the imposter can not crop the watermark as in visible watermarking. Invisible watermarking is more robust to signal processing attacks when compared to visible watermarking. As the quality of the image does not suffer much, it can be used in almost all the applications [24].

## 3.4.2 Division Based on applications

Based on application watermarks are sub-divided into fragile, semi-fragile and robust watermarks. (See Figure 3.6).

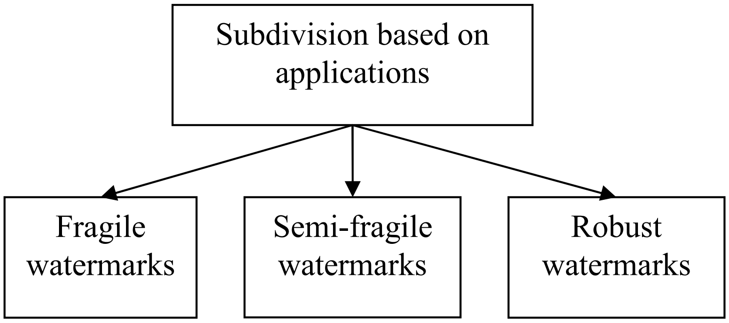
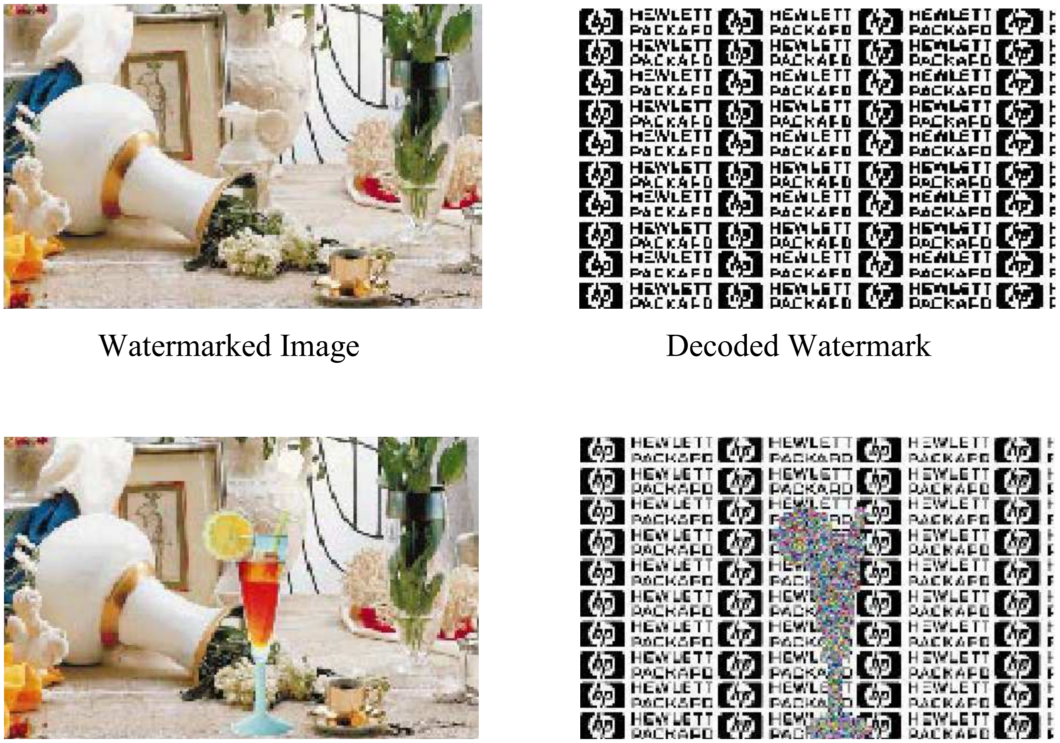


Figure 3.6: Classification of watermarks – Type II

### 3.4.2.1 Fragile watermarks

These watermarks are very sensitive. They can be destroyed easily with slight modifications in the watermarked signal which is shown in Figure 3.7.



Watermarked Image

Decoded Watermark

Watermarked Image with a change

Decoded Watermark

Figure 3.7: Fragile watermarked Images [21]

If there is any slight change to the watermarked signal, the fragile watermark is broken. This method ensures the data authentication, as the signal whose watermark is broken does not provide authentication [6].

### 3.4.2.2 Semi-fragile watermarks

These watermarks are broken if the modifications to the watermarked signal exceed a pre-defined user threshold. If the threshold is set to zero, then it operates as a fragile watermark. This method can be used to ensure data integrity and also data authentication [19].

### 3.4.2.3 Robust watermarks

These watermarks can not be broken easily as they withstand many signal processing attacks. Robust watermark should remain intact permanently in the embedded signal such that attempts to remove or destroy the robust watermark will degrade or even may destroy the quality of the image. This method can be used to ensure copyright protection of the signal [19].

## 3.4.3 Based on domains

Watermark is embedded into the original image in spatial-domain or in transform-domain. Figure 3.8 shows the classification of watermarks based on the domains used for watermarking.

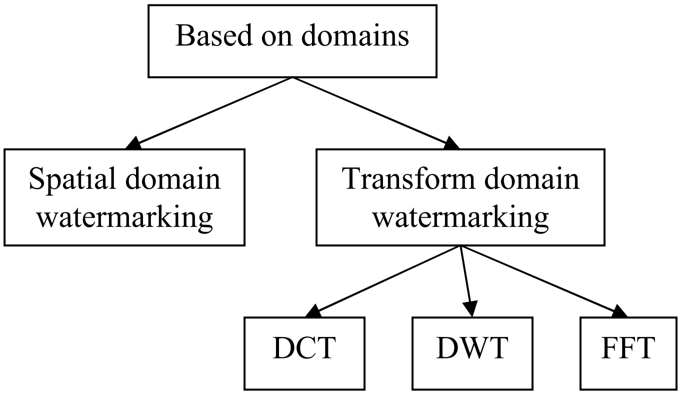


Figure 3.8: Classification of watermarks – Type III

### 3.4.3.1 Spatial domain watermarks

The watermark is embedded directly into pixels of the image, which involves less complex computation. Spatial domain watermarks are not robust to some signal processing attacks [18].

### 3.4.3.2 Transform domain watermarks

Also called as spectral or frequency domain watermarks. Watermark is embedded into the transform coefficients of the image. The transforms include Discrete Cosine transform, Discrete Wavelet transform and Fast Fourier transform applied to the image [18]. For example, for DCT watermarking, the image is transformed using discrete cosine transform and the watermark is embedded into these transformed coefficients of the image. Transform domain watermarks are more robust to signal processing attacks, for example, to compression, zooming, etc when compared to spatial domain watermarking [24].

## 3.4.4 Division Based on the level of information required to detect the embedded data

Based on the level of required information all watermarks are sub-divided into blind watermarks, semi-blind watermarks and non-blind watermarks. Figure 3.9 shows the classification of watermarks based on the level of information required to detect the embedded data.

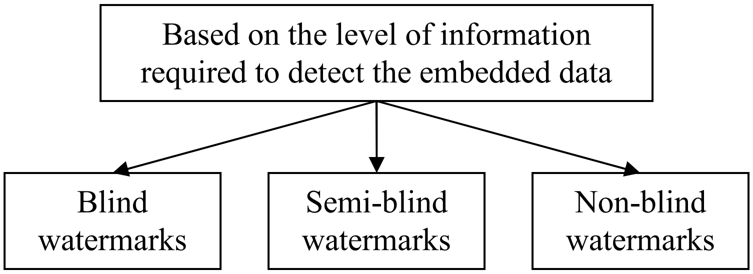

Based on the level of information required to detect the embedded data
Blind watermarks
Semi-blind watermarks
Non-blind watermarks

Figure 3.9: Classification of watermarks – Type IV

#### 3.4.4.1 Blind watermarks

These watermarks detect the embedded information without the use of original signal. They are less robust to any attacks on the signal [18].

#### 3.4.4.2 Semi-blind watermarks

These watermarks require some special information to detect the embedded data in the watermarked signal [18].

#### 3.4.4.2 Non-blind watermarks

These watermarks require the original signal to detect the embedded information in the watermarked signal. They are more robust to any attacks on the signal when compared to blind watermarks [18].

### 3.4.5 Based on user's authorization to detect the watermark

This is sub-divided into public watermarks and private watermarks. Figure 3.10 shows the classification of watermarks based on user's authorization to detect the watermark.

#### 3.4.5.1 Public watermarks

In this watermarking, the user is authorized to detect the watermark embedded in the original signal [18].

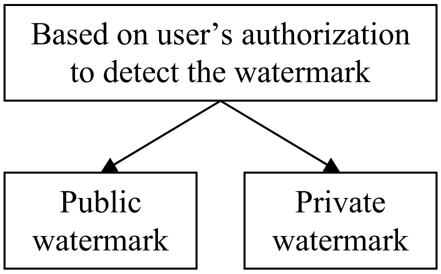


Figure 3.10: Classification of watermarks – Type V

### 3.4.5.2 Private watermarks

In this watermarking, the user is not authorized to detect the watermark embedded in the original signal [18].

### 3.4.6 Based on the keys used for watermarking

This is sub-divided into symmetric watermarking and asymmetric watermarking. Figure 3.11 shows the classification of watermarks based on the keys used.

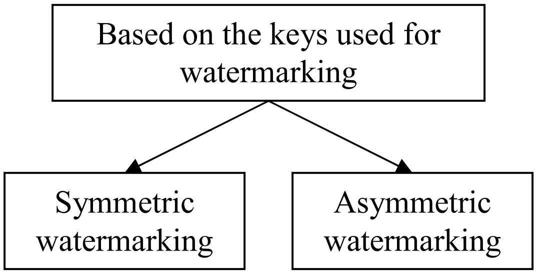


Figure 3.11: Classification of watermarks – Type VI

### 3.4.6.1 Symmetric watermarking

In this watermarking, same key is used for embedding and detecting watermarks, which is similar to symmetric key encryption in which same key is used for encryption and decryption [18].

### 3.4.6.2 Asymmetric watermarking

In this watermarking, different keys are used for embedding and detecting watermarks [18].

### 3.4.7 Division Based on knowledge of the user on the presence of the watermark

This is sub-divided into steganographic watermarking and non-steganographic watermarking [18]. (See Figure 3.12).

#### 3.4.7.1 Steganographic watermarking

The user is not aware of the presence of the watermark [18].

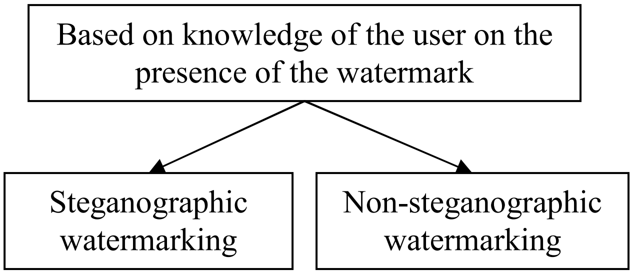

Based on knowledge of the user on the presence of the watermark
Steganographic watermarking
Non-steganographic watermarking

Figure 3.12: Classification of watermarks – Type VII

#### 3.4.7.2 Non-steganographic watermarking

The user is aware of the presence of the watermark [18].

### 3.4.8 Division Based on the quality of the original signal

By embedding a watermark into the original signal, the quality of the signal is degraded. The acceptable level of degradation is determined by the pre-specified user threshold.

## 3.5 Evaluation

In this section, a number of watermarking techniques which belong to spatial and frequency domain will be explained in detail.

### 3.5.1 A Spatial Method for Watermarking of Fingerprint Images [25]

Watermarking is a solution for protecting Intellectual Property Rights (IPR) problem. Watermarking of fingerprint images provides protection to the original images which are stored in the database, against any attacks. Fraud or tampering can be detected in the fingerprint image by using fragile watermarks which do not with stand any changes and the watermarked fingerprint image can be transmitted through an unsecured channel and upon reception, changes can be detected, if any made.

Watermarking a fingerprint image by embedding a watermark should not corrupt the features or minutiae points which will in turn affect the verification stage. Destruction of the feature points can be prevented by using one of the two methods. In the first method, the features are extracted before watermarking the image, so that the regions in which feature points are present, can be avoided for embedding watermark. And in the second method, the fingerprint image is first watermarked and then features are extracted. The watermark bits are embedded in such a way that the original image gradient orientations are not changed. As the extraction of features depends on the gradient orientations of the image, embedding the watermark will not affect the feature extraction step.

**Method I**

**Embedding**

In this method, watermark is embedded after extracting the features, so that the regions used for fingerprint classification can be avoided for embedding. (See Figure 3.13). This method utilizes image adaptive strength adjustment technique, so that the visibility of the embedded watermark is low.

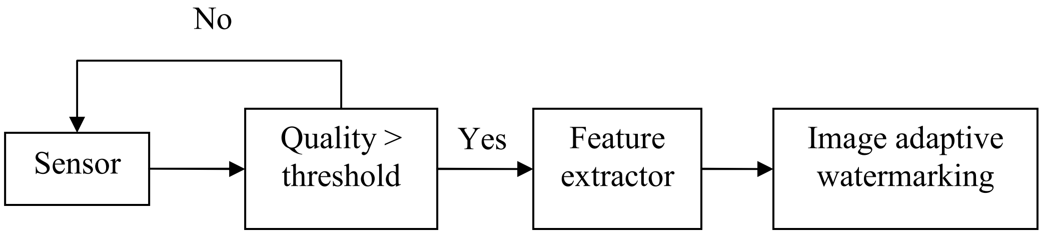


Figure 3.13: Watermarking embedding after feature extraction [25].

The acquired fingerprint image, after extracting the features is converted to gray scale image. A key is used as a seed for random number generator, which provides the locations for embedding the watermark. This key is kept as a secret. Watermark data is embedded into the gray-scaled fingerprint image using the formula

$$P_{WM}(i,j) = P(i,j) + (2s-1) * P(i,j) * q * \left[1 + \frac{SD(i,j)}{A}\right]\left[1 + \frac{GM(i,j)}{B}\right] * \beta(i,j),$$

where **(i, j)** is the location where the watermark is embedded which is determined by the secret key,

**$P_{WM}$ (i, j)** and **P (i, j)** are the pixel values of the watermarked and original image at the embedding location respectively,

**s** is the value of the watermark bit,

**q** is the strength of the embedding watermark,

**SD (i, j)** is the standard deviation of pixel values in a cross-shaped neighborhood of the embedding location (i, j),

**GM (i, j)** is the gradient magnitude at (i, j) which is calculated using the Sobel operator,

**A** and **B** are the normalization constants for standard deviation and gradient magnitude respectively,

**β (i, j)** is '0', if pixel (i, j) belongs to fingerprint feature region and is '1' otherwise.

The strength of the watermarking is adjusted by the standard deviation and the gradient magnitude. If a region in an image has high variance (SD (i, j) is high) or if it is an edge region (GM (i, j) is high), the watermark bit is added strongly. If the watermark bit is strong in the original or host image, decoding the watermark bit will be accurate. The human visual system is less sensitive to change in pixel values in busy region and edge regions, so the visibility of the watermark is decreased by embedding watermark in such areas. Each watermark bit 's' is embedded multiple times in a fingerprint image. Two reference bits '0' and '1' are embedded to the original or watermarked image along with the watermark data. These bits are used to determine the watermark bits at the decoding end by using an adaptive threshold.

**Extracting**

Locations of the embedded watermark bits are found by using the secret key used for embedding procedure. For each bit location, the original pixel value **Ê (i, j)** is estimated from the pixels in the cross-shaped neighborhood in watermarked ($P_{WM}$) image using the formula

$$\hat{E}(i,j)=\frac{1}{4c}*\left\{\sum_{k=-c}^{k=c}P_{WM}(i+k,j)+\sum_{k=-c}^{k=c}P_{WM}(i,j+k)-2*P_{WM}(i,j)\right\},$$

where **c** is the neighborhood size.

The difference between the estimated and current value is calculated

$$\delta = P_{WM}(i,j)-\hat{E}(i,j)$$

The average of the differences of all the embedding locations associated with the same bit is calculated $\boldsymbol{\delta^{avg}}$. These same averages are calculated for the reference bits '0' and '1', which are added at the embedding stage, $\boldsymbol{\delta^{avg}{}_{R0}}$ and $\boldsymbol{\delta^{avg}{}_{R1}}$. The watermark bit value is estimated by the following rule

$$\hat{S}=\left\{\begin{array}{l}1, if\delta^{avg} > \frac{\left(\delta^{avg}{}_{R0}+\delta^{avg}{}_{R1}\right)}{2}\\ 0, otherwise\end{array}\right\}$$

After extracting the features or the singular points, watermark is embedded. β (i, j) is zero for the feature regions, watermarking will not change the original image and the singular points are preserved.

**Method II**

In this method, features are extracted after the image is watermarked. Watermarking the fingerprint image is done in such a way that it will not change the gradient orientation around the watermark embedding location. This method utilizes feature adaptive watermarking technique. Figure 3.14 shows the block diagram of the steps followed in this method.

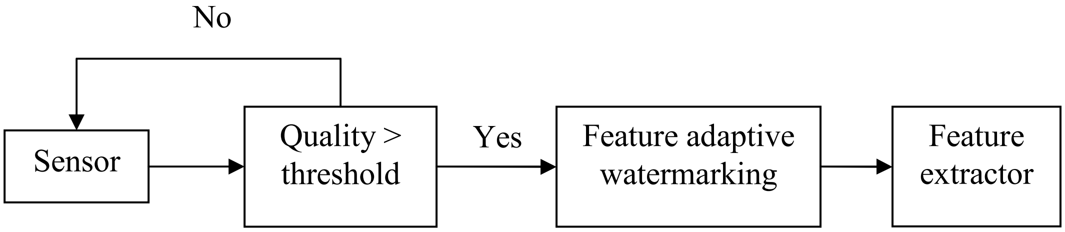


Figure 3.14: Extracting features after watermarking [25].

Features are extracted based on the gradient orientation. As watermark embedding does not change the quantized gradient orientations at the feature pixel and its neighbors, the singular points of the fingerprint image are preserved.

The gradient orientation circle around the pixel (i, j) is divided into 16 bands, each band is **π/8** radians. Let D (k, l) be the eight neighboring pixels of (i, j). Sobel operator is used to calculate gradients at all the pixels in D (k, l). For pixels at (i, j), its gradient is represented as α (i, j) and it is obtained by

$$\alpha(i,j) = \arctan\left[\frac{G_Y(i,j)}{G_X(i,j)}\right],$$

where **$G_X$ (i, j)** and **$G_Y$ (i, j)** are gradients in x and y directions respectively.

While embedding, the below equation is followed for all the pixels in D (k, l), so that the new gradient of the watermark bit preserves the features.

$$\left(\alpha_q - \frac{\pi}{16}\right) < \arctan\left[\frac{G_{yn}(i-1,j-1)}{G_{xn}(i-1,j-1)}\right] < \left(\alpha_q + \frac{\pi}{16}\right),$$

where **$\alpha_q$** is the gradient direction of the particular pixel,
**Gyn (i-1, j-1)** and **Gxn (i-1, j-1)** are the new values of vertical and horizontal gradients for pixel (i-1, j-1).

Embedding the watermark pixel does change the gradient orientation, but limits its change to the above interval, i.e. the gradient at the embedded pixel can change between **(-π/8, π/8)** radians.

## 3.5.2 Verification Watermarks on Fingerprint Recognition and Retrieval [20]

In a fingerprint system, the raw image is not used directly for verification purpose, instead minutiae points are used. Watermarking algorithms introduce small changes in the minutiae orientation, but they do not affect the performance of the fingerprint system as the matching system is tolerant to such small changes.

**Fragile Invisible Watermarking Technique**

This technique detects the changes or tampering from the extracted watermark.

**1) Using Look-Up-Table (LUT)**

In the embedding process, a watermark image is embedded into the source image by processing each pixel in the source image. Watermark extraction function is applied to a selected pixel in source image. The extracted bit is compared with the embedded watermark. If they are equal, the same process repeats for the next pixel. If they are not equal, the pixel value is incremented or decremented randomly. This is repeated multiple times till the embedded and the extracted watermark pixel values match. It is an iterative process. The value by which the extracted pixel is changed is calculated and propagated to other pixels which are not processed. This is called modified error diffusion procedure. The output of this stage is the watermarked image and a verification key.

In the extracting stage, the verification key produced at the embedding stage is used as a seed for a pseudo random number generator. Each binary entry in the look-up-table (LUT) is produced by recursively taking an output of a pseudo random number generator. The watermark extraction function is a function computed based on the verification key and the function can be a binary look-up-tables for a gray scale image or a set of binary look-up-table for a colored image. Watermark image, **b (i, j)** is extracted from the watermarked image **I'(i, j)** by applying the watermark extraction function from the verification key. If **b (i, j)** is the watermark image, then for gray-scaled image

$$b(i,j) = LUT(I'(i,j)),$$

where **I'(i, j)** is the watermarked image and LUT is the watermark extraction function.

This function is applied to all the pixels in **I'(i, j)**, which produces the extracted watermark **b'(i, j)**. The obtained watermark value is used to detect the alterations or any tampering.

The watermark image should be kept secret or else an imposter can reconstruct the original image from the watermarked image.

**Restricted tables**: As the LUT's are constructed randomly, there will be few entries having the same value consecutively. For watermarking, the pixel values have to be adjusted by large amounts to get the desired value. In order to overcome this large adjustment, the number of consecutive entries of same value should be restricted. Like the number of consecutive 1's and 0's should be constant. By using such restrictions, deciphering the tables will be easy by the unauthorized parties. Thus using this scheme makes them less resistant to deciphering attacks.

**2) Using chaotic mixing of watermark images**

For 8-bit monochrome gray scaled image, around 256 entries are needed in the look-up-table (LUT) to decode the watermark. Extracting the watermark would be easy for an interceptor. To prevent this problem, chaotic mixing on the watermark image is used. This mixing transforms the watermark image into an unrecognizable noise form. By mixing the watermark to a random textured pattern, watermarking of monochrome gray scaled images such as fingerprints become more resistant to attacks.

The following formula is used for mixing the original watermark image to get the new ‘mixed’ watermark image.

$$r' = A * r,$$

where $A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$

**r** = (x, y) is a pixel in the original watermark image and

**r’** = (x’, y’) is the new pixel in the mixed watermark image.

This mixed watermark is embedded to get the watermarked image by using the look-up-table (LUT) method. It is similar to adding some noise into the original image as the mixed watermark is a noise pattern which is embedded.

After extracting the mixed watermark image, original watermark can be recovered from it. By watermarking the monochrome gray scaled fingerprint image with the original watermark image, causes the traces of the pattern to be surfaced when attacked by signal processing steps. This is because the error diffusion process is less effective in monochrome image than in color images as error can not be spread sufficiently in monochrome images. By using chaotic mixing, this problem is solved. When fingerprint images are watermarked with a mixed watermark image, along with error diffusion, the embedded information will not surface out when attacked with signal processing steps.

### 3.5.3 Digital Watermarking based Secure Multimodal Biometric System [22]

This watermarking algorithm provides two level of security, for verifying an individual and also protecting the biometric template. Iris template is embedded into the face image, visible face image is used for verification and the watermarked iris is used to cross authenticate the user and also secure the biometric data.

Iris template for embedding is generated by applying wavelet filters as they resist against the watermarking attacks such as rotation, adding noise, etc. 1D log Gabor wavelet filter is used to generate a binary iris template which is unique. Any changes in the bits, do not affect the matching performance.

Two algorithms are described, which are resistant to signal processing attacks:

1) Modified correlation based algorithm and,
2) Modified 2D discrete cosine transform based algorithm.

**Modified correlation based algorithm**

**Embedding**

Correlation properties of additive pseudo-random noise patterns are applied to the image for embedding iris template which is generated by applying the 1D log Gabor filter to the iris image. A secret key is used as a seed to the random number generator for generating the locations where the watermark can be embedded. An iris binary code is added to the face image by following the below block diagram (See Figure 3.15):

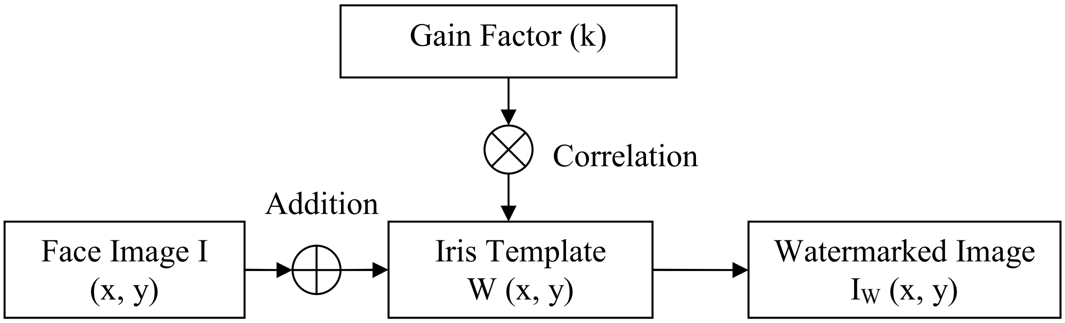


Figure 3.15: Embedding procedure for modified correlation based algorithm.

As ‘k’ is increased, the robustness of the watermark also increases at the cost of the quality of the watermarked image.

**Extracting**

Original image is not required to extract the watermark embedded. The secret key used for embedding and the iris code are presented to the system. For every location generated by the key, correlation between the noise pattern and the watermarked image is computed, if the result exceeds a certain threshold T, the watermark bit can be determined and that bit is set. Figure 3.16 shows the block diagram for the extracting procedure.

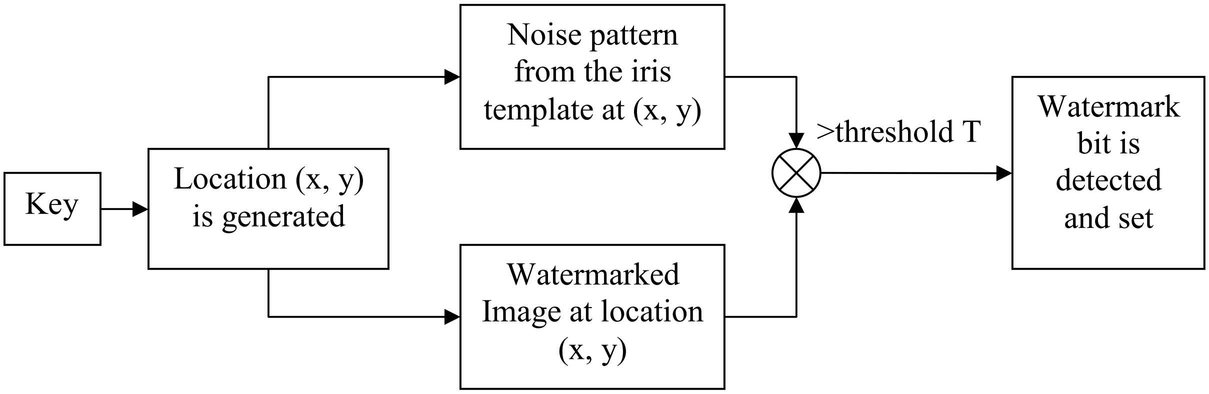

Key
Location (x, y) is generated
Noise pattern from the iris template at (x, y)
Watermarked Image at location (x, y)
>threshold T
Watermark bit is detected and set

Figure 3.16: Extracting procedure for modified correlation based algorithm.

The image can be divided into blocks and each block can be processed for a single bit, in order to embed multiple bits.

### Modified 2D discrete cosine transform based algorithm

**Embedding:**

Discrete Cosine Transform (DCT) is used for watermarking, as it can withstand to some of the signal processing attacks. The face image is divided into 8 x 8 blocks and DCT is applied to each block. After applying the transform, the entire image is represented as coefficients of different frequencies. The top left corner has the lowest frequency coefficient and the bottom right has the highest frequency coefficient. Figure 3.17 shows the block diagram for the embedding procedure.

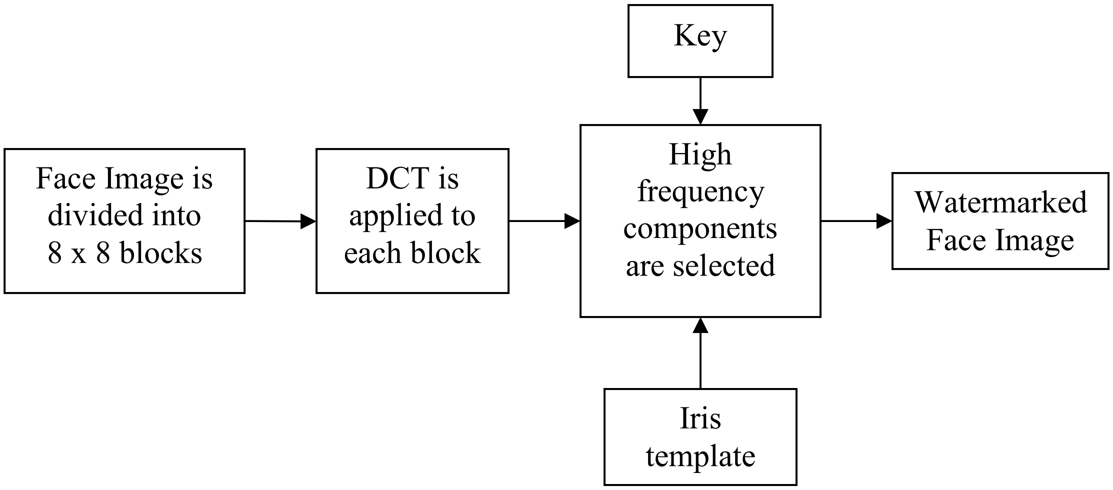

Key
Face Image is divided into 8 x 8 blocks
DCT is applied to each block
High frequency components are selected
Watermarked Face Image
Iris template

Figure 3.17: Embedding procedure for modified 2D discrete cosine transform based algorithm.

The binary iris template is embedded into the frequency domain of the original image. As the human vision system is sensitive to low frequency components, embedding a watermark in them would cause the visibility of the watermark to be high. So a random key is used to select the locations in the high frequency components to embed the iris template.

**Extracting**

Original image is not required for extracting the watermark. An inverse DCT is applied to the watermarked image. The watermark bits embedded are extracted by using the key to find out the locations of the watermark bits. The same key used for embedding is used in extracting stage. Figure 3.18 shows the block diagram for the extracting procedure.

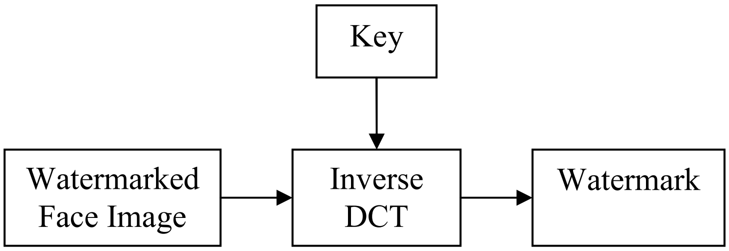


Figure 3.18: Extracting procedure for modified 2D discrete cosine transform based algorithm.

## 3.5.4 Methods Studied Practically [27]

### 3.5.4.1 Spatial Domain*:*

**Least Significant Bit:**

The watermark is embedded in the least significant bits of the cover image. As the entire cover image can be used for embedding, the watermark can be embedded multiple times, if the watermark used is small in size or dimensions. After signal processing attacks, at least one watermark out of multiple watermarks embedded, can be recovered.

The watermark which is embedded is an image which contains the text. This watermark is used rather than using the plain ASCII text directly, because a single bit error can change the meaning of the character which in turn can change the meaning of the whole text. When the ASCII text is compressed using JPEG technique, the output would be a group of random characters. And if an image with text is used, the degradation of the recovered watermark can be identified by a human eye.

When it comes to the robustness of this method against signal processing attacks, it can only withstand cropping. The main advantage of this method is that it can be applied to any image, but by replacing all the least significant bits in the cover image by '1' will destroy the watermark without much changes to the cover image [27].

### 3.5.4.2 Frequency Domain:

**Comparision of Mid - Band DCT Coefficients:**

In this technique, the JPEG quantization [44] values are used, so as to make the watermarked image resistant to lossy compressions.

By applying the DCT to the cover image, the image is divided into different frequency bands. By taking 8 x 8 block size, the DCT definition of each 8 x 8 blocks is as follows (See Figure 3.19):

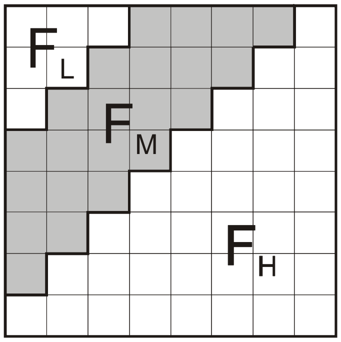


Figure 3.19: DCT definition of regions [27]

$F_L$ contains the low frequency components and any changes to the coefficients can be detected by the human eye. $F_H$ contains the high frequency components, which are the first ones to be attacked with noise or compressions. $F_M$ contains the middle frequency components, which are used for embedding the watermark. The changes to the coefficients in the $F_M$ region can not be detected by the human eye.

Two locations are selected from the $F_M$ region which is based on the JPEG quantization values. The two locations which have similar quantization values are selected for achieving robustness against compressions.

In the table in Figure 3.20, positions (5, 2) = 55 and (4, 3) = 56, have been selected for implementation. The DCT block encodes a ‘1’ if (5, 2) < (4, 3) and a ‘0’ if (5, 2) > (4, 3). In order to match to the bit which is encoded, the coefficients are swapped, but there will be no change in the watermarked image due to these swapping as the DCT coefficients in the $F_M$ regions have similar magnitudes. Another parameter called the ‘strength’ (k) is used i.e.

$$(5, 2) – (4, 3) > k$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 11 | 10 | 16 | 24 | 40 | 51 | 61 |
| 12 | 12 | 14 | 19 | 26 | 58 | 60 | 55 |
| 14 | 13 | 16 | 24 | 40 | 57 | 69 | 56 |
| 14 | 17 | 22 | 29 | 51 | 87 | 80 | 62 |
| 18 | 22 | 37 | **56** | 68 | 109 | 103 | 77 |
| 24 | 35 | **55** | 64 | 81 | 104 | 113 | 92 |
| 49 | 64 | 78 | 87 | 103 | 121 | 120 | 101 |
| 72 | 92 | 95 | 98 | 112 | 100 | 103 | 99 |

Figure 3.20: Quantization values used in JPEG compression scheme [27]

in order to increase the robustness of the embedded watermark. To achieve this, random noise is added to the image, so that the coefficients of (5, 2) and (4, 3) meet the above condition. The robustness of the watermark can be increased by increasing ‘k’, which in turn decreases the quality of the image [27].

## 3.6 Results of the study

In this section, two watermarking techniques are described in detail and applied to the iris biometrics. In spatial domain, least significant bit (LSB) technique has been implemented [27] and in frequency domain, comparision of mid - band DCT coefficients has been implemented [27]. Two watermark images have been used, small and medium copyright images. In LSB, the medium copyright image has been tiled, or embedded multiple times and in mid – band DCT coefficients technique, both small and medium copyright images have been used. The robustness of these techniques has been tested by applying Gaussian noise, random noise, degrading the quality by compressing the watermarked image and rotating the watermarked image.

MSE is an error metric which has been used to compare the extracted watermark image with the original watermark image. Mean square error is the average of the square of the difference between the original watermark image and the extracted watermark image. The lower the MSE value, the better it is.

$$MSE = \frac{1}{M * N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left( I(i,j) - I_W(i,j) \right)^2 ,$$

where I is the original watermark image,

$I_W$ is the extracted watermark image and

M, N are the dimensions of the image (same for both the images as size of the extracted watermark image is same as the original watermark image).

PSNR is another error metric which is based on MSE. The PSNR has been calculated between the original cover image and the watermarked image.

$$PSNR = 20 \log_{10} \left( \frac{255}{\sqrt{MSE}} \right),$$

255 is taken because in our iris image, each pixel is represented using 8 bits, so ($2^8$-1). As MSE is inversely proportional to PSNR, higher the PSNR value, the better it is, which means the ratio of signal to noise is higher. PSNR does not take human visual system into account.

Random noise is a disturbance added to the watermarked image. A matrix of size of watermarked image whose entries are uniformly distributed random numbers in the interval [0, 1] is added to the watermarked image. Before adding, the matrix is multiplied with a multiplier to increase the magnitude of disturbance added to the watermarked image. Gaussian noise follows normal distribution with constant mean and variance.

The quality of the image is degraded by using JPEG compression [44]. The lesser the quality value, the higher the degradation of the image due to JPEG compression. The compression ratio will be mentioned in the results. Compression ratio is calculated using the following formula [46]:

$$Compression\_ratio = \frac{M * N * bit\_depth}{8 * file\_size},$$

where M and N are the dimensions of the compressed image,

bit depth is the number of bits required to represent each pixel and

file size is the space occupied by the image in bytes.

By compressing an image, the quality of that image degrades.

Image is rotated in counter-clockwise direction with the specified angle in degrees. Nearest neighbor interpolation is used, where pixel in the new image is the nearest pixel in the original image.

The two coefficients which are selected by having same quantization value, their respective positions in the DCT coefficients $B_1(x_1, y_1)$ and $B_2(x_2, y_2)$ are selected only if they meet the following condition:

$$B_1(x_1, y_1) - B_2(x_2, y_2) > k$$

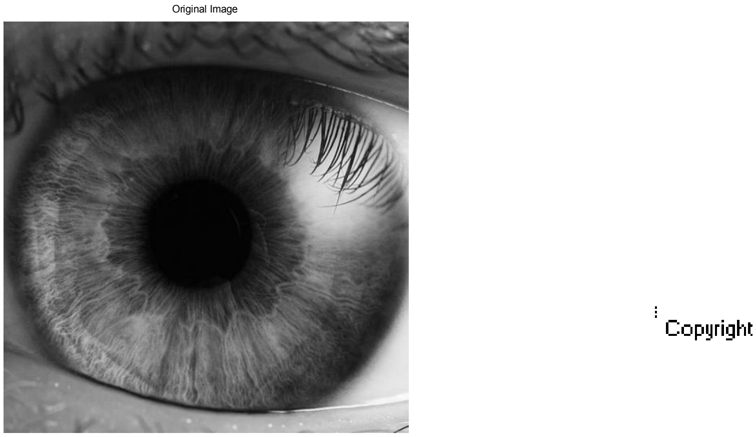
Even if noise is added, the embedded data can be recovered if the value of k is large. ‘k’ which is denoted as the watermark strength, the larger it is, the robustness of the image against signal processing attacks increases compromising the quality of the image. All the results are obtained for strength value k = 50.

## 3.6.1 Spatial Domain

**Least Significant Bit:**

A gray-scale iris image of size 512 x 512 is selected as a cover image, and the watermark to be embedded is an image of size 50 x 20 consisting of text. This watermark is embedded multiple times in the cover image, so that at least one watermark can be recovered when the cover image under goes signal processing attacks. Figure 3.21 displays the cover and watermark images used in the experiments.

As each pixel in the watermark image uses two bits, either ‘0’ or ‘1’, one least significant bit in the cover image (which has bit depth = 8) is used to hide the tiled watermark.

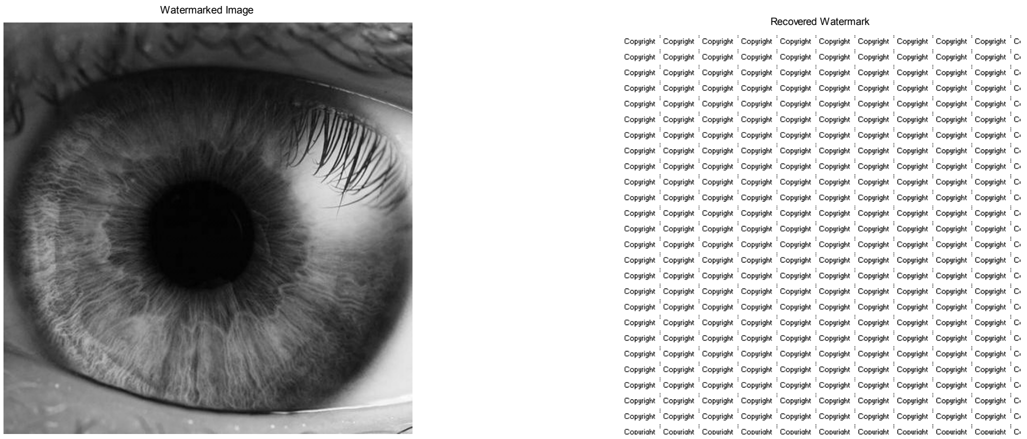


Original Image (512x512) Watermark (50x20) – medium

Figure 3.21: The left panel shows the cover image used in our experiment. The right panel shows the watermark image.

At the recovery end, the one least significant bit is used to recover the embedded watermark. The cover image is watermarked at PSNR = 1.2632e+005 dB.

The watermarked image looks similar to the original image with small changes which can not be detected by human eye. (See Figure 3.22).



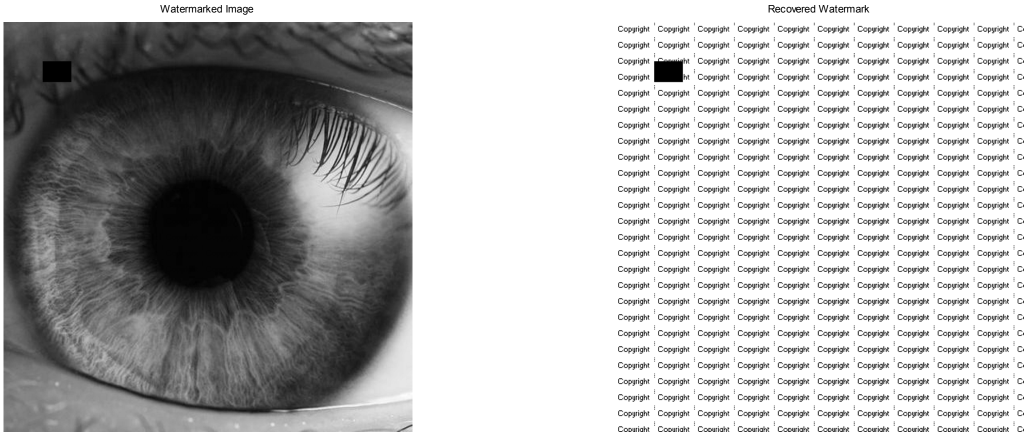
Watermarked Image PSNR = 1.2632e+005 dB

Extracted Watermark MSE = 0.0905

Figure 3.22: The left panel shows the watermarked image obtained. The right panel displays the extracted watermark image.

The embedded watermark has been recovered, but when the watermarked image is tampered, by assigning pixels in a small region of the watermarked image to zero (on the left top corner), the recovered watermark image shows those changes as shown in Figure 3.33. This is a fragile watermarking technique where tampering can be detected.

.

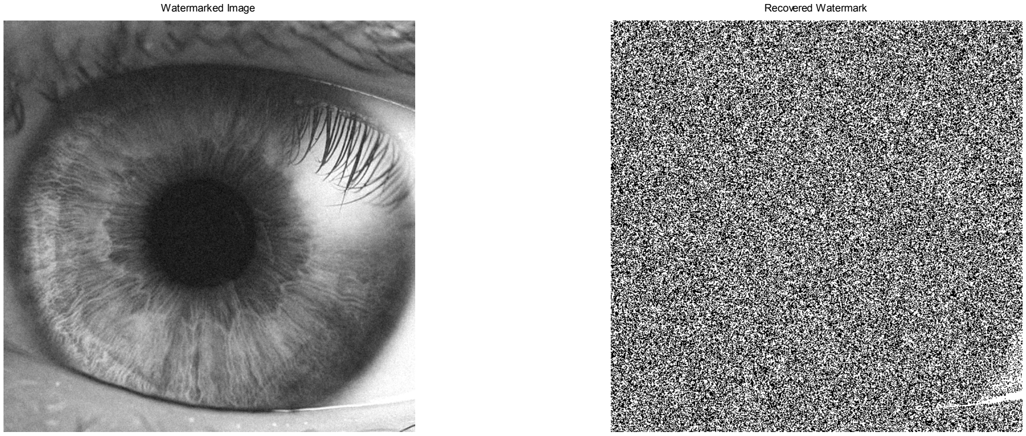


Tampered Watermark Image

Detecting the tampered region in the watermark

Figure 3.23: The left panel shows the tampered watermarked image. The right panel displays the tampered region in the extracted watermark image.

When the watermarked image is attacked by Gaussian noise, the watermark which is embedded is destroyed and can not be recovered. (See Figure 3.24). Here we involved additive Gaussian noise with mean 0.1 and variance 0.001. Note that the presence of the additive noise reduced PSNR to 88.0229 dB. The right panel in Figure 3.24 shows the result after watermark reconstruction.

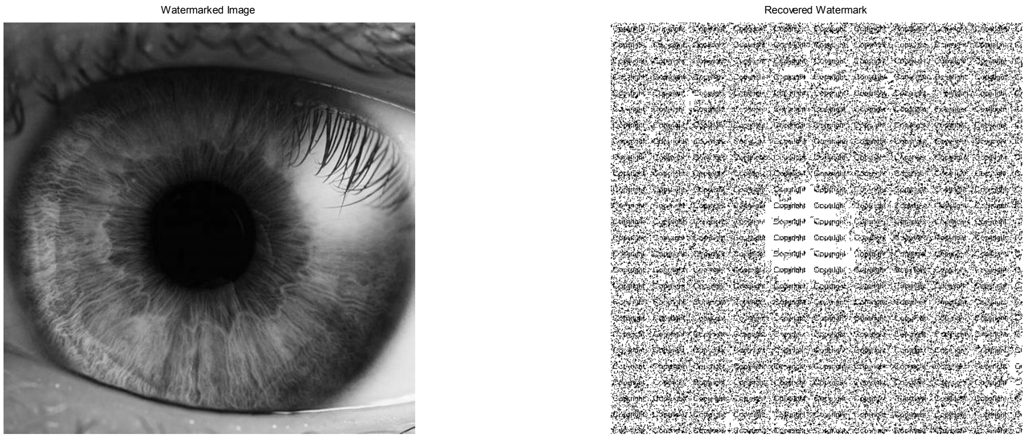


Watermarked Image with Gaussian noise
Mean = 0.1 and Variance = 0.001
PSNR = 88.0229dB

Extracted Watermark
MSE = 0.4438

Figure 3.24: The left panel shows the watermarked image with Gaussian noise. The right panel displays the extracted watermark image.

When the quality of the watermarked image is degraded, by compressing the image using JPEG compression, the watermark can not be recovered.



Watermarked Image with Quality degradation | Extracted Watermark

Quality = 99 or compression ratio of 2.5792:1 | MSE = 0.2205

PSNR = 9.4216e+004

Figure 3.25: The left panel shows the watermarked image with quality degradation. The right panel displays the extracted watermark image.

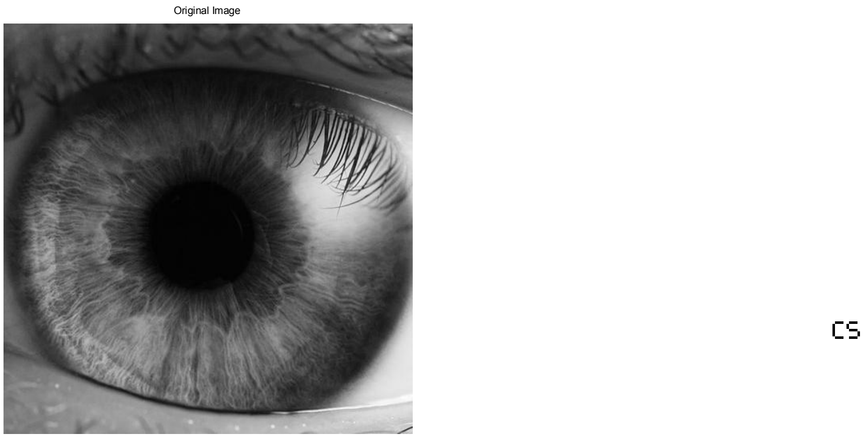
For quality degradation of 99 or compression ratio of 2.5792:1, the traces of the embedded watermark can be seen in the extracted watermark as shown in the right panel of Figure 3.25.

Based on our observations and the results of the experiments this technique could not withstand any of the signal processing attacks as the watermark pixel values are embedded directly into the cover image.

### 3.6.2 Frequency Domain

**Comparision of Mid – Band DCT Coefficients:**

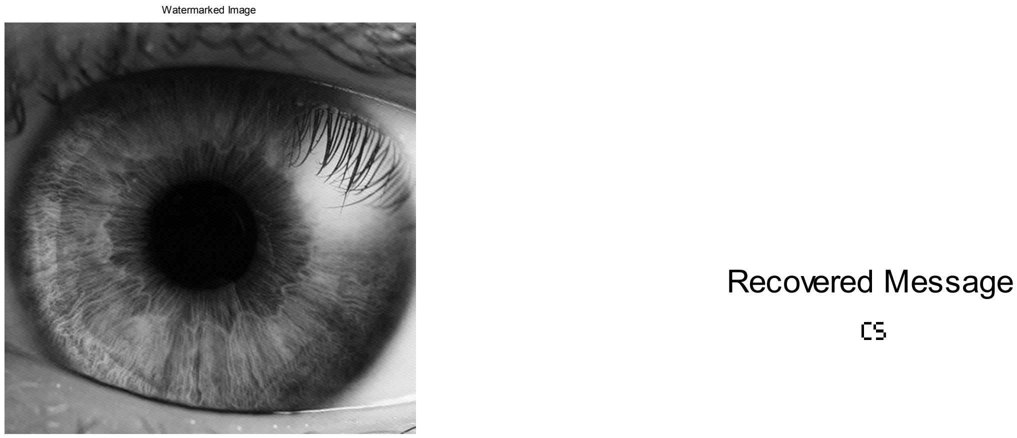
The second method performs watermarking in transform, DCT, domain. To perform watermarking, the image is processed with the DCT applied to image blocks of size 8 x 8. The obtained coefficients are manipulated for embedding the watermark image. Both, small and medium copyright images are used for testing and result in similar outcomes. Figure 3.26 shows the cover image and the small watermark image used in our experiments. Figure 3.27 shows the watermarked image and the perfectly recovered small watermark image. Figure 3.28 shows the medium watermark image used in our experiment, watermarked image with medium watermark embedded and the recovered medium watermark.



Original Image (512x512) Watermark (12x9) - small

Figure 3.26: The left panel shows the original cover image. The right panel shows the small watermark image.

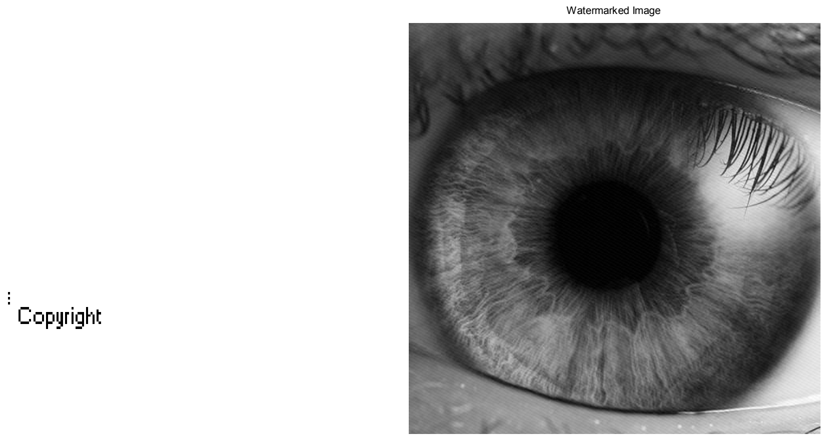
Two positions are selected in the JPEG quantization table, such that both have similar quantization values. The DCT value of these positions in the cover image are compared say $B_1$ and $B_2$, if $B_1 < B_2$, then the encoded bit is ‘1’ else a ‘0’. In order to match to the bit which is encoded, the coefficients are swapped, but there will be no change in the watermarked image due to these swapping as the DCT coefficients in the middle frequency regions have similar magnitudes.



Watermarked Image
PSNR = 2.5118e+003

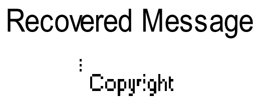
Extracted Watermark Image
MSE = 0

Figure 3.27: The left panel shows the watermarked image. The right panel displays the perfectly recovered small watermark.



(a) Watermark (50x20) – medium

(b) Watermarked Image
PSNR = 2.5098e+003

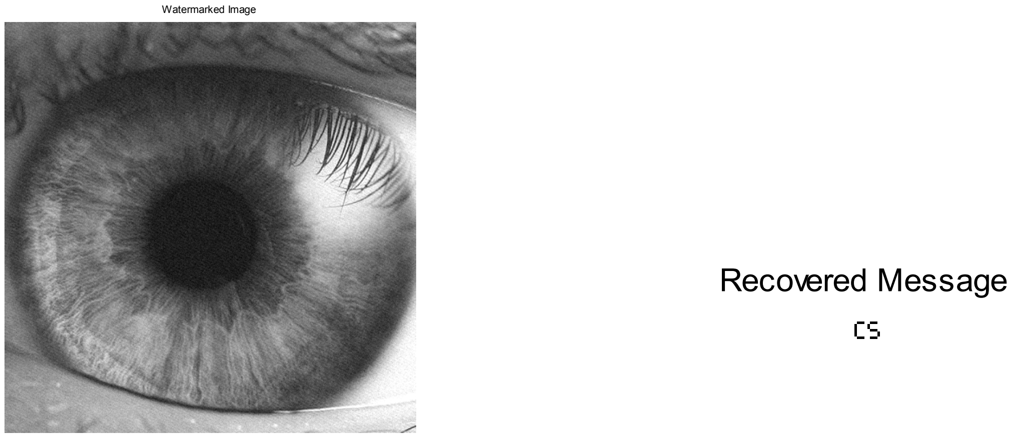


(c) Extracted Watermark Image

MSE = 0.006

Figure 3.28: (a) shows the medium watermark image, (b) shows the watermarked image with medium watermark embedded and (c) displays the recovered medium watermark.

When Gaussian noise is added to watermarked image, the embedded small watermark is resistant up to the noise variance value $\sigma^2 = 0.01$ and then gets affected by the noise and the watermark can not be recovered. But when noise variance value $\sigma^2 = 0.001$, the embedded small watermark can be recovered perfectly as shown in the right panel of Figure 3.29 At the noise level $\sigma^2 = 0.01$, the small watermark is moderately recovered. Table 3.1 summarizes the results of our experiment.



Watermarked Image with Small Watermark

Mean = 0.1, Variance = 0.001

PSNR = 87.2047

Extracted Watermark image

MSE = 0

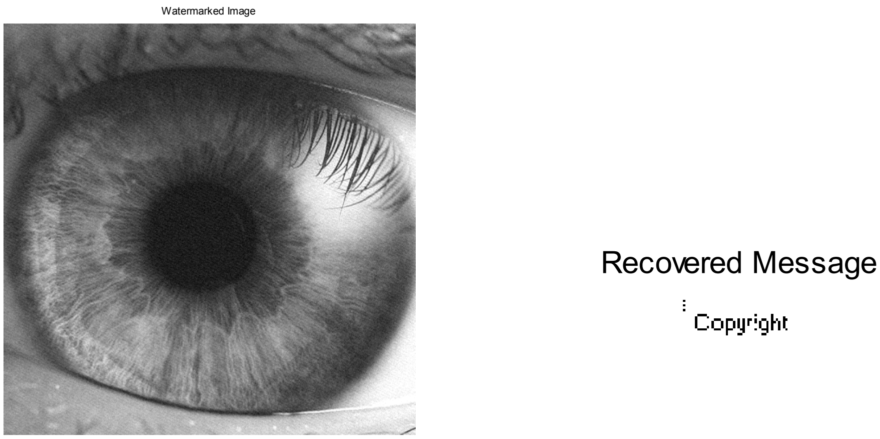
Figure 3.29: The left panel shows the watermarked image with Gaussian noise. The right panel displays the perfectly recovered small watermark.

Note that PSNR in this case is similar to PSNR in Figure 3.24.

Table 3.1: Gaussian Noise applied to watermarked Image with small watermark

| Mean | Variance | MSE | PSNR | Recovered Watermark |
|---|---|---|---|---|
| 0.1 | 0.001 | 0 | 87.2047 | Recovered |
| 0.1 | 0.005 | 0.0278 | 65.4248 | Recovered |
| 0.1 | 0.007 | 0.0556 | 58.0922 | Recovered |
| 0.1 | 0.01 | 0.0833 | 49.9303 | Moderately Recovered |
| 0.1 | 0.012 | 0.1204 | 45.6284 | Not Recovered |

When the noise variance value is $\sigma^2 = 0.001$, the embedded medium watermark can be recovered as shown in the right panel of Figure 3.30 At the noise level $\sigma^2 = 0.003$, the medium watermark is moderately recovered.



Watermarked Image with Medium Watermark

Mean = 0.1, Variance = 0.001

PSNR = 87.3163

Extracted Watermark image

MSE = 0.0060

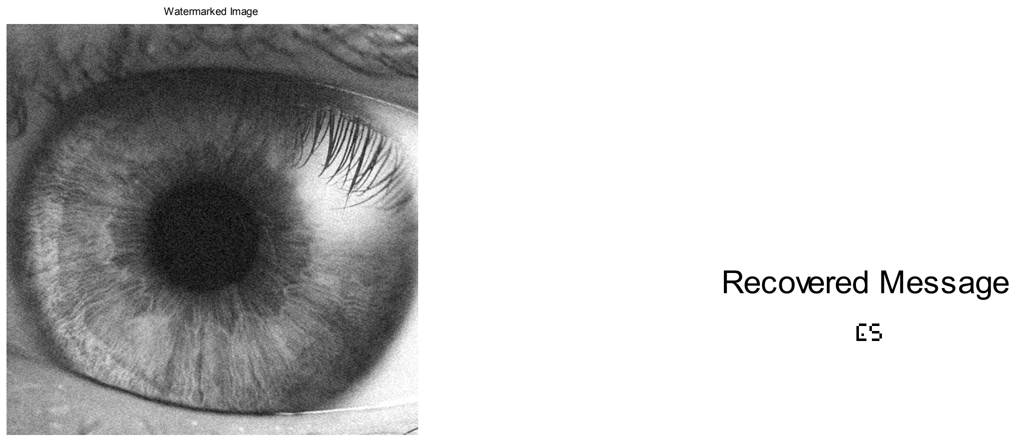
Figure 3.30: The left panel shows the watermarked image with Gaussian noise. The right panel displays the recovered medium watermark.

Table 3.2 summarizes results for the case when the medium size watermark is added to DCT coefficients. Note that the quality of the reconstructed watermark depends on its size in spite of the value of PSNR, which is kept approximately same.

Table 3.2: Gaussian Noise applied to watermarked Image with medium watermark

| Mean | Variance | MSE | PSNR | Recovered Watermark |
|---|---|---|---|---|
| 0.1 | 0.001 | 0.0060 | 87.3163 | Recovered |
| 0.1 | 0.003 | 0.0130 | 74.7727 | Moderately Recovered |
| 0.1 | 0.005 | 0.0230 | 65.3722 | Moderately Recovered |
| 0.1 | 0.007 | 0.0470 | 57.8661 | Not Recovered |
| 0.1 | 0.01 | 0.0750 | 49.7113 | Not Recovered |

When random noise is applied, the embedded small watermark can be recovered up to a multiplier of 52 (See Figure 3.31), but increasing the value of the multiplier for the random noise, the small watermark gets destroyed. Table 3.3 summarizes the results.



Watermarked Image with Small Watermark

Multiplier = 52

PSNR = 70.3931

Extracted Watermark image

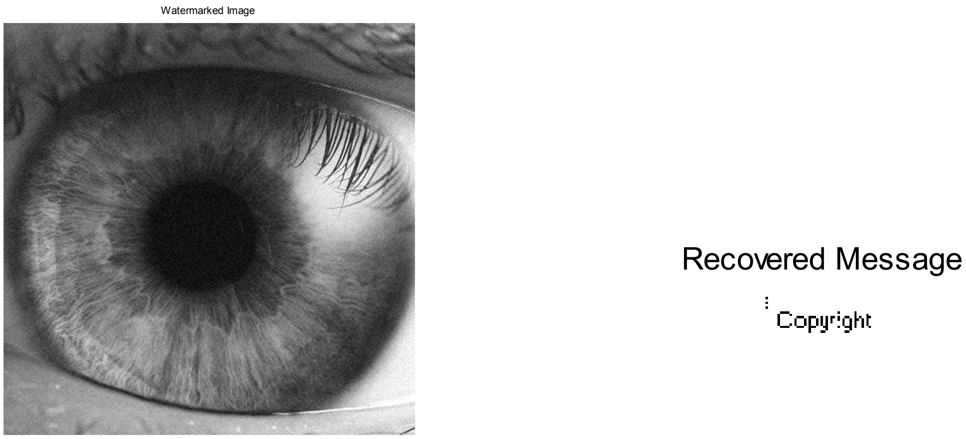
MSE = 0

Figure 3.31: The left panel shows the watermarked image with Random noise. The right panel displays the perfectly recovered small watermark.

Table 3.3: Random Noise applied to watermarked Image with small watermark

| Multiplier | MSE | PSNR | Recovered Watermark |
|---|---|---|---|
| 52 | 0 | 70.3931 | Recovered |
| 53 | 0.0093 | 67.6227 | Recovered |
| 54 | 0.0093 | 65.2410 | Recovered |
| 55 | 0.0278 | 62.9514 | Recovered |
| 60 | 0.0093 | 53.3722 | Recovered |

The embedded medium watermark is resistant to random noise upto a multiplier of 30 (See Figure 3.32) and if the multiplier value is further increased, the medium watermark gets destroyed. Table 3.4 summarizes the results when random noise is applied to watermarked image with medium watermark.



Watermarked Image with Medium Watermark Image
Multiplier = 30
PSNR = 203.2120

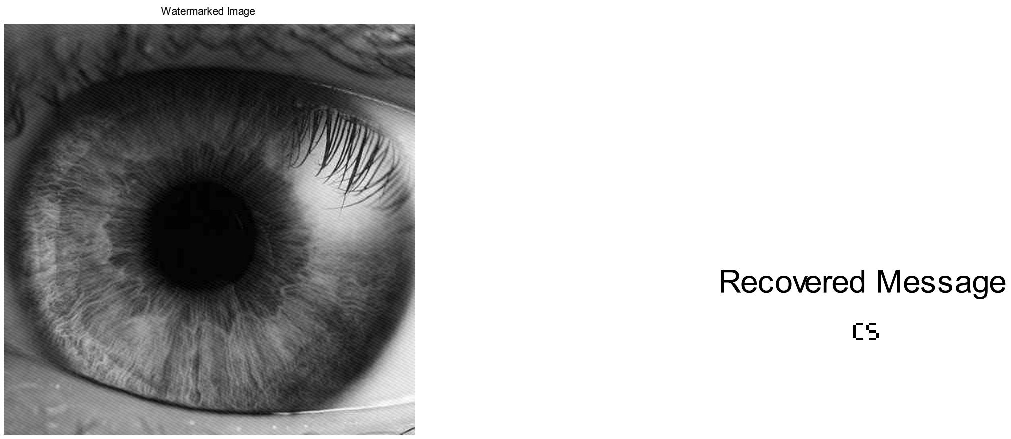
Extracted Watermark image
MSE = 0.006

Figure 3.32: The left panel shows the watermarked image with Random noise. The right panel displays the recovered medium watermark.

Table 3.4: Random Noise applied to watermarked Image with medium watermark

| Multiplier | MSE | PSNR | Recovered Watermark |
|---|---|---|---|
| 30 | 0.0060 | 203.2120 | Recovered |
| 33 | 0.0080 | 169.5732 | Recovered |
| 35 | 0.0090 | 151.4722 | Recovered |
| 40 | 0.0130 | 117.4614 | Recovered |
| 50 | 0.0260 | 76.3209 | Recovered |

Quality of the watermarked image is degraded by compressing the watermarked image using JPEG compression technique to estimate the robustness of the technique. And the technique withstands the quality degradation up to 28 or compression ratio 11.9996:1 for small watermark (See Figure 3.33). Table 3.5 presents the results when compression is applied to watermarked image with small watermark.



Watermarked Image with Small Watermark

Quality = 28 or compression ratio 11.9996:1

PSNR = 1.0596e+003

Extracted Watermark image

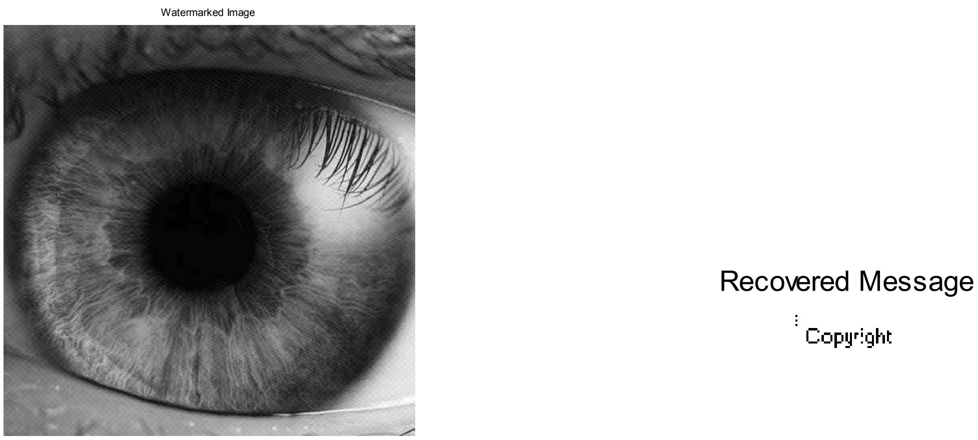
MSE = 0

Figure 3.33: The left panel shows the watermarked image with quality degradation of 28 or compression ratio of 11.9996:1. The right panel displays the perfectly recovered small watermark.

Table 3.5: Quality degradation is applied to watermarked Image with small watermark

| Quality | Compression ratio | MSE | PSNR | Recovered Watermark |
|---|---|---|---|---|
| 28 | 11.9996:1 | 0 | 1.0596e+003 | Recovered |
| 27 | 14.3436:1 | 0.1667 | 1.7166e+003 | Not Recovered |
| 26 | 15.8673:1 | 0.3981 | 1.9805e+003 | Not Recovered |
| 25 | 16.6779:1 | 0.3056 | 1.8859e+003 | Not Recovered |
| 24 | 18.0752:1 | 0.3426 | 1.7981e+003 | Not Recovered |

When quality degradation up to 28 or compression ratio 12.0194:1 is applied to watermarked image with medium watermark, the watermark can be recovered (See Figure 3.34). Table 3.6 presents the results when compression is applied to watermarked image with medium watermark.



Watermarked Image with Medium Watermark Image Extracted Watermark image

Quality = 28 or compression ratio 12.0194:1 MSE = 0.006

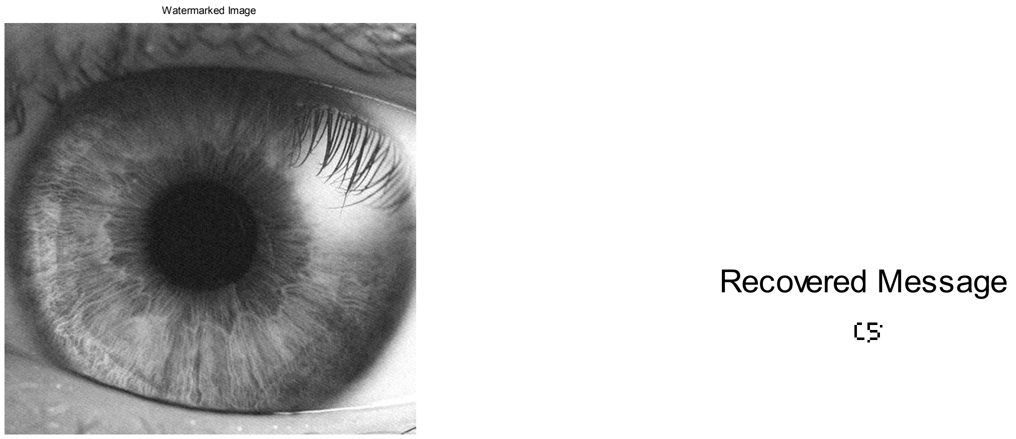
PSNR = 789.6005

Figure 3.34: The left panel shows the watermarked image with quality degradation of 28 or compression ratio of 12.0194:1. The right panel displays the recovered medium watermark.

Table 3.6: Quality degradation is applied to watermarked Image with medium watermark

| Quality | Compression ratio | MSE | PSNR | Recovered Watermark |
|---|---|---|---|---|
| 28 | 12.0194:1 | 0.0060 | 1.0596e+003 | Recovered |
| 27 | 14.3460:1 | 0.1400 | 1.7116e+003 | Not Recovered |
| 25 | 16.6642:1 | 0.2640 | 1.8848e+003 | Not Recovered |
| 23 | 18.8254:1 | 0.2160 | 1.7003e+003 | Not Recovered |
| 20 | 19.5908:1 | 0.1810 | 1.4478e+003 | Not Recovered |

Gaussian noise and quality degradation are applied together and this technique can withstand quality of 36 or compression ratio 10.0701:1 for the small copyright image (See Figure 3.35 and Table 3.7) and quality degradation of 70 or compression ratio 5.9025:1 for the medium copyright image (See Figure 3.36 and Table 3.8).



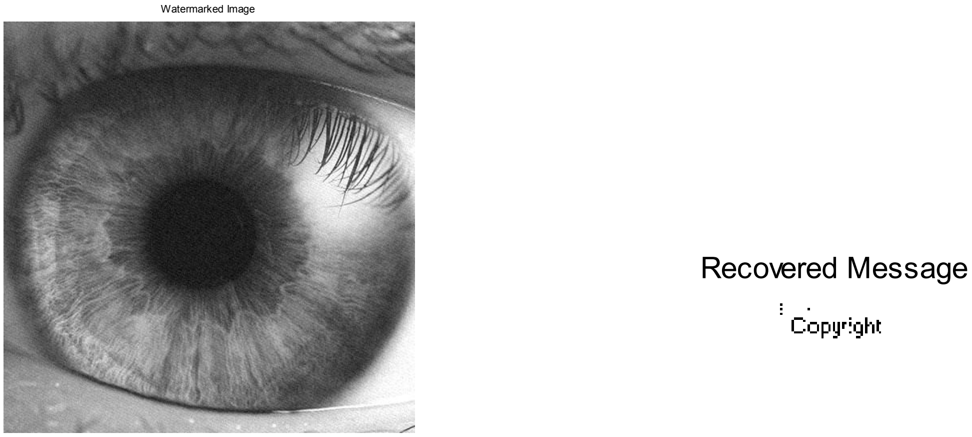
Watermarked Image with Small Watermark
Quality = 36 or compression ratio 10.0701:1,
Mean = 0.1, Variance = 0.001
PSNR = 88.4642

Extracted Watermark image
MSE = 0.0370

Figure 3.35: The left panel shows the watermarked image with Gaussian noise and quality degradation with compression ratio of 10.0701:1. The right panel displays the recovered small watermark.

Table 3.7: Gaussian noise and quality degradation are applied to watermarked Image with small watermark

| Mean | Variance | Quality | Compression ratio | MSE | PSNR | Recovered Watermark |
|---|---|---|---|---|---|---|
| 0.1 | 0.001 | 36 | 10.0701:1 | 0 | 87.4774 | Recovered |
| 0.1 | 0.005 | 40 | 6.5587:1 | 0.0463 | 65.4643 | Moderately Recovered |
| 0.1 | 0.005 | 45 | 5.8823:1 | 0.0278 | 65.2641 | Moderately Recovered |
| 0.1 | 0.005 | 50 | 5.3463:1 | 0.0833 | 65.3058 | Not Recovered |
| 0.1 | 0.007 | 50 | 4.7177:1 | 0.0463 | 58.0898 | Not Recovered |
| 0.1 | 0.009 | 50 | 4.3049:1 | 0.0833 | 52.3030 | Not Recovered |



Watermarked Image with Medium Watermark Image
Quality = 70 or compression ratio 5.9025:1,
Mean = 0.1, Variance = 0.001
PSNR = 87.2265

Extracted Watermark image
MSE = 0.006

Figure 3.36: The left panel shows the watermarked image with Gaussian noise and quality degradation with compression ratio of 5.9025:1. The right panel displays the recovered small watermark.

Table 3.8: Gaussian noise and quality degradation are applied to watermarked Image with medium watermark

| Mean | Variance | Quality | Compression ratio | MSE | PSNR | Recovered Watermark |
|---|---|---|---|---|---|---|
| 0.1 | 0.001 | 70 | 5.9025:1 | 0.0060 | 88.3781 | Recovered |
| 0.1 | 0.001 | 69 | 6.0558:1 | 0.0070 | 88.3505 | Recovered |
| 0.1 | 0.001 | 65 | 6.5945:1 | 0.0070 | 88.2254 | Recovered |
| 0.1 | 0.005 | 65 | 3.9679:1 | 0.0670 | 65.7851 | Moderately Recovered |
| 0.1 | 0.005 | 60 | 4.4230:1 | 0.0790 | 65.7104 | Moderately Recovered |
| 0.1 | 0.01 | 60 | 3.5390:1 | 0.1380 | 50.0720 | Not Recovered |

This technique is not robust against rotation of the watermarked image. Table 3.9 and 3.10 present the results when the watermarked image is rotated using nearest interpolation technique. The watermark cannot be recovered in any of the cases.

Table 3.9: Rotation applied to the watermarked image with small watermark.

| Type | Value | MSE | PSNR | Recovered Watermark |
|---|---|---|---|---|
| nearest | 2 | 0.4259 | 84.4342 | Not Recovered |
| nearest | -2 | 0.3981 | 82.1171 | Not Recovered |

Table 3.10: Rotation applied to the watermarked image with medium watermark.

| Type | Value | MSE | PSNR | Recovered Watermark |
|---|---|---|---|---|
| nearest | 2 | 0.4930 | 84.4342 | Not Recovered |
| nearest | -2 | 0.5050 | 82.1163 | Not Recovered |

# Chapter 4

## Steganography

### 4.1 What is steganography?

It is an art and science of hiding information by embedding messages within other, which are seemingly harmless messages [28]. Steganography is derived from Greek where “steganos” means covered and “graphie” means writing. It means covered or hidden writing. The advantage of steganography is that the message in which sensitive data is hidden does not attract attention to interceptors [6]. This process needs the cover image and secret image. The cover image hides the data of the secret image.

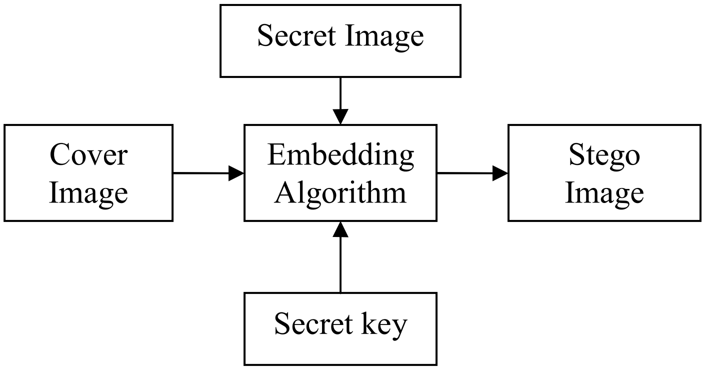


Figure 4.1: Stego embedding procedure [19]

Secret image is hidden in the cover image by the embedding algorithm which uses a secret key. This secret key is used to determine the location in cover image in which data of the secret image can be embedded. Cover image is transformed to stego image which is transmitted through an unsecured communication channel. This stego image is not suspected as it is similar to cover image, which is selected in such a way that it will not attract any attention by the interceptor. This is the embedding procedure. (Shown in Figure 4.1).

At the extracting end, the transmitted stego image along with the secret key, which was used at the embedding stage, is given to the extracting algorithm. The output is the secret image. At the extracting stage, if the key given to the extracting algorithm is the same used at the embedding stage and also the stego image is the same which was sent, then the secret image is extracted accurately, i.e., there was no tampering done to the stego image. (See Figure 4.2).

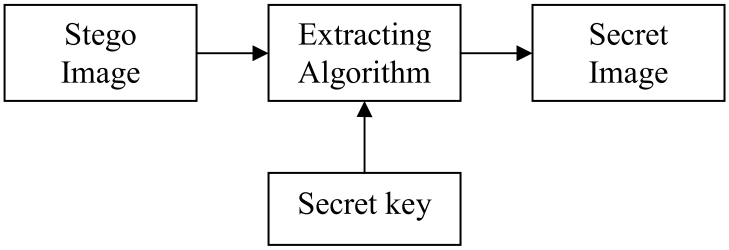


Figure 4.2: Stego extracting procedure [19]

If the stego image was tampered or a different key was presented, then the secret image will not be extracted accurately. After extracting, the cover image is discarded so that it can not be used again as a cover image.

## 4.2 Purpose of steganography (or) Why steganography?

The main purpose of steganography is to hide the occurrence of the communication itself along with the secret image from the third party [29]. Even if extracting the secret image is difficult or impossible, having the knowledge that the cover image will be transmitted, it will raise suspicion. If the presence of the secret image is detected or revealed or even suspected, then the purpose of steganography is defeated, even if the secret image is not extracted [30].

## 4.3 Applications

Steganography can be used for good and also bad purpose. Steganography is mainly used in military and government applications for protecting the sensitive data [31]. As most of the data is stored on computers and transmitted through the network, it is anticipated that the use of steganography for transmitting the data will increase. Publishing and broadcasting industries are using this technique for hiding their encrypted copyrights, serial numbers, etc. in books and multimedia files. Few countries's, government restricts the use of encryption for many purposes, so steganography is used for hiding sensitive data. It is also being used in medical field. Information such as patient's name, data of admittance, physician's name is hidden in medical images so that the patient's image is prevented from mingling with other patient's images. This is a safety measure, which is being used [33]. Steganography has many nefarious applications such as hiding data related to illegal activities, financial fraud, industrial espionage and communication among members of criminal or terrorist organizations [32]. Terrorist organizations are using steganography along with encryption for secured communication [31].

## 4.4 Types of Steganography

### 4.4.1 Attacks on a steganographic system [36]

Attacks on the steganographic system can be passive and active. In a passive attack, the attacker will only be able to get the data where as in an active attack, the attacker will be able to make changes to the data which he/she intercepts. There can be six types of attacks on a steganographic system.

1) **Stego - Only attack**: The attacker gets the stego data which is sent and can also analyze the data [36]
2) **Stego attack**: The sender sends different messages by using a single cover, every time he/she sends. The attacker has all the stego files containing different messages.
3) **Cover - Stego attack**: The attacker has the stego file which was sent by the sender and also has the knowledge of the cover file which was used by the sender in that process.
4) **Cover - Embed - Stego attack**: The attacker has the stego file, has the knowledge of which cover file was used in that process along with the message embedded.
5) **Manipulating the stego data**: The attacker can make changes to the stego data and is also capable of extracting or removing the secret message from the stego data.
6) **Manipulating the cover data**: The attacker can make changes to the cover data and also intercept the resulting stego data. By this attack, the attacker will be able to determine easily whether the stego data has any hidden secret message or not.

Steganography is sub-divided into many types based on different aspects.

### 4.4.2 Based on the application [34]

This is sub-divided into fragile steganography and robust steganography. Figure 4.3 shows the classification of steganography based on the application.

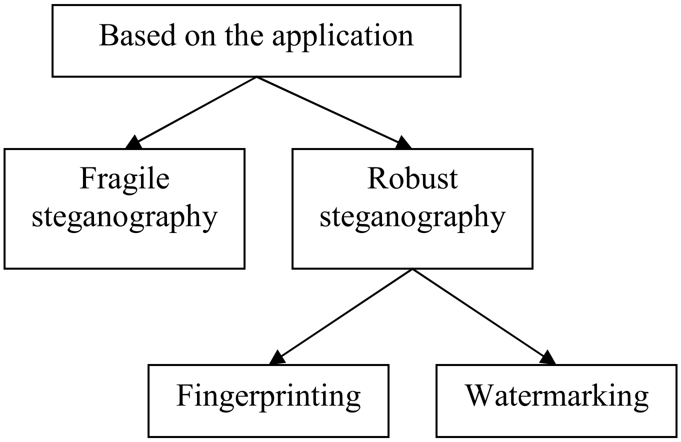


Figure 4.3: Classification of steganography – Type I

#### 4.4.2.1 Fragile steganography

This technique embeds information into a file. The embedded information can be destroyed easily, if the cover file is modified. This is used in situations where it is important to prove that the file is not tampered such as presenting a digital file in court of law as evidence, as the slightest change to the file will remove the embedded information [34].

#### 4.4.2.2 Robust steganography

This technique embeds information into a file which cannot be destroyed easily. No mask is indestructible, but a system is considered to be robust if the amount of changes required to remove the embedded information will make the file useless. This technique is sub-divided into fingerprinting and watermarking. **Fingerprinting** hides a unique identifier for the customer who has obtained the file and has rights to use it. If that file is found with someone other than the identified customer, the copyright owner of the file, by using fingerprinting, can trace which customer violated the license agreement by distributing the copy of the file. Watermarking involves embedding the identity of the copyright owner of the file, not the customer identity. It helps in prosecuting those who have an illegal copy where as fingerprinting are used to identify the customers who violate the license agreement. Fingerprinting technique is ideal for multimedia data, but when it comes for large production, it is not feasible to give each file a separate fingerprint [34].

### 4.4.3 Based on the keys used for steganography [31, 35]

This category also conveys the level of security with which the stego message is embedded, transmitted and extracted. This is sub-divided into pure, private-key or shared-key and public-key steganography [31, 35]. Figure 4.4 shows the classification of steganography based on the keys used.

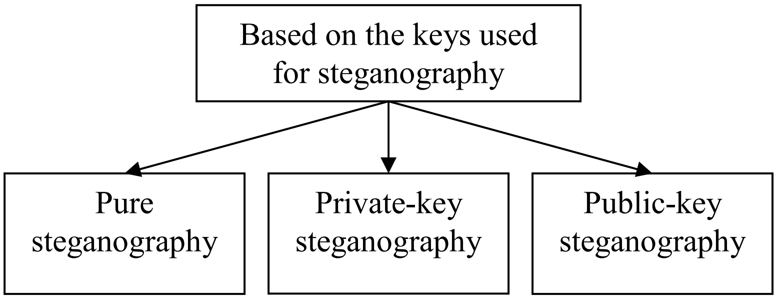

Based on the keys used for steganography
Pure steganography
Private-key steganography
Public-key steganography

Figure 4.4: Classification of steganography – Type II

#### 4.4.3.1 Pure steganography

This technique is the least secure method as it uses no keyed system to embed secret data into cover image. It does not require prior exchange of keys or data.

The embedding function $E: (C, M) \rightarrow C$ and the extracting function $D: C \rightarrow M$ and $D (E (C, M)) = M$ is the property of the pure steganographic system where C is the cover image, M is the secret data or message such that $|M| < = |C|$. This system's security lies in two aspects, the fact that only sending and receiving parties know of the secret message's existence and which steganographic algorithm was used to hide the message.

#### 4.4.3.2 Private-key (or) Shared-key steganography

In the keyed steganographic process, a secret message is first encrypted and then embedded into the cover image using one of the several steganographic techniques. At the receiving end, the encrypted secret message is extracted and then the message is decrypted using an appropriate key and algorithm. Adding encryption method to steganography increases the security of the whole system.

In the private-key steganography system, the secret key is exchanged. That mutual key is used to encrypt the secret message and the encrypted secret message is embedded into the cover data. The same mutual secret key is used to decrypt the secret message after extracting it from the cover data. As this system requires both the parties to know the key, once the key is compromised, the entire system looses its security.

#### 4.4.3.3 Public-key steganography

In this technique, two keys are used to add a layer of robustness to the process. There are two keys i.e. public key, which is obtained from the public database and private key, which is secret to each user. The public key of the recipient is used to encrypt the secret message and then is embedded into the cover. At the receiving end, the encrypted secret message is extracted and decrypted only by using received user's private key. This is the most secure type of steganography among the three as it combines the benefits of hiding the existence of a secret message with the security of encryption [31, 35].

### 4.4.4 Based on steganographic methods used [36]

It is sub-divided into three methods: injection, substitution and the generation of a new file. (See Figure 4.5).

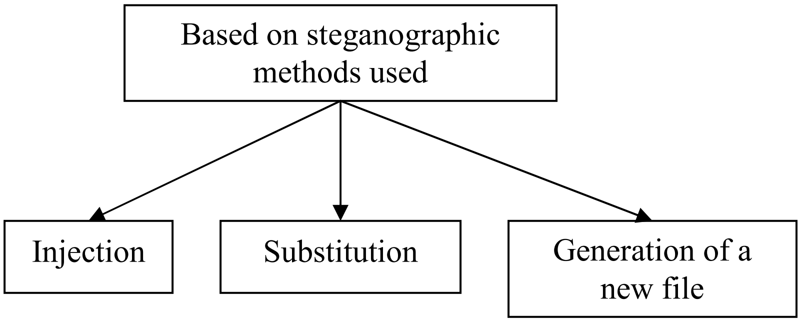


Figure 4.5: Classification of steganography – Type III

### 4.4.4.1 Injection

Injection hides the secret data in parts of the cover file which will be ignored by the application. The data can be hidden in comment tags or holes in the cover file.

### 4.4.4.2 Substitution

Substitution replaces the insignificant data in the cover file with the secret data. Least significant bit in the cover file can be replaced by the secret data.

### 4.4.4.3 Generation of a new file

Steganography can be implemented without the need for a cover file, the secret data generates a new file which is sent through the unsecured communication channel. The secret data is used as an input to a system which generates a new file.

Three common methods used to hide information in digital images are

- Least significant bit insertion
- Masking
- Transformations
- Spread Spectrum Image Steganography (SSIS)

**Least significant bit insertion (LSB)**

Each pixel is represented by using 8-bits in an 8-bit image. Human eye can detect around 6 – 7 bits of color. Changes in the most significant bits (MSB) can be noticed but least significant bits can be used to hide data as the changes in these bits are unnoticed. The least one or two bits can be used for this purpose. This method is simple to implement, has high capacity and low perceptibility. If the fact that the data is hidden in the cover image using the LSB method is known, extraction of the hidden data becomes easy and the cover image can be attacked by cropping or compressing [36].

**Masking**

Most significant bits are used to embed the secret data. This hidden data cannot be removed by signal processing attacks as it is stored in significant bits and removing them might cause loss of information which could be seen visually. It avoids attackers from attacking the image. There are few regions in an image which are busy or crowded, which might have significant data. By increasing or decreasing the luminance in those regions, secret data can be embedded. It is non-perceptible to human eye [36].

**Transformations**

The most commonly used transformations are discrete cosine transform (DCT), fast fourier transform (FFT) and discrete wavelet transform (DWT). The cover image is transformed using one of the above transforms and the obtained coefficients are changed to embed the secret data. Inverse transform is applied to the output after embedding the data. Capacity of the cover image is the maximum size of the secret data which can be embedded. Cover image has higher capacity to embed the secret data when transformations are applied rather than the simple LSB spatial domain method.

**Using DCT**: Image is divided into blocks, each block consists of 8x8 pixels. DCT is applied to each block which results in 64 DCT coefficients. The least significant bits of the DCT coefficients are used to embed the secret data. As the coefficients are in the frequency domain, changes will be imperceptible [36].

**Using FFT**: After applying the fourier transform to the image, the most significant coefficients are selected and used to hide the secret data. These significant coefficients contain the important data. Instead of embedding the secret data in the least significant bits, which can be lost by compressing the stego image. Most significant bits will not be lost in compression as it contains the important information of the image [36].

**Using DWT**: Image is subjected to wavelet transform and the least significant bits of coefficients of the transformation are used to embed the secret data. By embedding the secret data there will be change in the intensity of the image, but will not be visible to the human eye as the data is embedded in the wavelet coefficients [36].

**Spread Spectrum Image Steganography (SSIS)**

The secret data to be embedded into the cover image is encrypted using a key. The encrypted secret data is modulated so that it looks like Gaussian noise and then it is added to an image which is embedded into the cover image. The noise which is added to the secret data is the image acquisition noise. The resultant image, which is the stego image, has high SNR as the power of the embedded secret message is low when compared to the power of the cover image. Therefore the noise embedded is not visible, and the chances of the interceptor detecting is very low. The extracted data will appear as noise to an interceptor, but a designated receiver with a correct key and algorithm which was used to embed can retrieve the secret data. If a stego key is used for embedding the data, even that key is needed for extracting the secret data accurately. This method has more security when compared to other methods as it uses both encryption and Steganography techniques together. But the disadvantage with this method is that it is vulnerable to image processing attacks and compressions of the stego image [36].

## 4.5 Evaluation

### 4.5.1 BAAI: Biometric Authentication and Authorization Infrastructure [37]

Many applications now-a-days require authorization and authentication together. Authorization is ensured by using an attribute certificate which has the information about the user's privileges. Authentication of a user is achieved by using an identity certificate which has the identity of the user. Secret key system was used earlier, but this does not provide much security. Physical presence of the user is necessary, which is solved by using biometric systems. By combining the attribute and identity certificate, Authorization and Authentication Infrastructure (AAI) has been developed. By incorporating biometrics into AAI's, led to Biometric Authentication and Authorization Infrastructure (BAAI).

With the increase in the use of internet for communication purpose, the authorization should be shifted from centralized systems to distributed systems. In order to achieve authorization in the distributed systems, authentication is also needed along with authorization. This is implemented by using BAAI.

The BAAI system uses steganography and face recognition techniques. Steganography is used to bind two objects instead of hiding one object in another. There is no need to hide an object as all the data is known to the public. The purpose of using steganography is that the interceptor will not be able to get any part of the information that he/she needs. The method used to implement steganography is the least significant bit (LSB) insertion, as it does not affect the intensities of the pixels. So it is imperceptible to the human eye. Face recognition system is developed by taking the biometric signature of the individual by using a sensor. Features are extracted and are stored in a matrix [44]. At the time of matching, this matrix is compared with all the matrices in the database. Any algorithm can be used to obtain the feature matrix.

**Method Used**

The image of the user's face is obtained and hash function is applied to the image's most significant bits. This is stored in an Object field and the name of the user is stored in Name field. This completes the identity certificate. The authorization authority prepares the attribute certificate with all the privileges to that user. The attribute certificate is binded with the user's hash image file i.e the identity certificate using steganography technique explained above. Steganography helps in such a way that the attribute certificate will not interfere at the time of identification/recognition by the biometric device. After binding, the final output is called Visual Attribute Certificate (VAC) which has authentication and authorization in one single object.

**Authentication and Authorization process**

The sensor obtains the face image of the user, the system performs identification between the obtained face image and the VAC stored in the database. If there is a match, the system extracts the attribute certificate and the privileges of the user are accessed.

### 4.5.2 Method Studied Practically

The sensitive data is embedded in a cover image and the obtained stego image is transmitted through an unsecured communication channel. At the receiving end, the sensitive data is recovered from the stego image and the image is discarded. In this case, the first method deals with embedding text and the second method is related to hiding an image in another image.

#### 4.5.2.1 Method – 1 (Embedding Text [38, 39])

The sensitive text such as the credit card information (i.e., name, card number, expiration date) is embedded in an unsuspicious cover image. This method can be used in online transactions and online shopping. The text to be embedded is converted to binary format and is embedded only if the size of the binary data is less than that of the size of the cover image. If the size of the binary text is 'n', then it is embedded in the first 'n' positions in the cover image taking the $b^{th}$ bit plane of the cover image where 'b' can be varied. The cover image is reshaped into a row vector I. The value at I(i) is adjusted to

$$\mathbf{Mod\ (\ I(i),\ 2^{b}) - 2^{(b-1)}\ >= 0}$$

by subtracting $2^{(b-1)}$ from I (i) if the above condition is true. Binary bits are added to the obtained values of I. At the recovering end, the stego image is reshaped to a row vector and the first 'n' values are taken for decoding. The following condition is implemented to recover the binary data,

$$\mathbf{Remainder(\ I\ (i),\ 2^{b}) > 2^{(b-1)}}$$

If the above condition is true, then the bit that was embedded is encoded as '1', else '0'.

After decoding, the obtained binary stream is reshaped and converted to characters. This method is not robust against any signal processing attacks as any change in the first 'n' pixels would destroy the embedded text. Also if a single bit is decoded incorrectly, the meaning of the whole text is changed.

#### 4.5.2.2 Method – 2 (Embedding an Image [40])

In this method, an image is hidden in another image. This can be useful for multimodal biometrics in ideal conditions. An iris image is hidden in a face image and the obtained stego image can be stored or transmitted. From the stego image, the hidden iris image can be extracted and used for authentication along with the face. As explained in the above section, about the least significant bit technique, the number of least significant bits considered for embedding can be varied up to 7 bits in an 8-bit image. (that means 7 bits out of 8 can be replaced by the most significant bits of the secret image to embedded the secret data). As the number of least significant bits used increases, traces of iris image can be seen in the stego image. So 4 or 5 least significant bits can be used for embedding without being visible to the human eye. At the extraction stage, the least significant bits are detected and are used to recover the iris image. But we also need to have the knowledge of the number of least significant bits used for embedding. This method is not robust against any signal processing attacks, as the pixel value is directly substituted in the cover image.

## 4.6 Results

Two methods have been implemented, where both the methods belong to spatial domain steganography. In the first method, plain text is embedded directly and in the second method, an image is used as a watermark. The cover image and the watermark image are gray scaled image, in the second method. The robustness of these techniques has been tested by applying Gaussian noise, random noise, degrading the quality by compressing the stego image.

MSE is an error metric which has been used to compare the extracted watermark image with the original watermark image. Mean square error is the average of the square of the difference between the original watermark image and the extracted watermark image. The lower the MSE value, the better it is.

$$MSE = \frac{1}{M * N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left( I(i,j) - I_W(i,j) \right)^2 ,$$

where I is the original image,

$I_W$ is the watermarked image and

M, N are the dimensions of the image (same for both the images as size of the stego image is same as the original image).

PSNR is another error metric which is based on MSE. The PSNR has been calculated between the original cover image and the stego image.

$$PSNR = 20\log_{10}\left(\frac{255}{\sqrt{MSE}}\right)$$

255 is taken because in our iris image, each pixel is represented using 8 bits, so ($2^8$-1). As MSE is inversely proportional to PSNR, higher the PSNR value, the better it is, which means the ratio of signal to noise is higher. PSNR does not take human visual system into account.

The PSNR has been calculated between the original cover image and the stego image and MSE has been estimated between the original watermark image and the recovered watermark image, only for the second method. Because in the first method, MSE cannot be calculated for text as text is embedded and extracted.

Random noise is a disturbance added to the stego image. A matrix of size of stego image whose entries are uniformly distributed random numbers in the interval [0, 1] is added to the stego image. Before adding, the matrix is multiplied with a multiplier to increase the magnitude of disturbance added to the stego image. Gaussian noise follows normal distribution with constant mean and variance.

The quality of the image is degraded by using JPEG compression [44]. The lesser the quality value, the higher the degradation of the image due to JPEG compression. The compression ratio will be mentioned in the results. Compression ratio is calculated using the following formula [46]:

$$Compression\_ratio = \frac{M * N * bit\_depth}{8 * file\_size},$$

where M and N are the dimensions of the compressed image,

bit depth is the number of bits required to represent each pixel and

file size is the space occupied by the image in bytes.

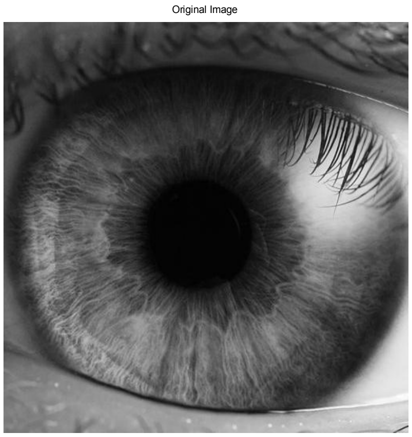
By compressing an image, the quality of that image degrades.

Stego image is rotated in counter-clockwise direction with the specified angle in degrees. In nearest neighbor interpolation, pixel in the new image is the nearest pixel in the original image. Bilinear interpolation, the value of the new pixel is calculated by the weighted average of the 4 pixels in the nearest 2 x 2 neighborhood of the pixel in the original image. In bicubic interpolation, the new pixel is the bicubic function using 16 pixels in the nearest 4 x 4 neighborhood of the pixel in the original image.

### 4.6.1 Method – I

**Embedding Text:**

A gray-scale iris image of size 512 x 512 is selected as a cover image, and text shown on the right panel in Figure 4.6 is embedded. The stego image with embedded text and the perfectly recovered text are shown in Figure 4.7.



**Mark 1234567823456789 09/09**

Original Image (512x512) Text

Figure 4.6: The left panel shows the cover image used in our experiments. The right panel displays the sensitive text which is embedded.

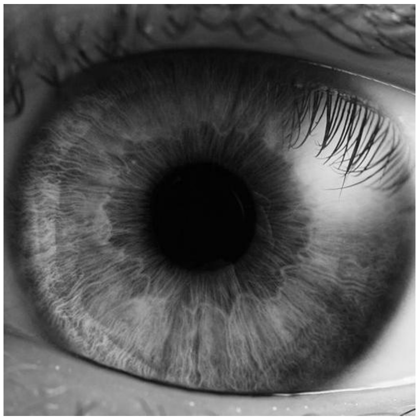
The sensitive text is converted to binary, reshaped to a row matrix and is embedded in the first ‘n’ bits in the cover image. The value at I(i) in the original or cover image is adjusted to

$$\mathbf{Mod\ (I(i),\ 2^{b}) – 2^{(b-1)} >= 0}$$

by subtracting $2^{(b-1)}$ from I (i) if the above condition is true. ‘b’ is the number of planes used, 1 in this case as gray scaled image is used. Binary bits are directly added to the obtained values of I. At the receiving end, the stego image is reshaped to a row vector and the first ‘n’ values are taken for decoding. The following condition is implemented to recover the binary data,

$$\mathbf{Remainder(\ I\ (i),\ 2^{b}) >= 2^{(b-1)}}$$

If the above condition is true, then the bit that was embedded is encoded as ‘1’, else ‘0’.

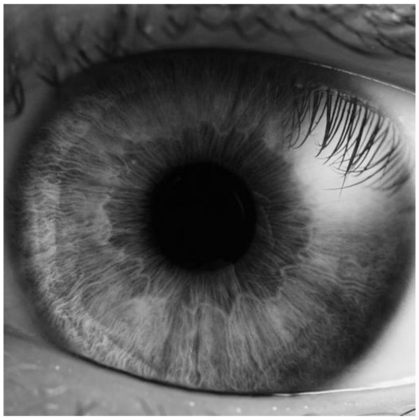


**Mark 1234567823456789 09/09**

Stego Image Recovered Text

Figure 4.7: The left panel shows the stego image. The right panel displays the perfectly recovered sensitive text.

When the stego image is attacked by Gaussian noise, the text which is embedded is destroyed completely and can not be recovered. An example of the recovered text is shown in Figure 4.8 (right panel).



**£< □;□I□¶□È□V²ûdf**

Stego Image with Gaussian noise
Mean = 0.1 and Variance = 0.001

Recovered Text

Figure 4.8: The left panel shows the stego image with Gaussian noise. The right panel displays the destroyed sensitive text.

When the quality of the stego image is degraded by compressing the stego image, the watermark can be partly recovered. Table 4.1 summarizes the results for compressing the stego image.

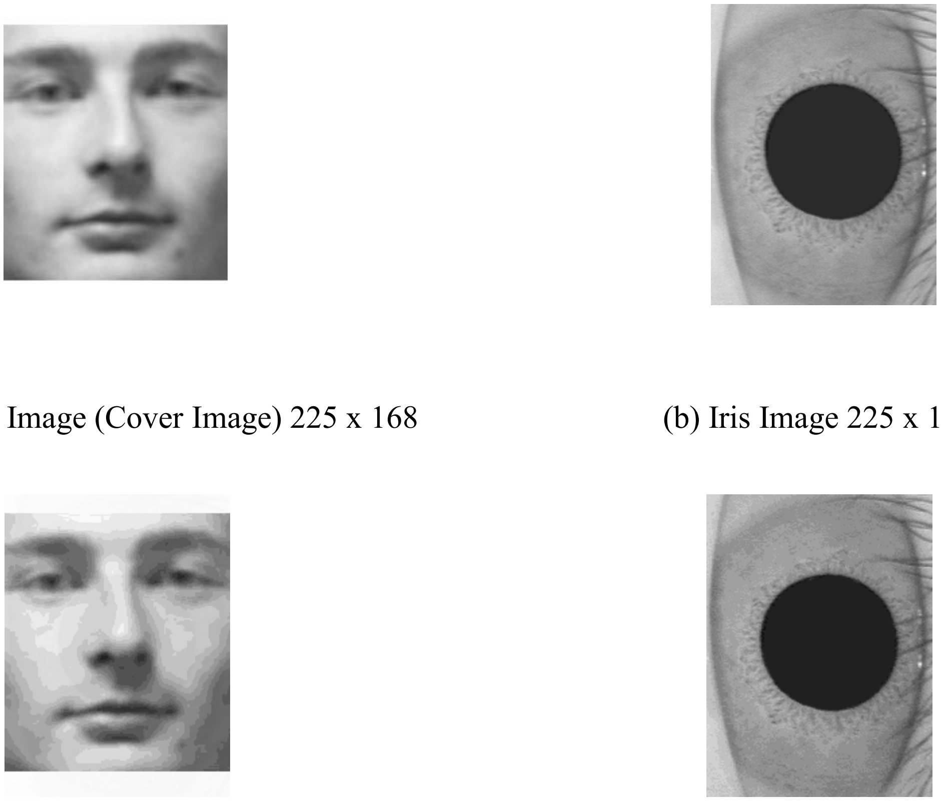
Table 4.1: Quality degradation applied to stego image by compression the image.

| Quality | Compression ratio | Number of characters recovered |
|---|---|---|
| 99 | 3.3331:1 | 12 |
| 98 | 3.8565:1 | 3 |
| 97 | 3.9762:1 | 3 |
| 96 | 4.2334:1 | 3 |
| 95 | 4.4801:1 | 1 |

### 4.6.2 Method – II

**Embedding Image:**

A gray scale face image is used as a cover image and a gray scale iris image is embedded into the cover image. Both the images are of size 225 x 168. Figure 4.9 shows the cover image, secret image, stego image and the extracted secret image. In the following table 4.2, 'N' is used to denote the number of least significant bits used for embedding the iris image. Here N = 4 is taken in Figure 4.9 and the 4 least significant in the face image are replaced by the 4 most significant bits in the iris image. At the time of recovery, as we have the knowledge of the number of least significant bits replaced by the most significant bits, those bits are used to recover the hidden iris image. As the changes in the least significant bits are not visible, they are replaced by one of the most significant bit repeatedly after removing the most significant bits of the iris image. Iris image is also reconstructed in the similar manner.



(a) Face Image (Cover Image) 225 x 168

(b) Iris Image 225 x 168

(c) Stego Image with iris embedded

N = 4

PSNR = 1.1139

(d) Recovered Iris Image

MSE = 81.5223

Figure 4.9: (a) Cover image used in this experiment, (b) iris image which is hidden in the cover (face) image, (c) stego image with iris embedded in face and (d) extracted iris image.

Table 4.2: For different 'N' values

| N value | MSE | PSNR |
|---|---|---|
| 4 | 81.5223 | 1.1139 |
| 5 | 15.0071 | 0.6021 |
| 6 | 3.6505 | 0.4771 |
| 7 | 0.4071 | 0.3010 |

For N = 6 and N = 7, Stego image gives a clue that another image has been hidden. For the remaining test cases, N = 5 has been considered.

Rotation is applied to the stego image (See Figure 4.10), and the results shown in Table 4.3, show that it is robust to some extent. The MSE is estimated for the original iris image and extracted iris image which is rotated.

Stego Image after applying rotation

Rotated by -2 degrees

Recovered Iris image

MSE = 54.5543

Figure 4.10: The left panel shows the rotated stego image. The right panel displays the recovered iris image.

Table 4.3: When stego image is rotated through an angle using different interpolation methods.

| Type | Angle | MSE |
|---|---|---|
| Nearest | -2 | 54.5543 |
| Bilinear | -2 | 63.6403 |
| Bicubic | -2 | 70.2239 |

When Gaussian noise is applied, the embedded iris could not resist and gets affected by the noise as shown in Figure 4.11. Table 4.4 summarizes the results when the stego image is affected with Gaussian noise.

Stego Image with Gaussian noise
Mean = 0.1, Variance = 0.001

Recovered Iris image
MSE = 114.9202

Figure 4.11: The left panel shows the stego image with Gaussian noise. The right panel displays the destroyed iris image.

Table 4.4: Stego image is affected with Gaussian noise

| Mean | Variance | MSE | PSNR |
|---|---|---|---|
| 0.1 | 0.001 | 114.9202 | 2.2945 |
| 0.1 | 0.005 | 91.3920 | 1.2553 |

When random noise is applied, the embedded iris can be recovered moderately (See Figure 4.12), but by increasing the value of the multiplier for the random noise, the iris image gets destroyed. Table 4.5 presents the results when stego image is added with random noise.

Stego Image with Random noise
Multiplier = 5

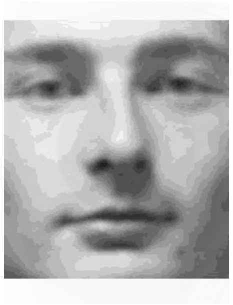
Recovered Iris image
MSE = 1.6146

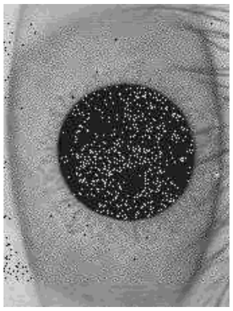
Figure 4.12: The left panel shows the stego image with Random noise. The right panel displays the recovered iris image.

Table 4.5: Random Noise applied to Stego Image

| Multiplier | MSE | PSNR |
|---|---|---|
| 5 | 1.6146 | 0.7782 |
| 10 | 4.6395 | 0.9031 |
| 15 | 26.6620 | 1.0414 |
| 20 | 54.7327 | 1.1461 |

Quality of the watermarked image is degraded by compressing the stego image to estimate the robustness of the technique. We show that the technique does not withstand the quality degradation.





Stego Image with Quality degradation                                   Recovered Iris image

Quality = 95 or compression ratio 3.1214:1                             MSE = 106.1663

Figure 4.13: The left panel shows the stego image with quality degradation or with compression ratio 3.1214:1. The right panel displays the recovered iris image.

Table 4.6: Compression or quality degradation is applied to Stego Image

| Quality | Compression ratio | MSE | PSNR |
|---|---|---|---|
| 99 | 1.9259:1 | 90.7032 | 1.2435 |
| 97 | 2.5005:1 | 105.2480 | 1.8063 |
| 95 | 3.1214:1 | 106.1663 | 2.0156 |

# Chapter 5

## Encryption, Watermarking and Steganography

### 5.1 Why this is used? [42]

Biometrics is widely used, as it has the ability to differentiate the authorized user from an unauthorized user. But the problem arises in ensuring the security of the biometric data. If the biometric data of a user is compromised or stolen by an interceptor, it can not be replaced by another biometric data. To provide security to the biometric data, encryption, watermarking and steganography techniques can be used.

Encryption makes the sensitive information meaningless to unauthorized parties. Steganography is used to hide sensitive data in an unexpected carrier/file and it also hides the communication itself. The probability of an interceptor knowing that the sensitive data is hidden in a cover file is very low. Watermarking embeds the proprietary information about the file, i.e., watermark into the file protecting the rights of the data in the file.

When encryption is used for protecting the biometric data, the encrypted biometric templates are saved in the database. At the time of identification/verification, the stored encrypted biometric template is decrypted and compared with the biometric template obtained online to generate the matching score. But the problem with this technique is that the biometric data becomes unsecured once it is decrypted. Watermarking embeds the sensitive data into the cover or host file. It is not related to encryption or decryption, thus using watermarking provides one more level of security even after decrypting the data. Even if an interceptor changes the file, it can be detected by using watermarking techniques. Encryption and watermarking can be applied together to increase the security of the biometric data. The sensitive data can be first encrypted and then embedded into the host file. Encryption can also be used along with steganography in a similar manner to increase the security of the biometric data.

## 5.2 Evaluation

### 5.2.1 Securing Online Shopping Using Biometric Personal Authentication and Steganography [41]

Company websites which encourage online shopping should give equal importance to easy access to their website for the customer as well as security to their website and also to customer's data. Data security can be ensured by providing confidentiality, integrity, availability and accountability.

**Confidentiality:** The content of the sent file is only seen by the authorized receiver and no interceptor has accessed or seen that file. This ensures confidentiality.

**Integrity:** The sent file has not been changed in its transition through unsecured communication channel. Integrity of the data is achieved if the data has not been altered or destroyed.

**Availability:** The file and the system are available to the authorized parties at any time in a specified interval.

**Accountability:** the accountability of a system is ensured, if the receiver is sure about the sender as the one who claims to be.

To achieve data security, biometrics is used. So the online shopping system uses fingerprint biometrics along with encryption and steganography to complete a secured transaction.

The system comprises of fingerprint verification for authentication, the sensitive card information is encrypted and hidden in the Electronic Internet Shopping Card (EISC) along with the extracted fingerprint minutiae using fragile steganographic technique which uses a stego key.

The system has three stages:

1) EISC creation stage

2) Using EISC in online shopping stage and

3) EISC validation stage

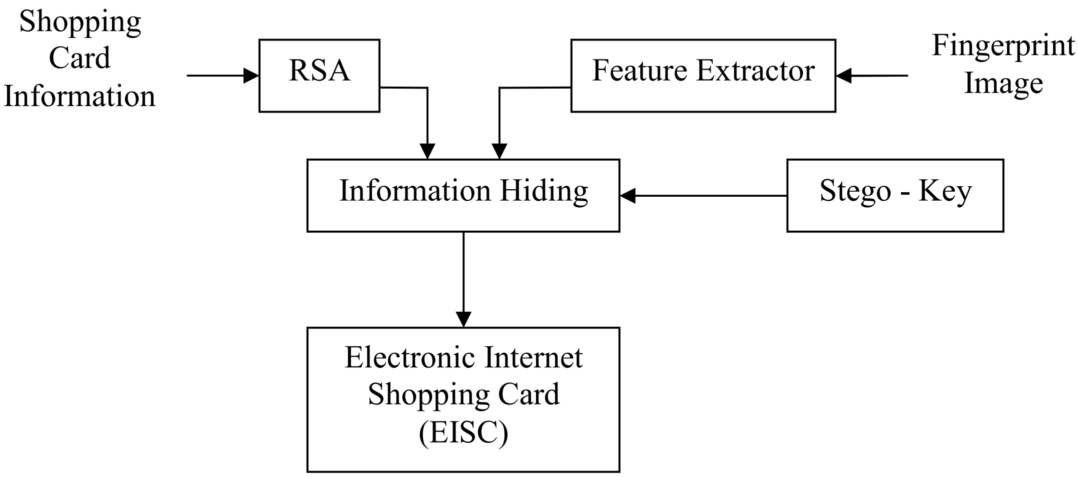


Figure 5.1: Method followed to create EISC [41]

**EISC Creation Stage**

At this stage, when the customer orders for an online shopping card, he/she is asked to visit the card issuing place. At this place, the customer's fingerprint image is collected and minutiae set is extracted from the customer's fingerprint. Then the customer is given card number, expiration date which is sensitive along with card serial number. The sensitive data is encrypted using RSA algorithm. The card serial number identifies both the customer as well as the issuer. A storage media is used to store the raw EISC image and the special software. The raw EISC image contains the encrypted sensitive data, card serial number and customer's fingerprint minutiae. All these pieces of information are converted to binary data and then hidden in an EISC image (See Figure 5.1). This binary data is embedded using a stego-key which will be used by the special software to extract the fingerprint minutiae set from the EISC image at the time of authentication for generating a valid EISC image from the raw EISC image. The valid EISC image is created by adding a verification tag which is related to the amount of the transaction and also serial number of the transaction which is added to the raw EISC image.

**Using EISC in Online Shopping Stage**

When the customer wants to shop online, after he/she decides about the product to be purchased, he/she will be asked to give the credit card number and the expiration date or to upload the valid EISC image in order to purchase the product. The customer has to use the special software to generate the valid EISC image. The special software requires to authenticate the customer which is done by presenting his/her fingerprint. The software compares the features extracted from the given fingerprint image, with the minutiae set extracted from the raw EISC image using a stego-key. If both of them match, then the software prompts the customer to enter the amount of transaction, using which a verification tag is generated. This verification tag is converted to binary stream and is added to the raw EISC image to create a valid EISC image. The customer uploads this image to the web site.

**EISC Validation Stage**

When the web site merchant receives the image uploaded by the customer, it is forwarded to the card issuer using a secured connection. The card issuer information is known by the card serial number. After sending the EISC image to the issuer, the web site merchant waits for the validation results.

The issuer's software extracts the hidden information i.e., the card number, expiration date, the fingerprint minutiae, card serial number and the amount of transaction from the received EISC image. Customer's information is retrieved from the database by using the card serial number which identifies the customer. The software compares both the pieces of information, checks whether the card is valid, whether the minutiae extracted matches with the one stored in the database, whether the other information matches i.e., the card number, expiration date and the serial number, and the transaction amount is within the limits. If all the above conditions are correct then the software sends a positive response to the merchant which will complete the transaction else it sends a negative response to the merchant, who will cancel the order.

If the software is stolen, no one else except the authorized customer can generate the valid EISC image as the software needs authentication from the customer to generate EISC image. Attacking the merchant's web site and obtaining the EISC image by the hacker will be of no use because each image has unique verification tag which is generated by the customer's special software. As the customer has the storage media containing the raw EISC image and special software, he can do online shopping from any computer. Any slight tampering to the image will cancel the whole transaction. As all the four security issues, confidentiality, integrity, availability and authentication are met, data security is guaranteed.

### 5.2.2 Hiding Biometric Data [42]

Two scenarios are considered in this paper. The first scenario implements steganography and encryption together and the second scenario implements watermarking and encryption together. The sensitive data which is embedded are the fingerprint and the face coefficients and the cover file in which sensitive data is embedded can be a fingerprint image or a face image or an arbitrary image. The method used for hiding the sensitive data is same in both the scenarios.

**Scenario - 1 (Steganography and Encryption)**

In this scenario (See Figure 5.2), the fingerprint minutiae can be hidden in the fingerprint image or an arbitrary image. The host image in which data is hidden is not related to the hidden data. The function of the host image is to carry the hidden information, once the data is extracted from the host image, the host image is discarded. If a fingerprint image is used as a host image, the overall security of the system is increased. If the fingerprint minutiae is hidden in the fingerprint image and transmitted, the interceptor who gets the fingerprint image treats the image as original biometric data. The security of the transmission can be increased to another level by encrypting the stego image before transmitting.

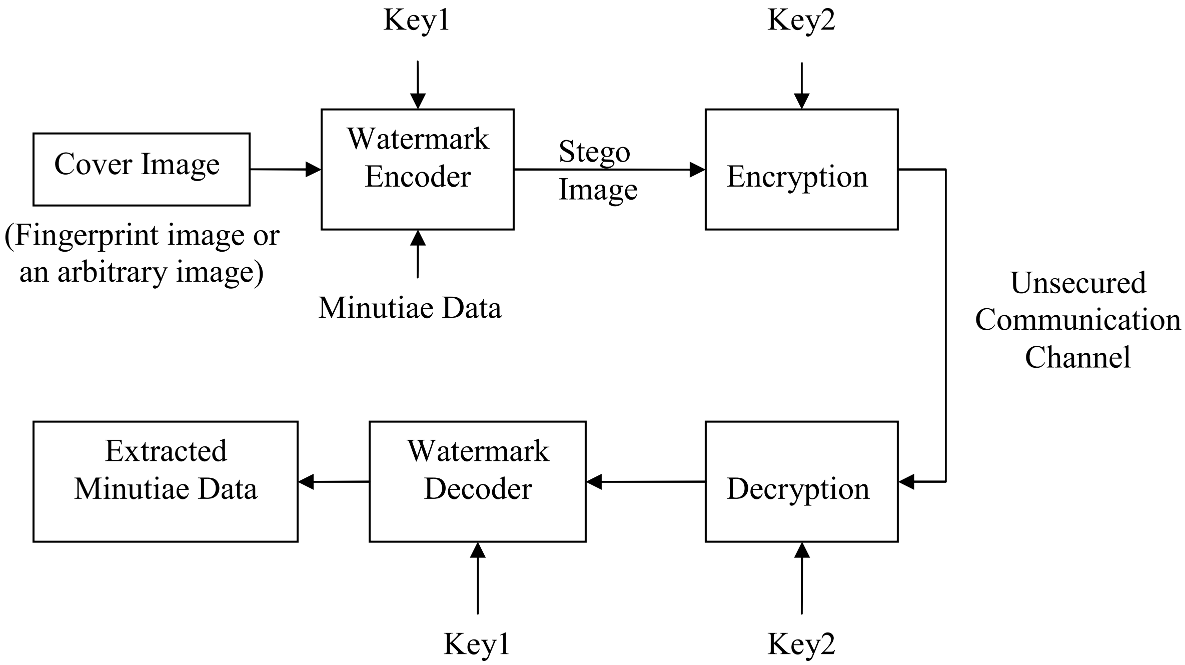


Figure 5.2: Scenario – 1 [42]

Symmetric or asymmetric key encryption can be used depending on the application. Symmetric key encryption is faster than asymmetric key encryption but needs more number of keys for communicating between many parties.

The fingerprint image which is obtained by the sensor is used to extract features. These features are a comprise of the minutiae points which are represented by the position and orientation. This data is hidden in some other fingerprint image using a key. The obtained image is the stego image which is encrypted using an encryption key to increase the security of the data. At the receiving end, the file is decrypted using a key which was used for encryption in case of symmetric key encryption or related key in case of asymmetric key encryption. The embedded data is extracted using the key used for embedding.

**Scenario – 2 (Watermarking and Encryption)**

In this scenario (See Figure 5.3), the face coefficients are hidden in the fingerprint image of the same user. The host image is related to the hidden data, the host and the hidden data belong to the same user. In this method, one biometric data is used to hide another biometric data to increase the security of the data. At the time of identification/verification, the fingerprint image is sensed and compared with the fingerprint image stored in the database. The face information is extracted from the fingerprint image stored in the database. This face information is used as another source for authentication. Multi-modal biometrics is implemented in this scenario.

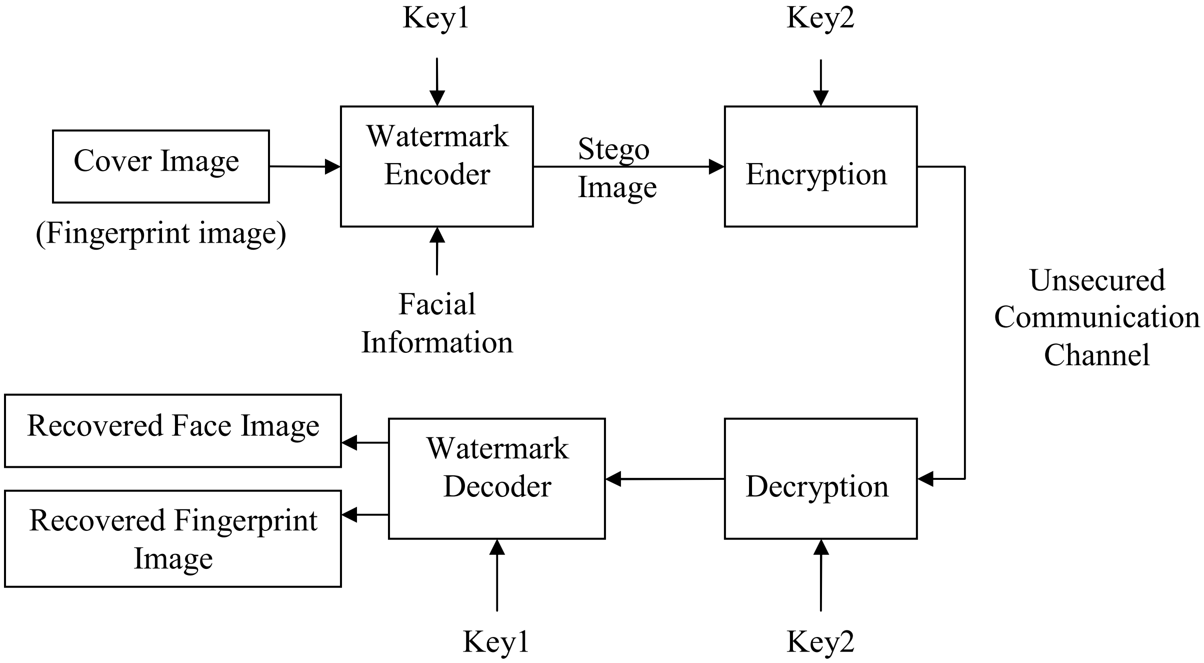


Figure 5.3: Scenario – 2 [42]

**Data Hiding Method**

**Embedding**

The data to be embedded is converted to binary form. In the first scenario, the fingerprint minutiae is converted to binary, i.e., every field, the position in x-axis and y-axis and the orientation is converted to 9-bit binary representation. In the second scenario, the face coefficients are converted to binary, each coefficient using 4 bytes of binary data. The secret key is used as a seed for the random number generator to generate the locations in the host image where data will be embedded. The random number generator, generates a sequence of random numbers in the interval [0, 1]. Every location of the pixel in the host image which will be marked consists of one number with odd indices and one number with even indices from the sequence of random numbers.

In scenario-2, the host image is used as a source for authentication, the significance map is generated which has ‘0’ if the current pixel belongs to the minutiae set or ‘1’ if it does not belong to the minutiae set. So the significant pixels (where significance map has ‘0’) in the host image are not changed while embedding the data, which leads to good authentication performance. But in scenario-1, the significance map, may or may not be considered as the host image is not used for authentication.

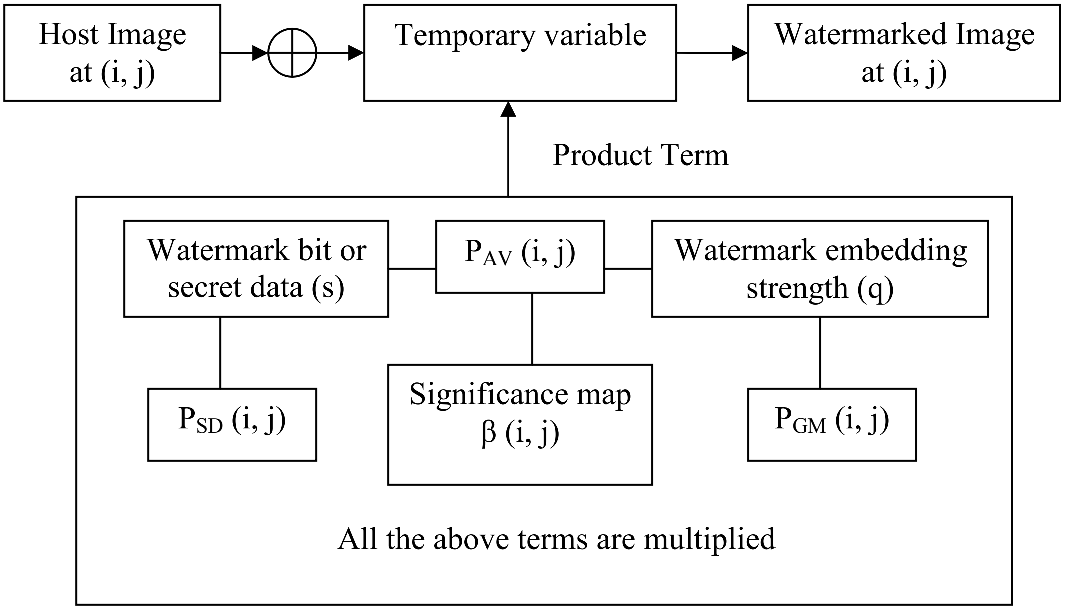

Host Image at (i, j)
Temporary variable
Watermarked Image at (i, j)
Product Term
Watermark bit or secret data (s)
$P_{AV}$ (i, j)
Watermark embedding strength (q)
$P_{SD}$ (i, j)
Significance map β (i, j)
$P_{GM}$ (i, j)
All the above terms are multiplied

Figure 5.4: Procedure followed for embedding secret data

Figure 5.4 shows the procedure for embedding the data. In this figure,
$P_{AV}$ is the average of the pixel values in a 5 x 5 square neighborhood,
$P_{SD}$ is the standard deviation of the pixel values in a 5 x 5 cross-shaped neighborhood,
$P_{GM}$ is the gradient magnitude calculated using 3 x 3 sobel operator.

All these terms are considered to increase the magnitude of the watermark in those areas of the image where increase in the magnitude does not become visible to the human eye.

β (i, j) takes the value '0', if the current pixel which is being considered in the host image is a minutiae pixel and the temporary variable is zeroed. That pixel is unchanged. The next pixel location is considered. When β (i, j) is '1', that pixel in the host image is marked with the data, else the next pixel location is considered.

Each and every watermark bit is embedded several times depending on the size of the image for correct decoding. Along with the watermark bits, '0' and '1' are also embedded as reference bits, which are useful for calculating the adaptive threshold, which is useful for extracting the watermarked bits. The obtained file is encrypted using a suitable encrypting algorithm using a key.

**Extracting**

After receiving the file, it is decrypted using the same or appropriate key used for encryption. The locations where the data is embedded is detected by using the same key as in the embedding procedure as a seed for the random number generator. At each location, an estimate Â is calculated using the linear combination of pixel values in a 5 x 5 cross shaped neighborhood of the pixels in the watermarked image. Difference between this estimate and the actual watermarked pixel at that location is calculated.

$$\boldsymbol{\delta = P_{WM}\,(i, j) - \hat{A}\,(i, j)}$$

This difference is calculated for all the embedding locations and is averaged ($\delta_{AVG}$). The average is calculated for the reference bits ‘0’ and ‘1’ in the same manner to obtain the adaptive threshold, $\delta_{AVG}(R_0)$ and $\delta_{AVG}(R_1)$ for bit ‘0’ and ‘1’.

If $\delta_{AVG}$ is close to $\delta_{AVG}(R_0)$, then the watermarked bit is decoded as ‘0’ and if $\delta_{AVG}$ is close to $\delta_{AVG}(R_1)$, then it is decoded as ‘1’. The decoding procedure is explained in Figure 5.5.

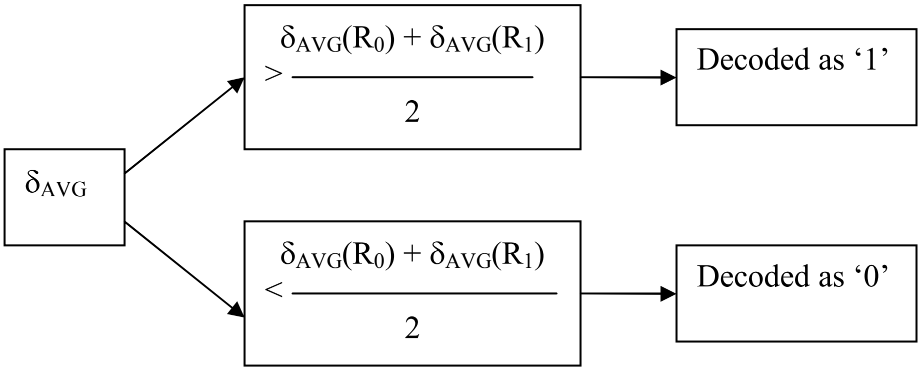

$\delta_{AVG}(R_0) + \delta_{AVG}(R_1)$
$> \overline{\qquad\qquad}$
2
Decoded as ‘1’
$\delta_{AVG}$
$\delta_{AVG}(R_0) + \delta_{AVG}(R_1)$
$< \overline{\qquad\qquad}$
2
Decoded as ‘0’

Figure 5.5: Condition for extracting the watermark

After the decoding procedure, the hidden data which is extracted is used for authentication. In scenario-1, the minutiae can be used directly for authentication and in scenario-2, the extracted face coefficients are used to construct the face image which is used for authentication. In scenario-1, the cover image is discarded, but in scenario-2, an estimate of the host image is formed by replacing the watermarked bits with the estimate Â at that location which is calculated as a linear combination of the pixel values in 5 x 5 cross-shaped neighborhood of the watermarked pixel. This obtained host image estimation is used for authentication.

### 5.2.3 Data Hiding for Multimodal Biometric Recognition [43]

A multimodal biometric system is implemented which uses fingerprint, iris and voice pattern of a user for authentication. The speech signal in time domain and the DWT of the iris image are hidden in the wavelet coefficients of the fingerprint image of the user. After embedding the speech and iris patterns, the fingerprint image is compressed to further reduce the memory occupied by that image. The file which has important information about the hidden places in the fingerprint image is encrypted using a key to increase the level of security. This solves the storage capacity and also security problems. At the receiving end, the obtained file is first decrypted and the obtained image is decompressed. The decompressed fingerprint image is used to extract the embedded iris and speech patterns by using the information in the file, which are used for authentication along with the fingerprint image. The obtained file can be decrypted by the key which was used for encrypting. This key is generated using the fingerprint of the user.

**Data Hiding Method**

Gray scaled fingerprint and iris images are used. The fingerprint image is subjected to N-level DWT. Each level has LL, LH, HL and HH sub-bands. Any changes to the low frequency components are visible to a human eye. So LL sub-band is not used, instead all the other three sub-bands of level N, N-1, ..2 are used for hiding the data. The remaining sub-bands have to be divided into 2 x 2 non-overlapping blocks where speech and iris patterns can be substituted directly into these blocks. In order to divide them into blocks, energy of each block is calculated using the formula,

$$\hat{E}_i = \frac{1}{(N_{b1} * N_{b2})} * \sum_{k \in b_i} | w_i(k) |^2 , i = 1,2....B$$

where $w_i$ (k) is the wavelet coefficient of the block $b_i$,

B is the number of blocks to be divided into,

$N_{b1}$, $N_{b2}$ are the block dimensions (2 x 2).

After calculating energy of each block, the blocks are arranged in increasing order of the energy. The blocks having energy greater than a threshold (5, in this paper) are considered and replaced by the data. $N_S$ is the number of blocks required to carry the speech data and $N_I$ is the number of blocks required to carry the iris data. $N_S$ + $N_I$ blocks whose energy is greater than a threshold are selected and their coefficients are made zeros and are used to substitute the speech and iris data.

Significance map is created by assigning block as ‘1’, if that block hides data or a ‘0’, otherwise.

Speech waveform is divided into 4 samples and these samples are substituted in the first $N_S$ blocks from level 2, 3,…..N. Iris image is subjected to 1-level DWT and is divided into 4 blocks which is embedded into the next $N_I$ blocks.

The embedded fingerprint image is obtained by applying inverse DWT, which is subjected to JPEG lossy compression to reduce the storage. This final compressed fingerprint image is used for storage or transmission along with the significance map. The number of blocks used by the iris and speech, $N_I$ and $N_S$ are also required at the time of extraction. The significance map along with $N_I$ and $N_S$ are encrypted with a key which is coded using the user’s finger pattern.

**Data Extraction Method**

At the extraction stage, the fingerprint of the user is taken and is used to decode the key. This key is used to decrypt the significance map, $N_S$ and $N_I$. The compressed file is decompressed and then decomposed using a N-level DWT. The obtained sub-bands are divided into 2 x 2 blocks. Using the significance map and $N_S$ and $N_I$ value, by scanning the sub-bands, an estimate of the speech pattern in time domain is obtained. Later the iris pattern estimate is obtained by applying IDWT to the extracted $N_I$ wavelet coefficients. Again by using the significance map, where the block has ‘1’, the same block is zeroed in the fingerprint image i.e., where the speech and iris patterns are extracted, there the fingerprint image is zeroed. The estimated fingerprint image, speech and iris patterns are used for authentication. This implementation takes less storage for storing three biometrics (two biometrics are hidden in one biometric and then compressed) when compared to storage capacity required for storing three independently compressed biometric templates.

## 5.3 Conclusions and Future Work

A survey has been conducted on the techniques: encryption, watermarking and steganography to provide protection, privacy and security to the biometric data. For the implemented watermarking techniques, results show that transform domain technique is more robust than spatial domain technique and also in transform domain technique, the quality of the reconstructed watermark depends on the size of the watermark in spite of the high PSNR value as PSNR value does not take human visual system into consideration

The spatial domain techniques are more prone to signal processing attacks than frequency or transform domain techniques because the watermark is embedded directly into the pixel. When the images is disturbed by noise or any other attack, the pixel value changes which results in destroying the embedded watermark. In transform domain, the watermark is added to the transformed coefficients so any changes in the pixels will have small changes in the transform coefficients.

For the implemented spatial domain steganographic techniques, the evaluation to the robustness showed that slight changes to the stego image, will destroy the whole sensitive data hidden.

The security for the biometric data against any attacks in the biometric system cab be increased by using two techniques such as encryption and watermarking or encryption and steganography. The sensitive data can be encrypted first by using a secret key and then watermarked or hidden in another image. This way even if an interceptor breaks the watermarking or steganography by extracting the hidden data, he/she will not understand the secret data as it is in an encrypted form which is meaningless to an interceptor.

Based on the results of the study, this project can be extended by developing watermarking technique which is robust to large watermarks to improve the performance. As the size of the watermark increases, the robustness of the watermarked image to signal processing attacks decreases. Frequency domain steganographic techniques can be developed which are more robust than the spatial domain techniques.

**References:**

[1] A. K. Jain, A. Ross, S. Prabhakar, "An Introduction to Biometric Recognition", IEEE Transactions on circuits and systems for video technology, Vol. 14, No. 1, pp: 4 – 20, January 2004.

[2] B. Schneier, "The Uses and Abuses of Biometrics", Communications of the ACM, Vol. 42, No. 8, pp: 136, August 1999.

[3] N. K. Ratha, J. H. Connel and R. M. Bolle, "An Analysis of Minutiae Matching Strength", Proc. Third International. Conf. Audio and Video Based Biometric Person Authentication, pp: 223 – 228, June 2001

[4] Ibrahim A. Al-Kadi, "The origins of cryptology: The Arab contributions", Cryptologia, Vol. 16, No. 2, pp: 97 – 126, April 1992.

[5] C. P. Pfleeger, S. L. Fleeger, "Security in Computing", Prentice Hall, Fourth Edition, 2007.

[6] http://en.wikipedia.org/wiki

[7] C. Brooks, "Computers and Society, Applications of Encryption", Department of Computer Science, University of San Francisco.

[8] B. Schneier, "Applied Cryptography: Protocols, Algorithms and Source Code in C", John Wiley & Sons, New York, Second Edition, 1996.

[9] D. Salomon, "Data Privacy and Security: Encryption and Information Hiding", Springer, 1st Edition, May 2003.

[10] W. Stallings, "Cryptography and Network Security: Principles and Practices", Prentice, New York, Third Edition, 2003.

[11] D. G. Messerschmitt, "Understand Networked Applications: A First Course", Morgan Kaufmann, 1999.

[12] A. Stockel, "Securing Data and Financial Transactions", IEEE Proceedings on 29th Annual International Carnahan Conference on Security Technology, pp: 397 – 401, October 1995.

[13] R. R. Vangala, S. Sasi, "Biometric Authentication for E-commerce Transaction", IEEE International Workshop on Imaging Systems and Techniques, pp: 113 – 116, May 2004

[14] M. Savvides, B. V. K. Vijaya Kumar and P. K. Khosla, “Cancelable Biometric Filters for Face Recognition”, IEEE Proceeding of the 17th International Conference on Pattern Recognition ICPR, vol. 3 pp: 922 – 925, August 2004.

[15] N. Morimoto, “Digital Watermarking Technology with Practical Applications”, Information Science special Issue on Multimedia Informing Technologies – Part 1, Vol. 2, No. 4, 1999.

[16] J. Zhao, E. Koch, C. Luo, “In Business Today and Tomorrow”, Communications of the ACM, Vol. 41, No. 7, pp: 67 – 72, July 1998.

[17] R. B. Wolfgang, E. J. Delp, "Overview of Image Security Techniques with Applications in Multimedia Systems," Proceedings of the SPIE Conference on Multimedia Networks: Security, Displays, Terminals, and Gatweys, Vol. 3228, pp. 297-308, November 2-5, 1997.

[18] A. Kejariwal, “Watermarking”, IEEE Potentials, Vol. 22, No. 4, pp: 37 – 40, October – November 2003.

[19] W. Bender, D. Gruhi, N. Morimota, A. Lu, “Techniques for Data Hiding”, IBM Systems Journal, Vol. 35, No. 3 & 4, 1996.

[20] M. M. Yeung, S. Pankanti, “Verification Watermarks on Fingerprint Recognition and Retrieval”, in Proc. SPIE, Security and Watermarking of Multimedia Contents, vol. 3657, pp. 66-78, January 1999.

[21] F. Hartung, M. Kutter, “Multimedia Watermarking Techniques”, Proceedings on IEEE, Vol. 87, No. 7, pp: 1079 – 1107, July 1999.

[22] M. Vatsa, R. Singh, P. Mitra, A. Noore, “Digital Watermarking based Secure Multimodal Biometric System”, IEEE International conference on Systems, Man and Cybernetics, pp: 2983 – 2987, 2004.

[23] M. Kutter, F. Jordan, F. Bossen, “Digital Signature of Color Images using Amplitude Modulation”, Proc. SPIE, Volume-3022, pp; 518 – 526, Storage and retrieval for Image and video Database, V. Ishwar, K. Sethi, R. c. Jain, Eds, 1997.

[24] M. M. Latha, G. K. Pillai, K. A. Sheela, “Watermarking based content Security and Multimedia Indexing in digital Libraries”, International Conference on Semantic Web and Digital Libraries (ICSD-2007). ARD Prasad & D. P. Madalli (Eds.): ICSD-2007.

[25] U. Uludag, B. Gunsel, M. Ballan, "A Spatial method for watermarking of Fingerprint Images", Proceedings of the 1st International Workshop on Pattern Recognition in Information Systems: In Conjunction with ICEIS, pp: 26 – 33, 2001.

[26] A. Noore, "An Improved digital Watermarking Technique for Protecting JPEG Images", IEEE International Conference on Consumer Electronics Consumer Electronics, ICCE, pp: 222 – 223, June 2003.

[27] N. F. Johnson, S. C. Katezenbeisser, "A Survey of Steganographic Techniques", Information Techniques for Steganography and Digital Watermarking, S. C. Katezenbeisser et al., Eds. Northwood, MA: Artec House, pp: 43 – 75, December 1999.

[28] P. Wayner, "Information Hiding Techniques for Steganography and Digital Watermarking", Artech House Publishers, 2000.

[29] N. Provos, "Defending Against Statistical Steganalysis", Proceedings on 10th Conference on USENIX Security Symposium, Vol. 10, pp: 24 - 24, August 2001.

[30] N. Johnson, S. Jajodia, "Exploring steganography: Seeing the unseen", IEEE Computing, Vol. 31, No. 2, pp: 26–34, February, 1998.

[31] N. F. Johnson, S. Jajodia, "Steganalysis of Images Created Using Current Steganography Software", Center for Secure Information Systems, George Mason University, Virginia, Fairfax.

[32] G. C. Kessler, "Hiding Data in Data", Windows & >NET Magazine, April 2002.

[33] R. J. Anderson, F. A. P. Petitcolas, "On the limits of Steganography", IEEE Journal of Selected Areas in Communication, Vol. 16, No. 4, pp: 474 – 481, May 1998.

[34] R. Popa, "An Analysis of Steganographic Techniques", The "Politehnica" University of Timisoara, Faculty of Automatics and Computers, Department of Computer Science and Software Engineering, 1998.

[35] P. Wayner, "Disappearing Cryptography – Information Hiding: Steganography and Watermarking", Second Edition, Morgan Kaufmann, 2002.

[36] A. D. Orebaugh, "Steganalysis: An Study of an Internet Search Engine and Steganography Detection Program", George Mason University, 2003.

[37] E. Dawson, J. Lopez, J. A. Montenegro and E. Okamoto, "BAAI: Biometric Authentication and Authorization Infrastructure", International Conference on Information Technology: Research and Education, pp: 274 – 278, August 2003.

[38]http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=14311&objectType=file

[39]http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=14312&objectType=file

[40] C. Kurak and J. McHugh, “A Cautionary note on Image Downgrading”, IEEE Proceedings on Computer Security Applications Conference, pp: 153 – 159, 30 November - 4 December, 1992.

[41] A. Ihmaidi, A. Al-Jaber, A. Hudaib, “Securing Online Shopping using Biometric Personal Authentication and Steganography”, IEEE Information and Communication Technologies, Vol. 1, pp: 233 – 238, April 2006.

[42] A. K. Jain, U. Uludag, “Hiding Biometric Data”, IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 25, No. 11, pp: 1494 – 1498, November 2003.

[43] A. Giannoula, D. Hatzinakos, “Data Hiding for Multimodal Biometric Recognition”, IEEE Proceedings on International Circuits and Systems, Vol. 2, pp: II-165 – 168, May 2004.

[44] G. Langelaar, I. Setyawan, R. L. Lagendijk, “Watermarking Digital Image and Video Data”, IEEE Transaction on Signal Processing Magazine, Vol. 17, pp: 20 – 43, September 2000.

[45] M. A. Turk, A. P. Pentland, “Face recognition using Eigenfaces”, IEEE proceedings on Computer Society Conference on Computer Vision and Pattern Recognition, pp: 586 – 591, June 1991.

[46] R. C. Gonzalez, R. E. Woods, S. L. Eddins, “Digital Image Processing Using MATLAB”, Pearson Prentice Hall, 2004.