Chapter 4

Iris Template Security

4.1 Introduction

Iris is the annular region of the eye bounded by the pupil and the sclera on either side. The visual texture of the iris is formed in the early years which quickly stabilizes after the first two years [10]. The iris texture is a very complex pattern which contains highly discriminating information. As such, iris based biometric systems are suitable for high end security applications where the FRR is the dominant factor over FAR. The iris texture pattern is considered to be the most distinguishable of biometric traits among individuals [146]. Furthermore, it is extremely difficult to surgically tamper the texture of the iris, which makes the resulting biometric system extremely secure. However, the stored iris features can be subjected to a wide variety of adversarial attacks. For instance, it has been demonstrated that the iris image can be reconstructed from the stored iris template [147]. The pioneering work for extraction

of iris features was done by Daugman [148], which still remains the benchmark model. The corresponding algorithm in [148] comprises four stages: (i) raw iris image acquisition, (ii) pre-processing stage wherein the iris portion is detected and unrolled to a normalized texture, (iii) feature extraction and generation of iriscodes, and (iv) feature comparison using a dissimilarity measure called *fractional Hamming distance*. These stages are now discussed briefly.

The first stage captures good quality iris images from the enrolling individuals. In most of the cases, the iris images are acquired in the near infrared region using specialized iris cameras. The next pre-processing step involves detection of the pupil and other boundaries of the iris. This iris ring is subsequently unwrapped into a normalized rectangular texture. Sometimes the contrast of the iris texture is enhanced by applying histogram stretching methods. In the third stage, features are extracted from the iris texture in the form of iris codes or Iris Pseudo Codes (IPC). The most popular approach for the extraction process follow the original scheme of [148], wherein 2D-Gabor filters were applied on the textures. The final phase consists of matching the iris codes by applying the bit-wise XOR-operator. This matching technique is utilized to count the mismatching bits such that the Hamming distance indicates a level of dissimilarity between the codes. Since this matching procedure contains only inexpensive XOR operations, several iris code comparisons can be efficiently performed withing a brief interval.

4.2 Motivation and Overview

The work presented in this chapter pertains to a recent development in the area of cancelable biometrics for iris based biometric systems. The authors in [122] introduced an adaptive bloom filter based cancelable scheme for iris which simultaneously provided critical security requirements such as *irreversibility* and *unlinkability*; as well as other desirable properties like recognition efficiency and compression of the stored data. This template protection scheme has become very popular due to its simplicity and non-requirement for pre-alignment of the biometric templates. However, the study of Hermans et.al [149] demonstrated that the original scheme was vulnerable to cross-matching based attacks, thereby refuting the claim of unlinkability. In particular, the authors presented a practical attack that distinguishes two Bloom filters (b, b') generated from the same data from two independent ones (c, c') with a probability of at least 96%. More recently, Bringer et.al [150] analyzed unlinkability on protected templates generated with two different iriscodes coming from the same iris (the attacks in [149] assumed that the protected templates were generated from the same iriscode). In this different setting, the authors were successful in performing a brute force attack for each block of codewords. Additionally, the authors were also able to disprove the irreversibility property by showing that the protected template leaks out information on the columns of the block without multiplicity and in disorder. They argued that with a block width of 16 or 32, this information was sufficient to reconstruct an iriscode which matched with other iris-codes from the same user.

The current work presents a cancelable scheme based on the study of [122] which provides all the desirable properties mentioned previously. The proposed framework borrows the concept of modified bloom filters from the previous works and adapts it in such a way so that the properties of unlinkability and irreversibility are preserved. Both empirical and theoretical analysis of the scheme have been performed for demonstrating the security requirements. The core idea behind the working of the scheme is based around the concept of *perfect secrecy*. This rigorous security notion states that the encoded data should leak no information about the original data, i.e. the apriori distribution of the original data should remain identical to its posterior distribution given the encoded data. Employing such a scrupulous constraint allows the proposed framework to be resilient against any adversarial threats under the ciphertext-only attack (COA) model. The basis for implementing perfect secrecy pertains to some critical facts. Firstly, as reported in [150], the security of the original Bloom filter based technique can be improved by randomly choosing one unique key per Bloom filter instead of using the same key for every Bloom filter. Secondly and most importantly, the key length should be at least equal to that of the Bloom filter itself. These properties are essential since the notion of perfect secrecy can only be applied under such stipulations. The proposed scheme consists of a key matrix where the size of each key is equal to the length of the Bloom filter. This key matrix forms the application specific secret (T) for this case. It should be noted that the guarantees of perfect secrecy act as an additional security layer over the original bounds attained by the modified Bloom filters.

4.3 Model Development

Now the detailed construction of the proposed scheme is presented.

4.3.1 Modified Bloom filters

As previously stated, a Bloom filter is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set. Basically, it is a bit array of length n, where all the bits are initially set to 0. For representing a set $S = \{x_1x_2, ...x_m\}$, a Bloom filter uses k different hash functions $\{h_1, h_2, ...h_k\}$ with range $[0, n-1]$. For each element $x \in S$, bits $h_i(x)$ of the Bloom filter are set to 1. An index can be set to 1 multiple times, but only the first change has an effect. To test if an element y is in S, it has to be checked whether all position of $h_i(x)$ are set to 1. Although this system is fast and efficient, there exists a chance of false positives whenever the bits have, by chance, been set to 1 during the insertion of another element y where $y \in S$ and $y \neq x$. A generic outline of a Bloom filter is diagrammatically illustrated in Figure 4.1.

The system proposed in [122] utilized modified Bloom filters to obtain alignment free cancelable iris biometric codes. Typically, extracted features from iris images are represented as two-dimensional binary feature matrix $\mathcal{I}$ of width W and height H. In their scheme, the authors divided $\mathcal{I}$ into a total of K number of equal sized blocks. Each column present is such a block is denoted as a *codeword*. Let the number of such columns present in a block be represented by l and the number of

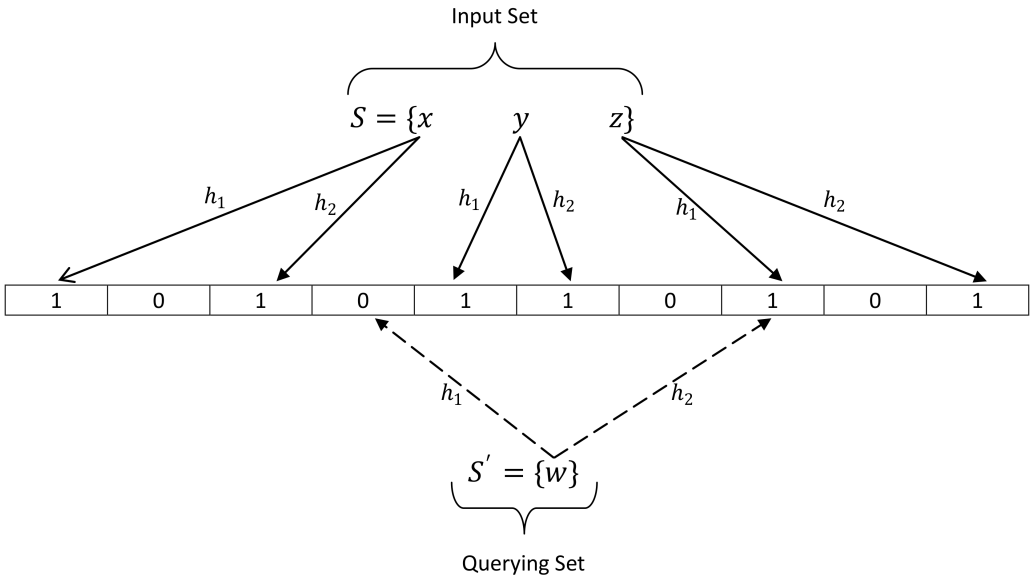


FIGURE 4.1: Generic outline of a Bloom filter.

bits (i.e. size of a codeword) in a column by w. The term l consequently also denotes the block size. Thus the total number of blocks equates to -

$$K = \frac{W \times H}{l \times w}$$

To make things convenient, the authors horizontally partitioned $\mathcal{I}$ into two equal halves of size $W \times (H/2)$ and constructed the blocks starting from the top of each half. In such a scenario, the total number of blocks is given by -

$$K = 2 \times \frac{W}{l}$$

It is noticeable that all the bits of the iriscodes may not get utilized in such a setting (all the bits are used iff $H = 2 \times w$).

In this adaptive model, each block corresponds to a separate Bloom filter. Thus a total number of K separate Bloom filters of length $n = 2^w$ are formed. Each codeword of a block is successively transformed to locations within the particular Bloom filter associated with the block. The transformation is realized by mapping each codeword to the index of its decimal value. For instance, a codeword c=[00110] of block i is mapped to the sixth index of the Bloom filter corresponding to block i. Thus the total size of the protected template equals $K \times 2^w = K \times n$ bits. Let the matrix representing the set of all Bloom filters be denoted by $\mathcal{B}$ where it takes values in the space $\mathcal{S} = \{0, 1\}^n$.

4.3.2 Proposed Modification

The basic scheme provided irreversibility since given a Bloom filter b, it is not clear from which codeword a distinct 1-bit in the protected template originated. Additionally, the notion of unlinkability was introduced by incorporating an application specific secret T (in the form of a bit vector equaling the size of codewords) which was XORed with each codeword prior to its mapping in the Bloom filters. The proposed scheme differs from the original one in this critical phase. As already demonstrated in [149, 150], this way of incorporating the secret T is flawed. In this framework, a key matrix $\mathcal{K}$ of dimension $K \times n$ is first constructed which consists of K keys each having length $n(= 2^w)$. Thus $\mathcal{K}$ is of the form -

$$\mathcal{K} = \begin{bmatrix} k_{1,1} & k_{1,2} & . & . & . & k_{1,n} \\ k_{2,1} & k_{2,2} & . & . & . & k_{2,n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ k_{K,1} & k_{K,2} & . & . & . & k_{K,n} \end{bmatrix}$$

$$= \begin{bmatrix} k_1 \\ k_2 \\ . \\ . \\ k_K \end{bmatrix}$$

The key matrix relates to the set $\mathcal{B}$ in an one-to-one correspondence (i.e. the first Bloom filter b_1 in $\mathcal{B}$ corresponds to the first key k_1 in $\mathcal{K}$).

4.3.2.1 Encoding (during enrollment)

Now an encoding function *Enc* is defined which is implemented during the enrollment phase. *Enc* basically XOR's the corresponding elements in $\mathcal{B}$ and $\mathcal{K}$. Thus -

$$Enc : b_i \oplus k_i = t_i \quad \text{where} \quad i = \{1, 2, ...K\} \tag{4.1}$$

The output of Enc is a set of K transformed templates (t_i), each of which consists of n bits. Let this set be represented by $\mathcal{T}$, where $\mathcal{T} \in \mathcal{S}$. The sets $\mathcal{K}$ and $\mathcal{T}$ form every subject's data which get stored.

4.3.2.2 Decoding (during authentication)

During the decoding phase, a querying feature code I' is presented for authentication purposes. A decoding function Dec is now defined. The function Dec converts the transformed templates back into their original form (with the help of keys) by XOR'ing them with the appropriate keys. Thus,

$$Dec : t_i \oplus k_i = b_i \quad \text{where} \quad i = \{1, 2, ...K\} \tag{4.2}$$

Dec outputs the original set of Bloom filters ($\mathcal{B}$) which are then compared with the Bloom filter based representation of the queried code I'. The complete framework is illustrated in Figure 4.2.

4.3.3 Comparison

As evident from the previous sections, recognition is implemented in this proposed scheme by comparing the Bloom filters obtained during enrollment (I) with those extracted from a feature trait (I') presented during authentication. Two Bloom filters b_i and b_j are compared by estimating the fractional Hamming distance (HD)

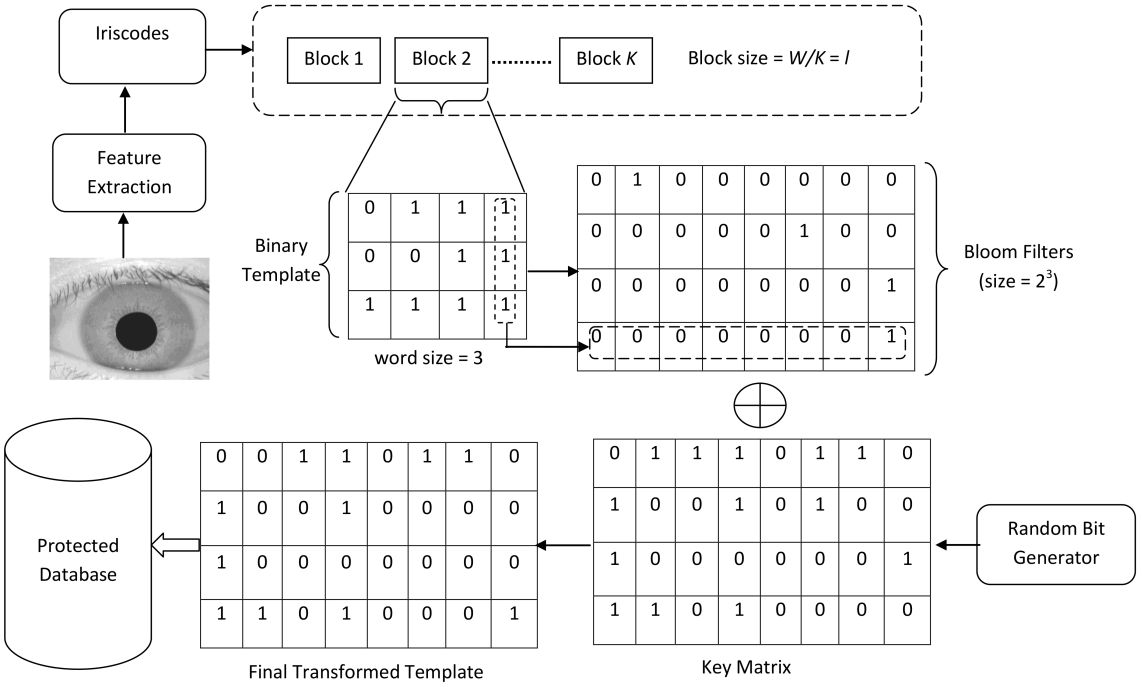


FIGURE 4.2: Modified Bloom filter based biometric template protection scheme.

between them, thereby producing a dissimilarity measure. HD is basically the sum of all detected disagreements between any corresponding pair of bits divided by the amount of compared bits. The overall dissimilarity score (DS) is then calculated by dividing HD by the total number of bits in b_i and b_j which are set to 1. Thus,

$$DS(b_i, b_j) = \frac{HD(b_i, b_j)}{|b_i| + |b_j|}$$

where $|b|$ represents the number of bits in b which are set to 1. This scoring technique has been used in all the previous related works.

4.3.4 Key Management

Generation and storage of the keys in a proper way is a vital part of this framework. Firstly, all the security guarantees of the model (analyzed in the next section) stems from the assumption that the keys are randomly generated from a uniform space $\mathcal{S} = \{0,1\}^n$. For practical implementation purposes, the keys should be generated using cryptographically secure pseudo random number generators (CSPRNG). CSPRNGs are pseudo-random number generators (PRNG) which follow some specific requirements like fulfillment of the next-bit test and tolerance against state compromise extensions [17]. A PRNG, on the other hand, is a family of deterministic polynomial time computable functions $G_k : \{0,1\}^k \rightarrow \{0,1\}^{p(k)}$ for some polynomial p if it stretches the length of its input ($p(k) > k$ for any k), and if its output is computationally indistinguishable from true randomness. In this model, the **CryptGenRandom** function which is included in Microsoft's Cryptographic API has bee used. This CSPRNG has been specifically designed to be cryptographically secure, thereby producing randomly generated bits.

Another facet of key management is related to their storage. In this design, the cancelable keys need to be stored in a separate storage unit so that they can facilitate during the authentication phase. Keeping the keys alongside the transformed templates is very risky since a leakage in the database would reveal all the template-key combinations, which consequently would nullify all the security properties of the system. For averting such scenarios, two alternatives are suggested. Firstly

the key matrix can be kept in user based tokens like smart-cards, which would be subsequently produced (by the user) at the time authentication. Secondly, the key matrix itself might be transformed into an encrypted form on the basis of another key (generated from a password) and a one-way trapdoor function. In this case, the user should present the password during authentication and decrypt the key matrix before matching.

4.4 Theoretical Security Analysis

The security of the framework is theoretically analyzed in this section from three perspectives. More specifically, the notions of unlinkability, irreversibility and information leakage associated with the framework are examined. It is noticeable that other desirable requirements like usability and pre-alignment are implicitly fulfilled in the framework. Regarding the adversary, it is assumed that he is a passive one who is able to observe only the data stored in the database. Thus the Ciphertext only attack (COA) attack model is considered in this analysis. This assumption is practical since other contemporary biometric security schemes are evaluated on the basis of this model. Some additional notations and observations related to this scheme are presented first. This would greatly facilitate in rigorously analyzing the associated security paradigms.

4.4.1 Additional Notations

As defined previously, the raw Bloom filters are denoted by the matrix $\mathcal{B}$, the key matrix by $\mathcal{K}$ and the protected templates by $\mathcal{T}$. Each row in these matrices basically represent a block of the feature biometric code. Thus $b_i(b_i \in \mathcal{B})$ represents a Bloom filter encoding the block i, $k_i(k_i \in \mathcal{K})$ represents the corresponding encoding key, and $t_i(t_i \in \mathcal{T})$ denotes the resulting protected Bloom filter. All the three matrices $\mathcal{B}, \mathcal{K}$ and $\mathcal{T}$ take values over a space $\mathcal{S} = \{0,1\}^n$, where n is the size of each Bloom filter ($n = 2^w$, where w is the size of the codewords). At this point a simplification of the model is made. The three matrices are mapped as bit vectors by concatenating the sequence of the rows present in them. For instance, all the K individual Bloom filters contained in $\mathcal{B}$ are sequentially row-wise concatenated to form a bit vector of size nK. These bit vectors are denoted by $\mathbb{B}$, $\mathbb{K}$ and $\mathbb{T}$ corresponding to the matrices $\mathcal{B}$, $\mathcal{K}$ and $\mathcal{T}$ respectively. Thus,

$$\mathbb{B} = b_1||b_2||...||b_K$$

$$\mathbb{K} = k_1||k_2||...||k_K$$

$$\mathbb{T} = t_1||t_2||...||t_K$$

where $||$ is the concatenation operator.

All the three bit vectors take value in the expanded sample space $S' = \{0,1\}^{nK}$.

Since the encoding (Enc) and decoding (Dec) functions are designed to work bitwise, these vector representations are equivalent to the original matrix representations.

Let the random variable X_B denote the Bloom filter vector $\mathbb{B}$, X_K denote the key vector $\mathbb{K}$ and X_T represent the protected template vector $\mathbb{T}$. For generalizing the framework, it is assumed that X_B follows an arbitrary distribution. However X_K is uniformly distributed over the space $\mathcal{S}'$ since the constraint that the keys should be randomly generated from a uniform key space was imposed beforehand. Henceforth the term 'vector' is dropped from this work to avoid any confusion or repetition.

4.4.2 Observations

Considering the construction of the proposed model, two vital observations are made. They are -

1. The keys are chosen randomly from a uniformly distributed key space. Since the protected template $\mathbb{T}$ results from only XOR operations with $\mathbb{K}$, X_T is also defined over a uniform distribution.

2. There exists a unique key $\mathbb{K}_i$ which decodes a protected template $\mathbb{T}_i$ to a Bloom filter $\mathbb{B}$. Here $[\mathbb{K}_i, \mathbb{T}_i, \mathbb{B}] \in \mathcal{S}'$. Formally,

$$\forall \mathbb{T}_i \in \mathcal{S}', \exists! \mathbb{K}_i \in \mathcal{S}' \quad \text{s.t.} \quad \mathbb{T}_i \oplus \mathbb{K}_i = \mathbb{B}$$

.

4.4.3 Unlinkability

Unlinkability stipulates that it should be impossible to distinguish between two protected templates originating from the same person. Let there exist two protected templates $\mathbb{T}_i$ and $\mathbb{T}_j$ which are formed from the same Bloom filter $\mathbb{B}$ (thus representing a single person) through two different keys $\mathbb{K}_i$ and $\mathbb{K}_j$. The following relation is required to be proven to establish unlinkability -

$$Pr[X_T = \mathbb{T}_i | X_B = \mathbb{B}] = Pr[X_T = \mathbb{T}_j | X_B = \mathbb{B}] \tag{4.3}$$

Now according to observation No.2,

$$Pr[X_T = \mathbb{T}_i | X_B = \mathbb{B}] = Pr[X_K = \mathbb{K}_i]$$

Moreover since X_K is uniformly distributed over the space $\mathcal{S}' = \{0,1\}^{nK}$,

$$Pr[X_T = \mathbb{T}_i | X_B = \mathbb{B}] = Pr[X_K = \mathbb{K}_i] = \frac{1}{2^{nK}} \tag{4.4}$$

Similarly,

$$Pr[X_T = \mathbb{T}_j | X_B = \mathbb{B}] = Pr[X_K = \mathbb{K}_j] = \frac{1}{2^{nK}} \tag{4.5}$$

Comparing Equation (4.4) and Equation (4.5), the unlinkability condition of Equation (4.3) is established.

4.4.4 Irreversibility

Irreversibility states that an adversary should not succeed (with high probability) in reconstructing the original templates from the protected templates stored in the database. Let's consider a protected template $\mathbb{T}$ which was encoded from a Bloom filter $\mathbb{B}$ through a key $\mathbb{K}$. The objective of the adversary in this case is to decode the value of $\mathbb{B}$ given $\mathbb{T}$ ($\mathbb{K}$ is not made public). Thus the success probability of the adversary is defined as -

$$Pr[success] = Dec(\mathbb{T})$$

Now the only way to decode $\mathbb{T}$ is by XOR'ing it with various possible combinations of the keys present in the space $\mathcal{S}'$ (brute force attack). Since the size of the key space is nK, the overall success probability of the adversary becomes -

$$Pr[success] = Pr[X_K = \mathbb{K}] = \frac{1}{2^{nK}}$$

With even standard values of n=256 and K=16, the success probability becomes minuscule. The basis for this high level of security are the facts that the key space is

made large by concatenating the individual keys and the keys are drawn randomly from a uniform distribution.

4.4.5 Information Leakage

Restricting the amount of information that the stored template leaks about the original biometric data is an important issue. For this model, the protected templates leak no information about the Bloom filters. The theoretical proof of this statement is closely associated with the concept of *perfect secrecy*. In a standard cryptographic scenario, perfect secrecy is an information theoretic guarantee that given an encrypted message (or ciphertext) from a perfectly secure encryption system (or cipher), absolutely nothing will be revealed about the unencrypted message (or plaintext) by the ciphertext. Regarding probabilities, it means that the probability distribution of the possible plaintexts is independent of the ciphertext.

For this case, the situation when a single person enrolls in multiple databases is considered. The final objective is to prove that the proposed scheme provides perfect secrecy in such a practical scenario. For achieving this purpose, a proof is presented corresponding to two databases. Without loss of generality, the proof can then be easily extended for more than two databases. Let an user U provide his biometric data $\mathbb{B}$ in two biometric databases DB_1 and DB_2 using two random keys $\mathbb{K}_1$ and $\mathbb{K}_2$. The protected templates are denoted by $\mathbb{T}_1$ and $\mathbb{T}_2$ for DB_1 and DB_2 respectively. The encoding functions for the two databases are -

$$Enc_1 : \mathbb{B} \oplus \mathbb{K}_1 = \mathbb{T}_1$$

$$Enc_2 : \mathbb{B} \oplus \mathbb{K}_2 = \mathbb{T}_2$$

For proving the notion of perfect secrecy, the essential condition is -

$$Pr[X_B = \mathbb{B} | X_T = (\mathbb{T}_1, \mathbb{T}_2)] = Pr[X_B = \mathbb{B}] \tag{4.6}$$

Theoretically, the apriori distribution of the message $\mathbb{B}$ should be equal to the posterior distribution of $\mathbb{B}$ given the template pairs $(\mathbb{T}_1, \mathbb{T}_2)$.

Now, let $\{\mathbb{T}_1, \mathbb{T}_2\} = \mathbb{T}$. Then,

$$\begin{aligned} Pr[X_B = \mathbb{B} \mid X_T = (\mathbb{T}_1, \mathbb{T}_2)] &= Pr[X_B = \mathbb{B} \mid X_T = \mathbb{T}] \\ &= \frac{Pr[(X_B = \mathbb{B}) \cap (X_T = \mathbb{T})]}{Pr[X_T = \mathbb{T}]} \\ &= \frac{Pr[X_B = \mathbb{B}] Pr[X_T = \mathbb{T} | X_B = \mathbb{B}]}{Pr[X_T = \mathbb{T}]} \end{aligned} \tag{4.7}$$

Again,

$$\begin{aligned} Pr[X_T = \mathbb{T}] &= \sum_{\mathbb{B} \in \mathcal{S}'} Pr[X_B = \mathbb{B}] Pr[X_T = \mathbb{T} | X_B = \mathbb{B}] \\ &= \sum_{\mathbb{B} \in \mathcal{S}'} Pr[X_B = \mathbb{B}] Pr[X_T = (\mathbb{T}_1, \mathbb{T}_2) | X_B = \mathbb{B}] \end{aligned} \tag{4.8}$$

Now $Pr[X_T = (\mathbb{T}_1, \mathbb{T}_2) | X_B = \mathbb{B}]$ denotes all the cases where $\mathbb{B}$ will be encoded to $\mathbb{T}_1$ and $\mathbb{T}_2$ through $\mathbb{K}_1$ and $\mathbb{K}_2$. Thus it is required to consider all the four cases

corresponding to the combinations of $\mathbb{T}_1, \mathbb{T}_2, \mathbb{K}_1$ and $\mathbb{K}_2$. Let an event $E(\mathbb{B}, \mathbb{K}, \mathbb{T})$ denote the case where a filter $\mathbb{B}$ is encoded to a protected template $\mathbb{T}$ through a key $\mathbb{K}$. Accordingly, four cases are obtained as follows:

$$E_1(\mathbb{B}, \mathbb{K}_1, \mathbb{T}_1), E_2(\mathbb{B}, \mathbb{K}_2, \mathbb{T}_2), E_3(\mathbb{B}, \mathbb{K}_1, \mathbb{T}_2), E_4(\mathbb{B}, \mathbb{K}_2, \mathbb{T}_1)$$

However according to observation no. 2, the probabilities of the occurrence of these events are-

$$Pr[E_1] = Pr[\mathbb{K}_1]; \quad Pr[E_2] = Pr[\mathbb{K}_2]$$

$$Pr[E_3] = 0; \quad Pr[E_4] = 0$$

To reiterate, this happens since there exists a unique key $\mathbb{K}_i$ which decodes $\mathbb{T}_i$ to $\mathbb{B}$.

Thus Equation 4.8 becomes -

$$P[X_T = \mathbb{T}] = \sum_{\mathbb{B} \in \mathcal{S}'} Pr[X_B = \mathbb{B}] Pr[X_T = (\mathbb{T}_1, \mathbb{T}_2) | X_B = \mathbb{B}]$$

$$\begin{aligned} &= \sum_{\mathbb{B} \in \mathcal{S}'} Pr[X_B = \mathbb{B}] \{Pr[E_1].Pr[E_2] \cup Pr[E_3].Pr[E_4]\} \\ &= \sum_{\mathbb{B} \in \mathcal{S}'} Pr[X_B = \mathbb{B}] Pr[X_K = \mathbb{K}_1] Pr[X_K = \mathbb{K}_2] + 0 \\ &= \sum_{\mathbb{B} \in \mathcal{S}'} Pr[X_B = \mathbb{B}] \frac{1}{2^n} \times \frac{1}{2^n} \\ &= \frac{1}{2^{2n}} \end{aligned} \tag{4.9}$$

Similarly,

$$\begin{aligned} Pr[X_T = \mathbb{T} | X_B = \mathbb{B}] &= Pr[X_T = (\mathbb{T}_1, \mathbb{T}_2) | X_B = \mathbb{B}] \\ &= \frac{1}{2^{2n}} \end{aligned} \tag{4.10}$$

Substituting values of $Pr[X_T = \mathbb{T} | X_B = \mathbb{B}]$ and $P[X_T = \mathbb{T}]$ from Equation 4.9 and Equation 4.10 into Equation 4.7,

$$\begin{aligned} Pr[X_B = \mathbb{B} \mid X_T = (\mathbb{T}_1, \mathbb{T}_2)] &= \frac{P[X_B = \mathbb{B}] P[X_T = \mathbb{T} | X_B = \mathbb{B}]}{P[X_T = \mathbb{T}]} \\ &= \frac{P[X_B = \mathbb{B}] \times \frac{1}{2^{2n}}}{\frac{1}{2^{2n}}} \\ &= P[X_B = \mathbb{B}] \end{aligned} \tag{4.11}$$

which is the required condition for perfect secrecy (stated in Equation 4.6). This concludes the proof.

4.5 Experiments and Results

In this section, the results obtained from experiments carried out on the proposed model are evaluated and analyzed.

4.5.1 Data Acquisition

4.5.1.1 Database

All of the experiments were carried out on the CASIA-IrisV1 database [1]. The database is publicly provided by the National Laboratory of Pattern Recognition (NLPR), Institute of Automation (IA), Chinese Academy of Sciences (CAS). The database itself includes 756 iris images captured from 108 eyes. For each eye, 7 images were captured in two sessions by an indigenous developed close-up iris camera. Among the 7 images, the first three samples were collected in the first session, whereas the last four samples were collected in the second session. All the images had a resolution of 320×280 pixels. For this purpose, the first 100 subjects of the database were selected, thus totaling to 7×100 = 700 iris images.

4.5.1.2 Feature Extraction

Regarding the pre-processing and feature extraction algorithms, the same procedures as those described in previous works [122, 124] have been followed. During

[1] CASIA-IrisV1, http://biometrics.idealtest.org/

pre-processing, the iris of a given sample image was detected and un-wrapped to an enhanced rectangular texture of 64×512 pixels. The weighted adaptive Hough algorithm proposed in [151] was used for this purpose. This two-stage segmentation algorithm employs a weighted adaptive Hough transform which iteratively refines a region of interest (ROI), thereby finding an initial center point. This is subsequently utilized to polar transform the image and extract polar and limbic boundary curves one after another from an ellipso-polar representation.

In accordance with previous works, modified versions of two different iris recognition algorithms provided in USIT – University of Salzburg Iris Toolkit v1.0 [2] were implemented. Herein normalized iris textures were divided into stripes to obtain 10 one-dimensional signals, each one averaged from the pixels of 5 adjacent rows (the upper 50×512 pixels were considered). The first feature extraction technique follows the proposed scheme of Masek [152], in which filters obtained from a Log-Gabor function were applied. Herein the texture was divided into 10 stripes thereby obtaining 5 one dimensional signals. Each signal resulted from averaging the pixels of 5 adjacent rows of the preprocessed iris textures obtained previously. Subsequently, a row-wise convolution with a complex Log-Gabor filter was performed on the texture pixels. The phase angle of the resulting complex value for each pixel was discretized into 2 bits, thus generating a total of 20×512 = 10240 bits. The second feature extraction technique was the implementation of the scheme proposed by Ma et.al [153]. In this case, a dyadic wavelet transform was performed on the 10 signals,

[2] http://www.wavelab.at/sources/

and two fixed sub-bands were selected from each transform. This resulted in a total number of 20 sub-bands. In each sub-band all local minima and maxima above an adequate threshold were located and a bit-code alternating between 0 and 1 at each extreme point was extracted. This procedure also produced $20 \times 512 = 10240$ bit iris codes. The various stages of feature extraction are shown in Figure 4.3.

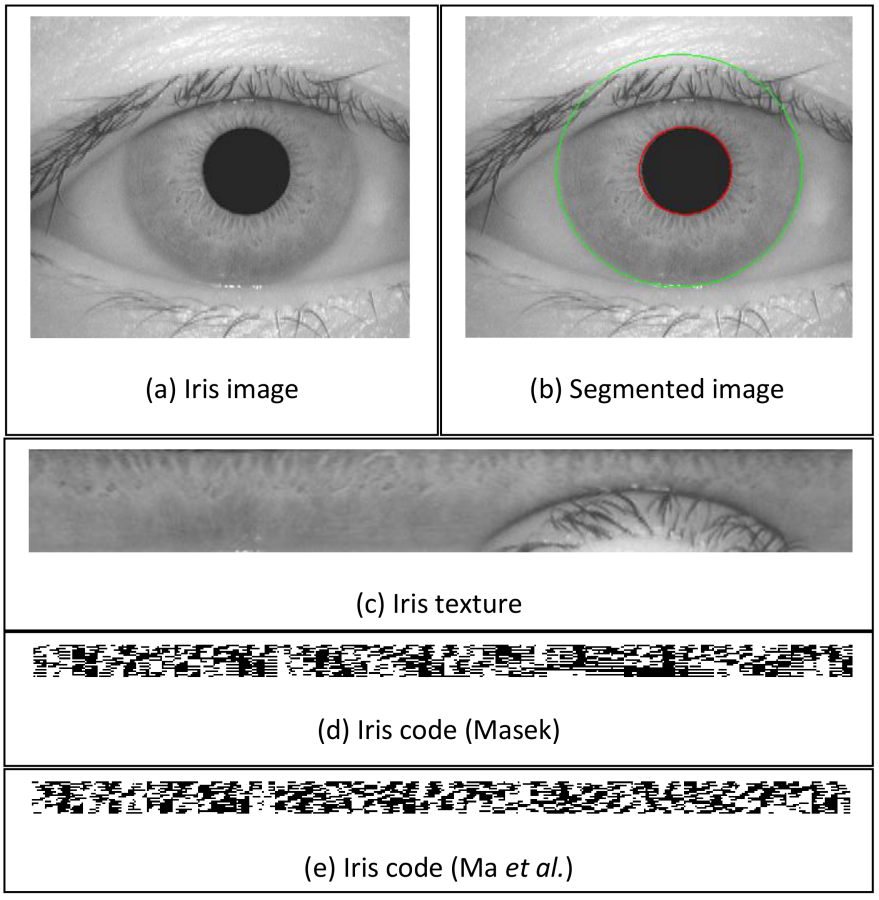


FIGURE 4.3: Various stages of feature extraction from iris images.

4.5.1.3 Matching Protocol

Each subject in the CASIA-IrisV1 database has a total of 7 iris samples which were obtained over two sessions. For genuine comparisons (intra-class matching), each

sample of every individual user was compared with all the other samples of the same user. This resulted in $\frac{7\times6}{2} = 21$ matching cases for every user, and a total of 21×100= 2100 overall genuine comparisons. Alternatively, the impostor scores (inter-class matching) were computed by sequentially matching a sample of a user with every other samples of all the other users. For instance, the first sample of user 1 was matched with all the 1st samples of all the other users. This was repeated for all the seven samples. Hence this protocol resulted in a total of $\frac{100\times99}{2} \times 7 = 34650$ impostor matching scores.

4.5.2 Framework Analysis

The analysis part of the framework is divided into two sections. In the first part the results obtained regarding the performance of the model is studied, while in the second part it's security properties are empirically demonstrated.

4.5.2.1 Performance Evaluation

The metrics which have been used for evaluating the performances of a setting are GAR at a targeted FMR and the resulting ROC curve. Firstly, the results obtained by implementing the feature extraction algorithms of Masek [19] and Ma et.al [20] without utilizing the Bloom filter based scheme are presented. The iris templates were aligned by shifting 8 circular bits in each direction and the minimum hamming distance among the resulting cases was considered as the best result for the

extractors. The biometric performance of both the feature extractors are compared through the resulting ROC curves in Figure 4.4. As seen from the illustration, the process of Ma et.al outperforms that of proposed by Masek. For instance, at a standard FAR rate of 0.01%, the GAR's of Ma et.al and Masek are 99.6% and 98.25% respectively. From an accuracy point of view, this is a considerable improvement.

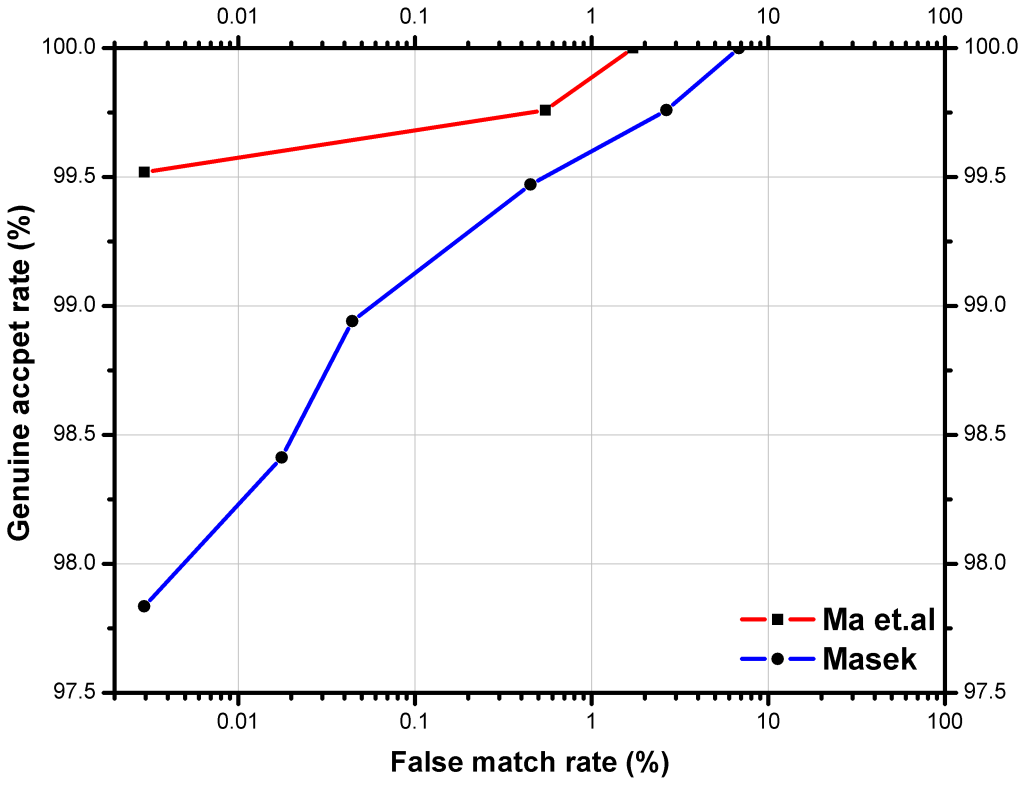


FIGURE 4.4: ROC curves of the original iris based biometric systems for both feature extractors.

The 20×512 bit iris codes were horizontally partitioned into two equal halves consisting of size 10×512 bits. This was done since they represent real and complex values or minima and maxima extracted from different wavelet sub-bands, respectively. The performance of the model was tested by varying the codeword length (w) in the set of $\{10, 9, 8\}$ and the number of blocks (K) in the set $\{2^6, 2^5, 2^4, 2^3, 2^2\}$

Word Size (w bits)	**Block Size (l bits)**				
	2^4	2^5	2^6	2^7	2^8
10	93.5	93.3	93	92.8	87
9	94.7	94.5	94.1	92	85
8	96.2	96	95	90	77

TABLE 4.1: GAR's (in %) at FAR = 0.01% for different system configurations using feature extraction technique of Masek [19]

(corresponding to the block length (l) in the set $\{2^4, 2^5, 2^6, 2^7, 2^8\}$). The performance of the biometric system subjected to the feature extraction algorithm of Masek is diagrammatically depicted by the ROC curves for codeword lengths (w) of 10, 9 and 8 bits in Figure 4.5(a), Figure 4.5(b) and Figure 4.5(c) respectively. As observable from all these figures, the best performance of the system results when K equals 64 (correspondingly l equals 16). The performance then systematically reduces for values of K ranging from 32 to 4. This decrease in the system performance for smaller number of blocks (corresponding to larger block size) can be attributed to the fact that too much local information gets lost in large sized blocks. The resulting GAR values at a FAR of 0.01% for all these cases are shown in Table 4.1. The best performance of the system was achieved for a codeword size of 8 bits and a block size of 16 codewords. In this optimum case, a GAR of 96% was observed for a FAR of 0.01%.

The same trend is noticed when the feature extraction procedure of Ma et.al was utilized. The performance of the biometric system peaked highest for the block size of 16 for any configuration of codewords. The ROC curves for codewords of size 10, 9 and 8 bits are shown in Figure 4.6(a), Figure 4.6(b) and Figure 4.6(c)

Word Size (w bits)	**Block Size (l bits)**				
	2^4	2^5	2^6	2^7	2^8
10	98.9	98.5	96.6	95	85
9	99	98	95.1	91	81
8	99.2	98.1	97	90.3	79

TABLE 4.2: GAR's (in %) at FAR = 0.01% for different system configurations using feature extraction technique of Ma et.al [20]

respectively. As observable from these figures, the best performance is observed in each case for K= 64, which corresponds to block length of 16. The reason for this superior performance of smaller sized blocks is similar to that of the previous case, i.e. incorporating more number of codewords in a block diminishes the local information associated with each block. Also similar to the previous case, the best performance amongst the three codeword sizes was achieved for w=8. Herein a GAR of 99.2% was observed at a FAR of 0.01%. The rest of performance results are depicted in Table 4.2.

The globally optimum configuration of the biometric system occurs when the feature extraction technique of Ma et.al was used along-with the system parameters of block size 16 bits and codeword size 8 bits. This observation is largely consistent with the original work [122].

4.5.2.2 Security Evaluation

The *unlinkability* property of the proposed model is experimentally demonstrated in this section. As previously stated, unlinkability refers to the infeasibility of cross-matching different protected templates of a single subject. This was one of the

major inadequacies of the original Bloom filter based scheme, due to which multiple attacks were successfully mounted against it. The unlinkability of any scheme can be empirically demonstrated by observing the overlapping region between the genuine (intra-class) and impostor (inter-class) scores distributions. The naive strategy of an adversary would be to match a sample template T' with a stored template T and observe the matching score. If the score falls outside the overlapping region, the adversary can deduce with very high probability whether or not the sample template T' belonged to the same subject whose template was matched with (T). Thus for the criteria of unlinkability, the overlapping region of the scores should be as large as possible (thereby diminishing the success probability of the adversary).

In this analysis, the case when l=16 bits (corresponding to K=64) was considered since this setting provides the best result. The score distributions involving the feature extraction technique of Masek is illustrated in Figure 4.7(a), Figure 4.7(b) and Figure 4.7(c) for codeword size of 10, 9 and 8 bits respectively. As evidently visible from these plots, the overlapping regions of the Bloom filter based templates after encoding approximates the maximum possible value (i.e. the genuine and impostor scores completely overlap). This result is very much contrasting to the score distributions before encoding, where the corresponding scores scarcely overlapped. This same observation is perceived while using the feature extraction technique of Ma et.al. The score distributions in this case are illustrated in Figure 4.8(a), Figure 4.8(b) and Figure 4.8(c) for codeword size of 10, 9 and 8 bits respectively. Thus it can be concluded that after encoding, the overlapping area attains an upper

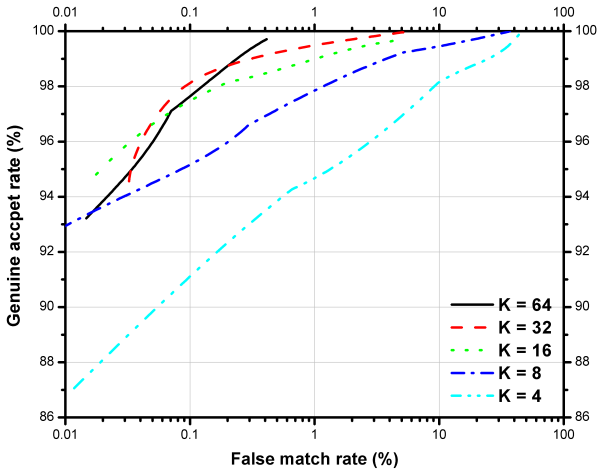
bound irrespective of the choice of any feature extraction algorithm. This perfect overlapping of the scores largely stems from the fact that the protected template vectors follow a uniform distribution, thus excluding any biased information.

4.6 Conclusion

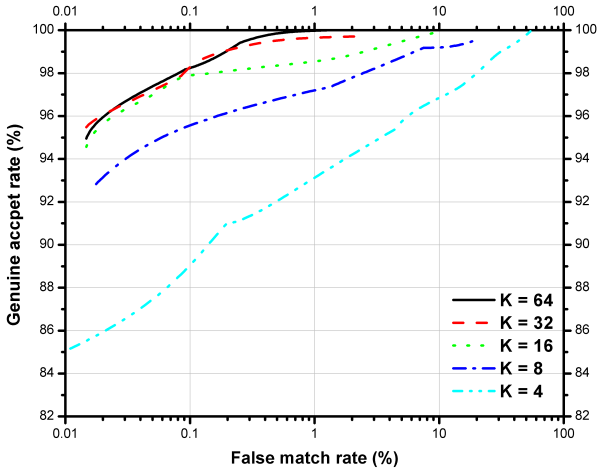
Biometric template protection is an active and challenging area of research. A good template protection scheme must adhere to certain properties or characteristics to render themselves useful. Achieving all of these properties simultaneously is a very difficult task. This issue has been partially addressed in this work by proposing a secure biometric protection scheme based on adaptive Bloom filters. Essentially, the original technique proposed in [122] has been altered to include robust security properties such as irreversibility, unlinkability and zero information leakage. These essential requirements perfectly complement other implicit properties associated with the Bloom filter based systems like efficiency (with respect to time) and non-requirement of any pre-alignment steps. Overall, the proposed design guarantees strong security and privacy properties in addition to other usability criterion.

The core of the security notions lies on the devised encoding and decoding functions. The functions themselves are conceptually very simple and are easy to implement. As opposed to the original work, the matching of the Bloom filter based templates is not performed in some transformed space. Instead, they are first decoded prior to matching, and subsequently matched in their original format. This enables in

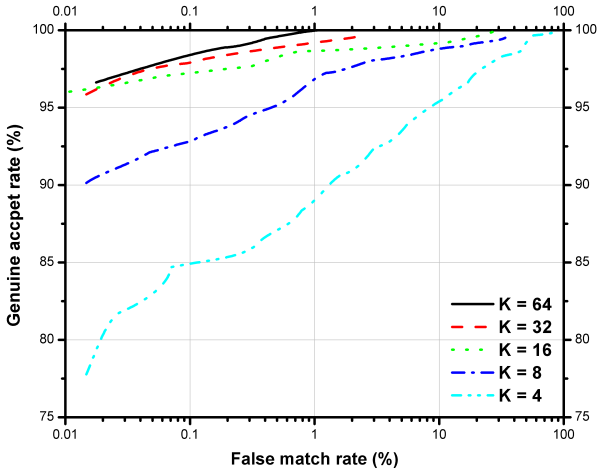
preserving the original performance rates of the Bloom filter based systems. It may be argued that the security of this model gets nullified if an adversary can intercept the decoded Bloom filters during authentication, but then again the filters themselves comprise of irreversibility guarantees (analyzed in [122] and confirmed in [149]) which render certain attacks (e.g. reconstruction of raw biometric templates) impractical. Thus the proposed model essentially guarantees a dual layer of security measures.



(a) w = 10

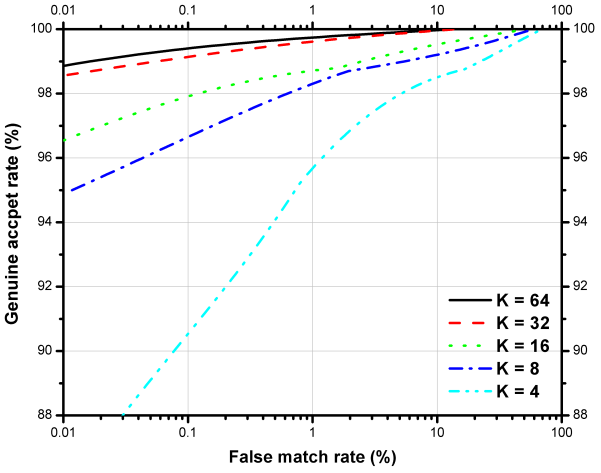


(b) w = 9

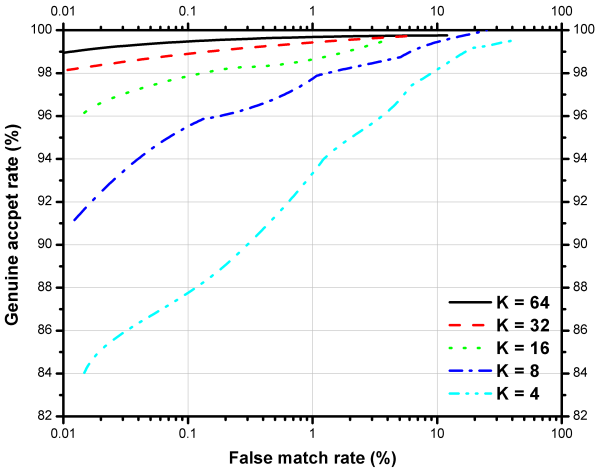


(c) w = 8

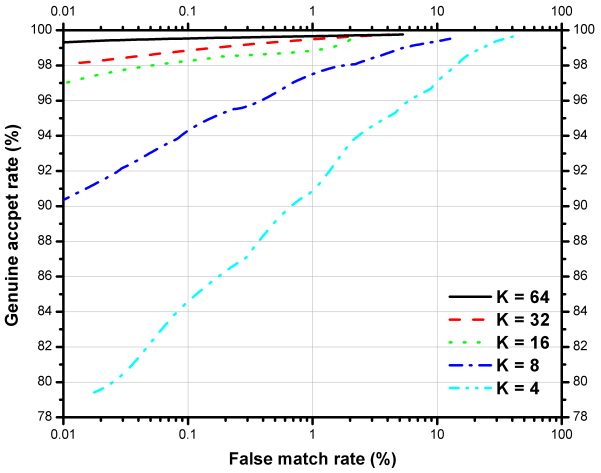
FIGURE 4.5: ROC curves for systems implementing the feature extraction algorithm of Masek [19].



(a) w = 10

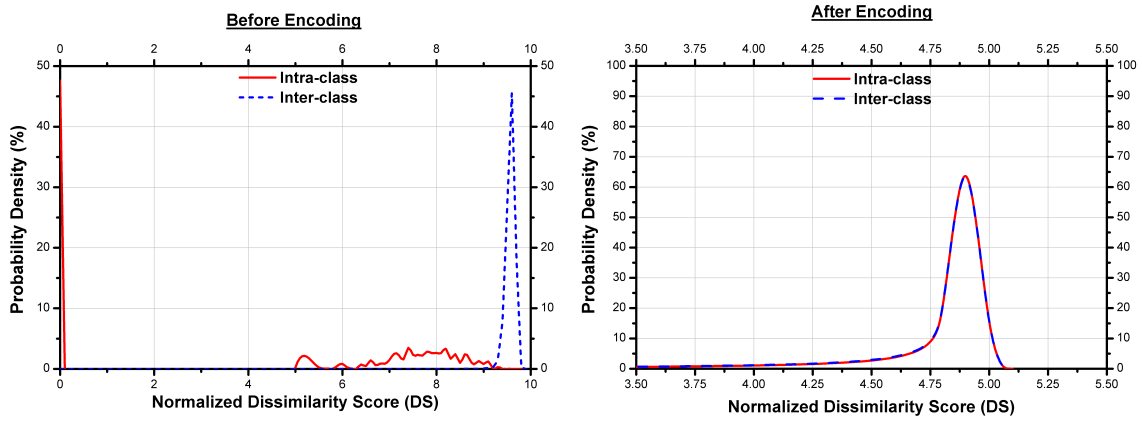


(b) w = 9

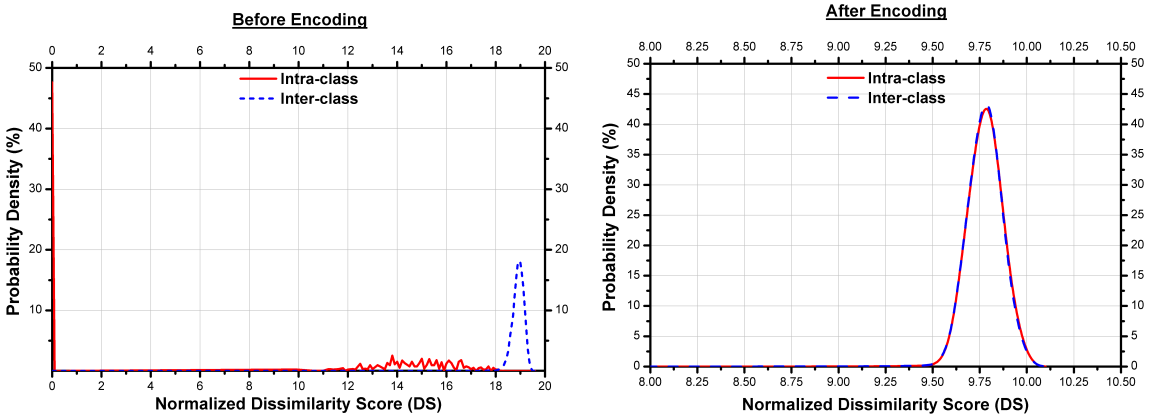


(c) w = 8

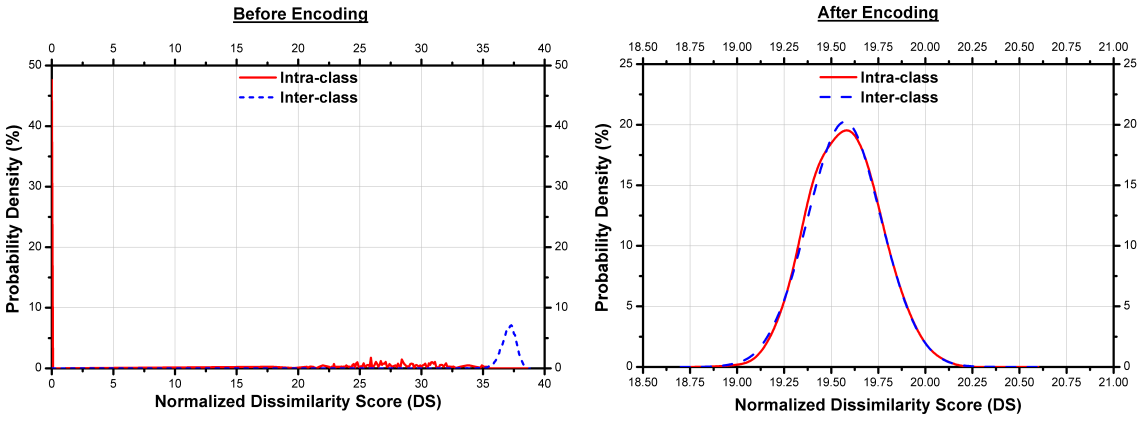
FIGURE 4.6: ROC curves for systems implementing the feature extraction algorithm of Ma et.al [20].



(a) w = 10

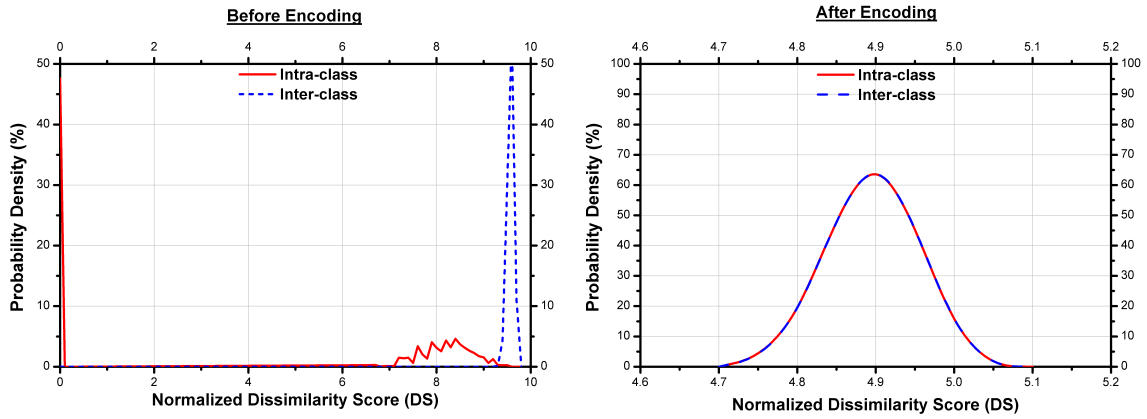


(b) w = 9

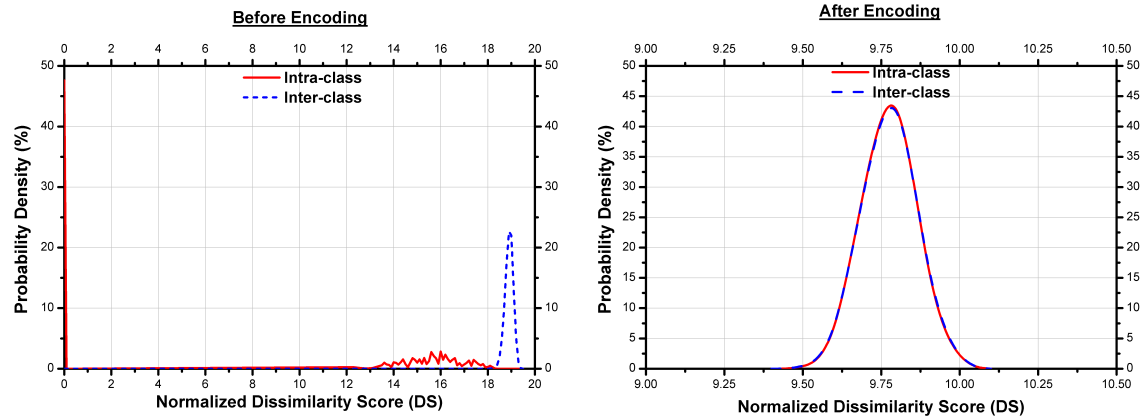


(c) w = 8

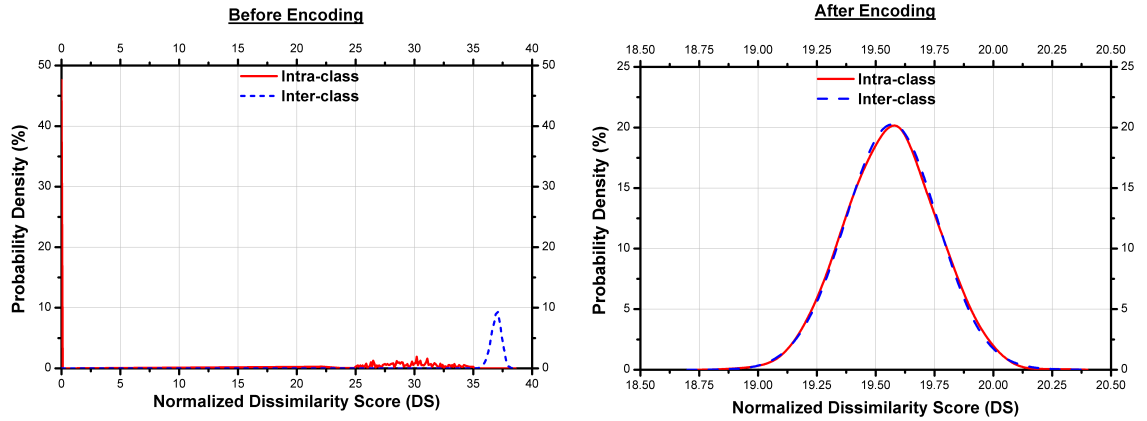
FIGURE 4.7: Scores comparison before and after encoding following feature extraction method of Masek [19]



(a) w = 10



(b) w = 9



(c) w = 8

FIGURE 4.8: Scores comparison before and after encoding following feature extraction method of Ma et. al [20]