Biometric Template Protecting Applied to Iris Recognition Systems

by K.T.J. de Groot

Master of Science thesis

Project period: March 2005 – March 2006
Report Number: 19-06

Commissioned by: Prof.Dr.Ir. P.H.N. de With (TU/e)

Supervisors:
Prof.Dr.Ir. P.H.N. de With (TU/e)
Dr. M. van der Veen (Philips Research)

Additional Commission members:
Dr. Ir. T.A.M. Kevenaar (Philips Research)
Ir. A.H.M. Akkermans (Philips Research)

The Department of Electrical Engineering of the Eindhoven University of Technology accepts no responsibility for the contents of M.Sc. theses or practical training reports
Title: Biometric Template Protecting Applied to Iris Recognition Systems

Author(s): K.T.J. de Groot

Keywords: biometrics, iris recognition, privacy protection, template protection

Abstract: Nowadays reliable authentication of people by their biometrics has become necessary in many applications. Biometric authentication systems have the potential to offer more security and convenience to users than common cryptographic methods can. A biometric represents the physical properties of an individual that cannot be lost or forgotten like for example a password. However storing biometrics in a database introduces privacy risks. Once a biometric has been measured, it is compromised forever and cannot be renewed, updated or destroyed. Moreover, unprotected biometric data can be used to perform cross-matching. The goal of this work is to show the feasibility of a privacy protected authentication system based on noisy iris biometrics. For this purpose two existing schemes, the iris recognition system designed by J. Daugman and a method for the secure protection of biometric templates, were joined. The idea is to prevent impersonation by storing information in a database or smartcard that is independent from the measured true iris features. Several strategies are proposed to select components in an iris template that are robust to noise. Moreover, a method to integrate mask vectors into the iris privacy protection system is explained. Mask vectors are typically used in iris systems, indicating whether an information feature component is possible corrupted by noise. This work shows that privacy and security can be achieved against an acceptable probability (1.24%) of rejecting a genuine verification attempt. This is higher than the performance of the unprotected iris recognition, which lies close to 0%. The probability of a false accepted impostor attempt remains 0%. All experiments in this work have been tested using iris images taken from the CASIA database.
Acknowledgements

I would like to gratefully acknowledge my professor Peter de With from Eindhoven Technical University (TU/e) and Dr. Michiel van der Veen from Philips Research for their supervision and advises during my Master's thesis work.

I also wish to thank all colleagues from the Information and System Security group at Philips Research, especially Dr. Ir. Tom Kevenaar and Ir. Ton Akkermans, as well as Fei Zuo, a PhD student at TU/e, for their support and encouragement. It was a pleasant experience to work in the atmosphere they provided.

Furthermore I gratefully recognize Dartagnan Biometric Solutions for providing an iris database.

Koen de Groot
Contents

1 General Introduction ........................................ 5
  1.1 Introduction to biometric systems ....................... 5
  1.2 Aim of this work .......................................... 6

2 Review of Biometrics ......................................... 7
  2.1 Biometric properties ....................................... 7
  2.2 Overview of biometrics .................................... 7
  2.3 Biometric authentication system ......................... 8
    2.3.1 Biometric enrollment ................................. 9
    2.3.2 Biometric verification ............................... 9
    2.3.3 Biometric identification ............................ 10
  2.4 Classification ............................................ 10
    2.4.1 Principle Component Analysis ....................... 10
    2.4.2 Linear Discriminant Analysis ....................... 11
  2.5 System performance ..................................... 12
    2.5.1 Matching algorithms ................................ 12
    2.5.2 Intraclass and interclass distribution ............. 13
    2.5.3 FRR and FAR ........................................ 13
    2.5.4 Receiver Operating Curve ........................... 15
    2.5.5 Statistics of system accuracy ..................... 15

3 State-of-the-Art in Privacy Protection ...................... 17
  3.1 Template protection scheme ............................. 17
    3.1.1 Enrollment and verification ......................... 17
  3.2 Earlier implemented template protection systems ....... 19

4 Iris Recognition ............................................. 20
  4.1 Historical background ................................... 20
  4.2 The iris as biometric ................................... 20
  4.3 Deriving a binary feature vector ....................... 21
    4.3.1 Detection of the iris region ...................... 22
    4.3.2 Normalization of iris .............................. 23
    4.3.3 Feature Extraction ................................ 25
    4.3.4 Quantization ...................................... 26
  4.4 Circular zones ......................................... 28
  4.5 Matching of binary templates ......................... 29
    4.5.1 Compensation for rotational differences ........ 29

© Koninklijke Philips Electronics N.V. 2006
4.5.2 Deriving parameter values for matching

4.6 Modeling the imposter distribution

5 Implementation of Iris Recognition System
5.1 Synthetic database
5.2 The CASIA data set
5.2.1 Rotation invariant approach
5.3 Statistics of system accuracy

6 Iris Recognition System with Template Protection
6.1 General architecture
6.1.1 Challenges
6.2 Template protection for iris systems
6.3 Classification in Hamming Distances
6.3.1 Bit Error Rate
6.3.2 Hamming Distance
6.4 Candidate secrets
6.5 System performance analysis
6.6 Composition of vectors Z'

7 Reliable Component Selection
7.1 Rotation alignment
7.2 Reliable components
7.3 Reliable component subset selection
7.3.1 No reliable component selection
7.3.2 Use all available reliable components
7.3.3 Random selection
7.3.4 Probability of a valid mask component
7.3.5 Discriminating components
7.3.6 Signal to Noise Ratio
7.3.7 Dependency between components
7.3.8 Combining several strategies
7.4 Overall evaluation of simulation results

8 Conclusions and Recommendations
8.1 Conclusions
8.2 Recommendations

A Error Correction Code (BCH)

References
Section 1

General Introduction

1.1 Introduction to biometric systems

Nowadays, reliable authentication of people is becoming increasingly necessary for various activities. Examples are: credit card purchase, boarding an airplane, visiting a foreign country or accessing highly restricted areas. Most common ways to authenticate people are by means of tokens or Personal Identification Numbers (PIN) or passwords. In some cases these traditional methods are not secure enough to guarantee a person's identity. Biometric identification refers to identifying or verifying the identity of people, based on their behavioral or physiological features [1]. Many physiological features could be used for identifying people [2]. The most common techniques are fingerprint, face, iris, retina, voice and signature recognition. The first scientific publications about measurements of humans for identification purposes were published at the end of the 19th century.

In general, biometric technologies offer additional convenience (compared to e.g. memorizing passwords) and security (e.g. in border control). Despite these benefits, it also introduces additional privacy and security risks. Examples are:

(i) Identity theft: it is easy to steal an unprotected biometric stored in a database of an authentication system. When an intruder obtained a biometric that belongs to another person, he has access to secure content or areas that require authorization. If for instance this stolen biometric is used for accessing a bank account, he would gain all financial details or even worse, could do monetary transactions. Stealing a stored biometric template1 is similar to stealing a password used for securing private content.

(ii) Renewability/Revocability: a stolen password can easily be changed to a new one, but an individual has a limited number of biometric features. If a fingerprint is used for identification, it can be renewed nine times. In case of iris identification, using the other eye would be the only alternative. Once a biometric has been measured, it is compromised forever and can not be changed.

(iii) Cross matching: storing biometric templates in a database creates a possibility for database owners to find out in which other databases their users are stored.

(iv) Extraction of medical data: biometric templates stored in a database can be used to retrieve

---

1 A template is a sampled, or contains representations of samples from a biometric.
information about possible diseases. For instance iridologists claim that it is possible to reveal physical and mental problems by examining the iris pattern.

Some of the security and privacy issues can be overcome with so-called template protection techniques, described in e.g. [3], [4], [5], [6], [7] and [8]. Recently, Philips has developed a method for securing and protecting biometric templates as well, [9], [10] and [11]. The basic idea of this method is not to store the biometric templates themselves but rather a scrambled version of them. Some important properties of this protection scheme are that it can deal with noise and can verify, with a high probability, that template X is equal to a noisy version of the same biometric X. The template protection system was already successfully integrated in face [12], fingerprint [13] and ear recognition [14].

1.2 Aim of this work

The aim of this project is to show the feasibility of the privacy protection authentication scheme, based on iris templates that allow for privacy protection and renewability. In this work, the unprotected iris recognition system developed by J. Daugman is used as reference model for integration of the theoretical privacy protection scheme. A deployment of template protection to iris recognition was already reported in [15] and [16]. It was based on error correction codes for which a maximum of 10% bit errors was allowed. This assumption seems rather unrealistic, as e.g. [16] reported that up to of 30% bit errors may occur in known iris recognition systems. Furthermore, the review in [16] mentioned that storing error correction bits, derived from an iris template, leads to some information leakage. In addition, the approach described in [15] provides no statistics about the achieved system performance.

This report is organized as follows: first, chapter 2 gives an overview of biometric properties and used systems as well as methods to analyse the system performance. Chapter 3 explains theory and aspects on the existing biometric template protection scheme. Subsequently, chapters 4 and 5 elaborate on the system on recognizing people by their irises developed by J. Daugman. The newly joint iris recognition and template protection system is discussed in chapter 6. Reliable component selection and accompanying analysis is discussed in chapter 7. Finally, recommendations and conclusions are given in chapter 8.
Section 2

Review of Biometrics

This chapter provides an overview of commonly used biometrics and its properties. Furthermore, different biometric system operation modes are explained, as well as methods to analyse their performance.

2.1 Biometric properties

Each different biometric trait of a person can be measured, quantified or stored in a system database, but not every biometric is just as good as the other one. To qualify the various biometrics there are characteristics defined that determine the quality of a certain biometric. An ideal biometric has the following properties:

- **Availability**: the entire population should have this biometric characteristic.
- **Permanence**: the biometric should not change over time.
- **Uniqueness**: ideally, no two persons should have the same biometric characteristics. But less strictly formulated, there should be sufficient variation over the whole population to distinguish individual biometrics.
- **Collectability**: the biometric is easy to measure with a sensor.
- **Acceptability**: there should not be a strong resistance in a user population to measure and collect the biometric.

Every implementation of a biometric system is a compromise between these biometric properties. There is no biometric that suits all qualifications completely. Especially the property acceptability is a sensitive discussion subject.

2.2 Overview of biometrics

This paragraph provides a brief description of commonly used biometrics. The generally used techniques for automated people authorization are: fingerprint, face, iris, retina, voice and signature recognition. Table 2.1 shows a comparison of several biometric recognition techniques. The table shows that iris recognition is one of the most secure and accurate means of biometric identification, as it scores high on the properties *permanence* and *uniqueness*. More on the iris as biometric will be explained in section 4.2.
<table>
<thead>
<tr>
<th>Biometric</th>
<th>Availability</th>
<th>Permanence</th>
<th>Uniqueness</th>
<th>Collectability</th>
<th>Acceptability</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNA</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ear</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Face</td>
<td>++</td>
<td>++</td>
<td>-</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Fingerprint</td>
<td>+</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Iris</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Retina</td>
<td>++</td>
<td>+</td>
<td>++</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Signature</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Voice</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>++</td>
</tr>
</tbody>
</table>

Table 2.1: Comparison of several biometrics. ++, + and - means respectively High, Medium and Low. (Adopted from [17]).

2.3 Biometric authentication system

A biometric authorization system can be seen as a pattern recognition system. A typical biometric recognition system is illustrated in Figure 2.1. It consists of three main modules:

(i) **Biometric localization**: for capturing the biometric out of a raw digital image. The output of a biometric detection device (e.g., an iris, fingerprint or face detector) is referred to as raw data. Some preprocessing is allowed in this definition, with the exception that neither redundancy nor information is added or decreased. In this raw data the biometric is localized.

(ii) **Feature extraction**: in order to make matching of biometrics more convenient, some features are extracted from the raw biometric data. This representation is called a template. A template only contains only data of a biometric that is necessary for matching.

(iii) **Recognition**: compares the extracted biometric pattern with candidate biometric templates stored in the database.

![Generalized biometric recognition system](image-url)
2.3.1 Biometric enrollment

During enrollment, see figure 2.2, the biometric measurements are captured, relevant information from the raw measurements are computed by the feature extractor and stored in a database or smartcard. Together with the template that represents the relevant data attributes, a unique ID number along with the subject’s name is linked to the template and stored in a database. In general, it is the case that more than one measurements are taken. From these measurements, the one with the best quality will be used to generate a template. In general this will improve the statistical performance of the authentication system.

![Figure 2.2: Enrollment stage of a biometric system](image)

2.3.2 Biometric verification

A biometric verification system, see figure 2.3, has as input two templates. The captured biometric data from a person who claims the identity, is compared with the template belonging to the claimed identity, stored in the database or smartcard. There is a unique identifier associated with every biometric template stored in the database. When making use of this ID number, the template belonging to the person who he claims to be, is easily retrieved. The system conducts a one-to-one comparison to validate whether the claimed identity is justified or not.

![Figure 2.3: Verification of a biometric](image)
2.3.3 Biometric identification

In identification mode, see figure 2.4, the system acquires the biometric sample from the subject and extracts the features from the raw measurements. It searches in the entire database for entries that have similarities to the input template. Consecutively, all database templates are matched to the input template. The identity of the person that belongs to the input template is assigned to the template from the database that resembles the input biometric. The person is rejected in case no database template entry that matches with the input biometric is found.

![Identification of a biometric](image)

2.4 Classification

This section describes two well known pattern classification methods, Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA). A feature vector that is created after extracting the features from the measured biometric might have a high dimensionality. A high dimensional feature vector could contain a lot of redundant information. The dimensionality of the feature vectors used in a biometric system, is related to the system's computation complexity. The aim of PCA/LDA is to discriminate between different feature vectors while reducing the redundant information in a feature vector as much as possible. Consequently, we obtain a more compact representation of the corresponding feature vector [18][19].

2.4.1 Principle Component Analysis

The main idea behind PCA is to find an accurate data (feature vector) representation in a lower dimensional space. Taking all of the data into account, PCA will compute a feature vector that has the largest variance associated with it, i.e. it preserves as much variance in the vector as possible [18]. The steps are as follows:

1. Acquire a data set (feature vector) \( D = \{x_1, x_2, \ldots, x_n\} \).

   Each sample \( x_i \) is a \( d \)-dimensional vector. With use of PCA, we wish to reduce from \( d \) to \( k \) dimensions.

2. Find the sample mean \( \hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i \)

3. Subtract the sample mean from every sample, giving \( z_i = x_i - \hat{\mu} \)
4. Compute the scatter matrix \( S = \sum_{i=1}^{n} z_i z_i^T \)

The scatter matrix is a measure of variance within the data set.

5. Compute the eigenvectors \( e_1, e_2, \ldots, e_k \), corresponding to the \( k \) largest eigenvalues of \( S \).

6. Let \( e_1, e_2, \ldots, e_k \) be the columns of matrix \( E = [e_1, e_2, \ldots, e_k] \). By choosing the \( k \) eigenvectors which correspond to the largest eigenvalues, achieves the dimension reduction and gives the new feature vector.

2.4.2 Linear Discriminant Analysis

If directions of maximum variance are important for classification, the data can be classified in different classes. However, the directions of maximum variance may be useless for classification. The aim of LDA, also known as Fisher Discriminant Analysis, is to find a projection to a line which preserves the direction of maximum variance, so that samples from different classes are well separated. The computed line represents the best discrimination between classes rather than describing data the best [18]. LDA minimizes the variance within a class of data, and maximizes the variance between different classes of data [20]. The steps are as follows:

1. Let there be a class \( C_1 \) of \( n_1 \) genuine samples \( x \), and a second class \( C_2 \) of \( n_2 \) imposter samples \( x \). Each sample in \( x_i \) in both classes has \( d \) dimensions.

2. Compute the sample mean in both classes. Giving

\[
\mu_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} x_i, \quad \text{and} \quad x_i \in C_1
\]

\[
\mu_2 = \frac{1}{n_2} \sum_{i=1}^{n_2} x_i, \quad \text{and} \quad x_i \in C_2
\]

3. Compute for each class the measure of variance within the class in terms of a scatter matrix denoted as

\[
S_1 = \sum_{i=1}^{n_1} (x_i - \mu_1)(x_i - \mu_1)^T
\]

\[
S_2 = \sum_{i=1}^{n_2} (x_i - \mu_2)(x_i - \mu_2)^T
\]

4. The total within class scatter matrix is given as

\[
S_w = S_1 + S_2
\]

5. A corresponding measure of variance between classes can be, in terms of a scatter matrix, defined as

\[
S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T
\]
6. The following expression describes the ratio that maximizes the separability between the classes as function of a vector $v$

$$J(v) = \frac{v^T S_B v}{v^T S_W v}$$

The function $v$ which maximizes the ratio is given as

$$v = S_W^{-1}(\mu_1 - \mu_2)$$

7. The desired $y$ for both classes is found by computing $y_1 = v^T C_1$ and $y_2 = v^T C_2$.

### 2.5 System performance

To authenticate people, the enrollment template is matched with the authentication template. Matching is usually done in a fuzzy manner. In this section we discuss some matching strategies, the intra and interclass distribution, and related FAR, FRR and EER.

#### 2.5.1 Matching algorithms

A biometric matching algorithm makes a decision by computing a measure of probability if both measurements are derived from the same person. There are different methods for matching two biometrics [1].

- A typical matching algorithm computes a similarity score, $S = \text{Score}(X, Y)$. Where $X$ and $Y$ are machine representation of an respectively enrolled and verification template. It gives a sort of likelihood if both measurement do match. This matching score is usually indicated with an amount of bit errors. A certain threshold value decides whether a verified template belongs to the same user, or to another person. If the matching score is below the system threshold level, both templates are assessed to be from the same person, i.e. a matching pair. When the score is above or equal to the threshold level, the template is assessed to belong to another person, i.e. a non-matching pair.

- An alternative way to compute matching scores is to determine distances, or dissimilarities $D = \text{Distance}(X, Y)$ between two samples $X$ and $Y$. Such distances might be determined as 'edit distances' from one biometric pattern to the next, or distances between exemplars in some vector space of biometric features.

When an enrolled biometric template and the verified template come from the same individual, this person is stated as a genuine person. When a person tries to claim another person's identity, this person is stated as an imposter. For each match of two biometric templates, the authentication system can have four possible outputs:

1. A genuine person is rejected
2. A genuine person is accepted
3. An imposter is rejected
4. An imposter is accepted

The decision output one and four are possible wrongly made decisions.
2.5.2 Intraclass and interclass distribution

A verification template can be modeled as the enrolled template with a certain amount of noise. The *intraclass* variation is defined as the variation within a *class* of genuine biometric measurements. Collecting scores of similarity when comparing template pairs belonging to the same person, results in a genuine or *intraclass distribution*. The *interclass* variation is specified as the variation between classes of genuine biometric samples. The *imposter* or *interclass distribution* represents similarity when templates belonging to different people are matched. The intraclass variation should be as low as possible to classify a biometric measurement to the same class. On the contrary, the interclass variation should be as high as possible, to be able to separate the different classes. Examples of both distributions are illustrated in figure 2.5. The distribution on the left part of the picture is the genuine distribution and the right side depicts the imposter distribution. Typically, these distributions follow a gaussian shape.

![Genuine and imposter distribution](image)

**Figure 2.5: Genuine and imposter distribution**

2.5.3 FRR and FAR

Figure 2.5 shows that both distributions overlap each other. This is generally the case in biometric authentication systems. This also means that a threshold level, visualized as a bar in the figure, usually cannot separate both distributions.

Because the distributions overlap each other due to noise difference between enrolled and verification templates, the biometric system will inherently falsely reject or accept authentication attempts. *False rejection* means a false match during a genuine request. In contrary, a *false acceptance* means an imposter attempt is wrongly matched. The area of the genuine distribution on the right side of the threshold level indicates the probability that the system wrongly rejects a template belonging to a genuine user. This distribution part is signified as the *False Rejection*
Ratio (FRR). The probability that the system falsely accepts a biometric template that does not belong to the same person, is called the False Acceptance Ratio (FAR). The FAR is the surface of the imposter distribution on the left side of the decision threshold.

Let's denote $X$ as the enrolled template and $Y$ stands for the verified template for authentication. The threshold level, used by the system to decide if two templates belong to the same person, is specified by $t$. Now we can define two hypotheses for the authentication system:

$H_1 : X = Y$, the enrolled template belongs to the same person as the verified template.

$H_2 : X \neq Y$, the enrolled template and verified template do not belong to the same person.

By making use of both hypotheses, we can specify the FRR and FAR as:

\[
FRR(t) = \int_0^t p(\tau | H_1 = true) d\tau
\]

\[
FAR(t) = \int_t^1 p(\tau | H_2 = true) d\tau
\]

Where $p$ is a probability distribution. Both probability curves are shown in figure 2.6. The intersection of the curves indicates the Equal Error Rate (EER). At this point both FRR and FAR have the same rate at a threshold $t$:

\[
EER(t) = FRR(t) = FAR(t)
\]

Setting the threshold level $t$ to a certain operating point gives an indication of the system security. A low threshold level $t$ in practice means that a high security level is reached, as in this case a high amount of errors or a large distance between two templates is not tolerated. In contrast, if the operating point is set to a high threshold level, a low security level is accepted, i.e. easy access is more likely.
2.5.4 Receiver Operating Curve

Plotting the FAR against the FRR produces another performance indication of a biometric system. Such a plot is called the Receiver Operating Curve (ROC). Figure 2.7 depicts the ROC when plotting the FAR and FRR of figure 2.6 against each other. The diagonal line in this plot indicates when the FRR and FAR are equal, i.e. \( x = y \). The intersection of the diagonal line and ROC-curve specifies the EER rate. The ROC gives a clear representation of the separability of the genuine and imposter distribution. Suppose both distributions do not overlap. In this case the ROC-curves goes straight along the \( x \) and \( y \) axis. Shifting both distribution towards each other until they overlap entirely will move the ROC-curve towards the thick diagonal line in figure 2.7, departing from \( x = 0 \). When the distributions entirely overlap each other, the ROC curve would lie on the diagonal line from FAR=1 to FRR=1, i.e. \( y = 1 - x \). The greater the overlap between both distributions, the smaller the surface between the ROC-curve and diagonal line will be, indicating a worse system performance.

\[
\text{FRR}=\text{FAR} \quad \text{EER}
\]

Figure 2.7: Example of the Receiver Operation Curve (ROC)

2.5.5 Statistics of system accuracy

This section provides the method for calculating the mean matching score \( \mu_S \) and standard deviation value \( \sigma_S \) for the genuine and imposter distribution. After matching all templates belonging to the intra class, the collected scores of similarity \( S \) compose a set \( S_{gen} \) as follows:

\[
S_{gen} = \text{Score}(X_m^p, Y_q^m)
\]  

for all \( m, p \) and \( q \) where,

\[
m = [1..N_U]
\]
\[
p = [1..N_X]
\]
\[
q = [1..N_Y]
\]

© Koninklijke Philips Electronics N.V. 2006
This set contains all $S$-values that define the genuine distribution. Something similar holds for the set $S_{imp}$, containing all $s$-values that define the genuine distribution.

$$S_{imp} = \text{Score}(X^p_m, Y^q_n)$$

for all $m, n, p, q$ and $m \neq n$ and where in this case,

$$m, n = [1..N_U]$$
$$p = [1..N_X]$$
$$q = [1..N_Y]$$

Mean of Matching scores The mean $s$-value of set $S_{gen}$ is defined as:

$$\mu_{S_{gen}} = \frac{1}{N_U \cdot N_Y \cdot N_X} \sum_i S_{gen,i}$$

(2.8)

Where $N_U, N_Y, N_X$ is related to the total number of templates matches for the intra class. For the set $S_{imp}$ the mean value is computed as:

$$\mu_{S_{imp}} = \frac{1}{N_U \cdot (N_U - 1) \cdot N_Y \cdot N_X} \sum_i S_{imp,i}$$

(2.9)

Variance of Matching scores The variance of set $S_{gen}$:

$$\sigma^2_{S_{gen}} = \frac{1}{N_U \cdot (N_U - 1) \cdot N_Y \cdot N_X} \sum_i (S_{gen,i} - \mu_{S_{gen}})^2$$

(2.10)

For set $S_{imp}$ the variance is computed as:

$$\sigma^2_{S_{imp}} = \frac{1}{N_U \cdot (N_U - 1) \cdot N_Y \cdot N_X - 1} \sum_i (S_{imp,i} - \mu_{S_{imp}})^2$$

(2.11)
Section 3

State-of-the-Art in Privacy Protection

This chapter discusses the theoretical scheme to protect the templates stored in a biometric authentication system as well as earlier implemented template protection systems.

3.1 Template protection scheme

A straightforward method to ensure security and privacy would be to store templates in a database by making use of a one-way hash function. It is considered to be impossible to retrieve the input back from the output produced by the one-way hash function. As input of the hash function, one should use the biometric enrollment template and store this in the database. To authenticate a person, a hashed version of the verification template is generated and matched with the hashed enrollment template. However, a verification measurement is with high probability almost never exactly the same as the enrollment biometric stored in the database. A verification measurement can be modeled as:

$$Y = X + N$$  \hspace{1cm} (3.1)

This means that the verification template $Y$ is in essence a noisy version of enrollment template $X$. These noise components can be introduced due to difference in camera angle and luminance variation between both capturing sessions. Also biometrics can change over time. However, the hash function produces completely different outputs even when there is only a small difference between two input vectors. It is very likely that the hashed versions of biometric $X$ and $Y$ differ, when both outputs produced by the hash function are compared. The template protection scheme illustrated in figure 3.1 overcomes this problem.

The principle idea behind this scheme is that a random bit string (secret) acts as a reference for an individual that has access to the data. The exact same random bit string can be retrieved during verification in order to authenticate this individual, by making use of helper data. Helper data makes it possible to maintain the advantage of a cryptographic hash function. The retrieved secret is corrupted due to noise difference between enrollment template $X$ and verification template $Y$. An error correction algorithm should correct sufficient disagreeing bits between both secrets in order to decide if they belong to a genuine person.

3.1.1 Enrollment and verification

A detailed explanation about the functionality of the template protection scheme is described in this section. The left and right side of the figure 3.1 refers respectively to the enrollment and
verification phase.

Enrollment During enrollment, helper data $W_1$ is derived from template $X$. Helper data $W_1$ indicate those components in the enrollment template that are reliable or robust to noise influences. It also guarantees that a unique vector can be derived from an individual, which contains the same components from the enrollment and verification template. The vector that contains the bit values from the reliable components indicated by $W_1$ is denoted by $Z$. Every enrolled person obtains a random generated secret $S$. This secret $S$ is encoded with an error correction algorithm to code word $C$. In order to prevent impersonation, the output $C$ of the encoder is added to vector $Z$ which results in vector $W_2$. This vector is a second form of helper data that is stored in the database. Furthermore, the secret $S$ is hashed, giving $h(S)$, and stored in the database.

Verification During verification, helper data $W_1$ and $W_2$ corresponding to the user that claims his or her identity, are chosen in the database. Helper data $W_1$ selects in verification template $Y$ the same reliable components as were chosen in enrollment template $X$. The resulting vector is specified by $Z'$. Addition of $W_2$ and $Z'$ reassembles the original code word $C$, but is corrupted by noise. Subsequently, codeword $C'$ is decoded to a guessed secret $S'$. Only in case the number of disagreeing bits between the original secret $S$ and guessed secret $S'$ is within the range of the error correction capability, codeword $C'$ is decoded to the correct secret $S'$. The output of the decoder is hashed and matched with the stored hashed secret $S$. Only if $h(S)$ and $h(S')$ are equal, a positive match is registered.

The data stored in the database does not contain sufficient enough information to retrieve the original features measured during enrollment, as helper data $W_1$ and $W_2$ are both statistically independent of secret $S$ and because of the fact that it is very unlikely to regain the input secret $S$ of the one-way hash function from the generated output. The privacy of every individual that has access to the database is protected by this scheme.
3.2 Earlier implemented template protection systems

Philips already demonstrated practical template protection schemes applied to binary templates, derived from fingerprint [21][13], face [19][12] and ear data [14]. For all schemes it holds that binary feature vectors are used in the template protection scheme. Concerning fingerprint template protection, a binary representation of the fingerprint images is created by extracting and quantizing the reliable components with the highest signal to noise ratio during the enrollment phase, using statistics derived from the individual’s feature vectors as well as the overall enrollment fingerprint data. Experiments have shown that the implemented scheme achieves an EER of approximately 4.2% with a secret length of 40 bits.

For face template protection as well, it holds that in order to find reliable and robust bits in user’s feature vector, statistics derived from the individual’s enrolled face data and entire enrolled population are used. The value at each \( k^{th} \) position in the feature vector is modeled using the normal distribution. The probability of an error at position \( k \) when making a new measurement, gives information about the robustness of this component. Components with the highest reliability are chosen for creating the binary feature vector. An acceptable performance of \( \text{FRR} = 3.5\% \) and \( \text{FAR} \approx 0\% \), is achieved with a secret length of 58 bits.

Regarding the ear template protection, prior to deriving helper data from the ear measurements, the system performs a random orthogonal transformation on the ear feature vectors, obtained after measuring the Headphone-to-ear-canal Transfer Function. The achieved system performance is \( \text{EER} = 3\% \) with a secret length of 100 bits.
Section 4

Iris Recognition

This chapter explains Daugman’s method to recognize people by their irises, provided in [22][23].
The test data in this project is generated using his algorithm.

4.1 Historical background

The idea to use irises for identification of people was originally suggested by ophthalmologist
Frank Burch in 1936. He claimed that the structure of an iris seems to be unique and stays stable
with age, at least after childhood. In 1949 James Doggart was the first who published about it in
an ophthalmology textbook. Two other ophthalmologist scientists, Alan Safir and Leonard Flom,
patented this idea in 1987, but did not know how to implement it. They consulted John Daugman
of Harvard University to design an algorithm to implement the idea of recognizing people by
their irises. Daugman patented the algorithms he designed in 1994. In the field of iris recognition
technologies, John Daugman is considered to be the founder of all current commercial systems
and research activities. Systems based on his iris recognition algorithms described in [22][23],
are the most successful and well known, although there are other researchers who developed iris
recognition algorithms. Important iris recognition systems different from Daugman’s method,
are created by Wildes [24][25], Boles and Boashash [26], Sanchez Reillo and Sanchez Avila
[27] and Lim [28].

4.2 The iris as biometric

The iris of an eye, see figure 4.1, is defined as the pigmented, round, contractile membrane of
the eye, suspended between the cornea and lens and perforated by the pupil. The iris controls
the amount of light on the retina inside the eye. This is done by small muscles that can adjust the
pupil’s size and inherently the size of the iris. The pupil can cover between 10 and 80 percent
of the iris size. The average diameter of the iris is 12 mm. The creation of the irises of a human
being starts somewhere after 3 months of the gestation. The development of the iris pattern is
finished after about 5 months. A chaotic process determines the structure without any genetical
dependence. The only property of an iris that is dependent of genetic factors is the iris color.
Consequently twins do not have the same iris structure. Even the iris patterns in both eyes of
one individual are different. During the life of a human being the iris pattern will not change
significantly. Only after e.g. a disease or operation the pattern can change [24][29].

Iris recognition is one of the most reliable methods of recognizing people by their biometrics.
The iris is more stable compared to other biometrics (e.g. face and voice). The basic principle of reliable and secure biometric authorization implies that reliable biometric measurement must have minimal variation in time and maximal variation between people [29]. Iris characteristics almost approach this definition of an optimal biometric, as independent variations produce greater uniqueness.

In addition to the fact that iris recognition has a good performance concerning the reliability of identifying people, it is also one of the least invasive biometric recognition systems. It is almost impossible to mislead an iris recognition system with an artificial iris [23]. During the recognition session, a number of iris images are made with different luminance values to illuminate the iris. As a consequence, the pupil diameter varies as well, due to the luminance variations. An iris pattern for example, printed on a contact lens will not be recognized by the system. The algorithms used for iris recognition discover when the pupil’s diameter did not change during the luminance variations, with the result that it is detected as a fake iris. Therefore, the combination of uniqueness, permanence and non-invasiveness, denotes that iris biometrics result in a high security level to authorization applications [30].

4.3 Deriving a binary feature vector

The matching starts with a picture of the eye region. This section describes the process of generating a binary iris feature vector, representing the uniqueness of the iris, from the picture of the eye. The general process flow is as follows:

- *Detection of the iris region:* localize the boundaries of the pupil, iris and eyelids.
- *Normalization:* create a representation of the iris to be able to match different irises with varying sizes.
- *Feature extraction:* extract the discriminant features of the iris.
- *Quantization:* encode the feature to a binary representation.

These steps are typical for an iris recognition system and will be further explained this section. Figure 4.2 depicts the iris recognition scheme in functional blocks.
4.3.1 Detection of the iris region

The first step in figure 4.2 is localizing the iris in the input picture. During enrollment, multiple pictures of the same eye are made. Out of the sequence of images, a picture is selected that satisfies a minimum picture quality criteria, e.g sufficient focus and visible iris texture. This picture is further being analyzed to localize the iris region. It is necessary to exclude the eyelids when they occlude the iris, and to detect the inner and outer boundaries of the iris.

The pupil and iris regions each have their own bounding circle radius and center location. The center point of the iris rarely has the same position as the pupil's center point. This means that the center points and the bounding circles of the pupil and iris must be found separately. To locate the circular iris and pupil regions, Daugman makes use of an integro-differential operator.

\[
\max_{(r,x_0,y_0)} \left| G_\sigma(r) \star \frac{\partial}{\partial r} \int_{r_0,y_0} \frac{I(x,y)}{2\pi r} \, ds \right|
\]

The above formula behaves as a circular edge detector, where \( I(x,y) \) is a picture of an eye in image domain \((x, y)\). The parameter \( r \) stands for the radius of contour circle \( s \) with center points \( x_0 \) and \( y_0 \). The operator searches in a circular defined path for the biggest difference in pixel values in image domain \((x, y)\) by varying the radius \( r \) and the center coordinates \((x_0, y_0)\) of contour circle \( s \). \( G_\sigma(r) \) is a Gaussian smoothing function. The operator is applied iteratively on the by \( G_\sigma(r) \) blurred image. For every iteration new values in parameter space \((r, x_0, y_0)\) are computed with reduced amount of smoothing scale \( s \), in order to achieve precise localization of the maximum changed pixel values in contour integration path \( s \). After with single pixel precision coarse to fine iterative search of the iris boundaries, the location of the eyelids is found in a comparable way. The contour integration path \( s \) for finding the iris boundaries eyelids, is changed from a circular to an arc shaped path, to be able to find the eyelid boundaries. A second degree polynomial function is fitted, as a parabolic shape is a good estimation for the eyelids.

From input picture\(^1\) 4.3(a) the boundaries after detecting the iris region are visualized in figure 4.3(b). Knowing all boundaries of the iris region will set up a mask \( M(x,y) \), see figure 4.4(a), that can be projected on the original image 4.4(b). In Daugman’s system, noise forms signified by the mask can be due to:

1. Occluding eyelids and/or eyelashes in the iris region.

\(^1\)This photo is taken from the CASIA database[31]. The filename of the image is 027_1_1.bmp, it has a resolution of 320x280 pixels.

© Koninklijke Philips Electronics N.V. 2006
Figure 4.3: On the left the input image is depicted. The figure on the right illustrates the localized iris boundaries in the input picture.

2. Specular reflections visible in the iris region, caused by the capturing device.
3. Boundary artifact caused by hard contact lenses.

Figure 4.4: The left picture illustrates the mask which covers the area that is not considered to be part of the iris region. In the picture on the right, the mask is projected on the input image.

4.3.2 Normalization of iris

Once the iris is isolated from, a picture the following processing step is normalization of the iris representation. The size of the iris varies due to variations in luminance. When, during enrollment and verification, different luminance values were used while illuminating the person's eye, the amount of isolated iris region will vary as well. Furthermore, the distance between capturing device and iris could also vary between both sessions causing different iris regions. In order to deal with variations in size between iris images, it is important to have a representation of the iris pattern that is invariant for changes in:

- **Size of eye**: depends upon the distance between the eye and lens. Also the magnification of the camera itself can increase or reduce the size of the captured eye.

- **Size of pupil**: different illuminations might cause variations in the pupil's diameter.
- **Location**: the location of iris within the image
- **Orientation**: depends on head tilt and eye rotation in the socket
- **Angle**: variation in angles can be due to the position of the camera, person and mirror angles.

The normalization after localization of the iris region will produce iris regions that have the same constant dimensions. This means that two images of the same iris, captured under different conditions, will have characteristic features at the same spatial location. This normalization is carried out by remapping the iris from the cartesian \((x, y)\) domain to a polar coordinate system \(r, \theta\). Defined as:

\[
I(x(r, \theta), y(r, \theta)) \rightarrow I(r, \theta) \tag{4.2}
\]

This transformation results in a double dimensionless coordinate system. The polar coordinate \(\theta\) is dimensionless as it is ranged from \([0, 2\pi]\). The radial variable is dimensionless as well, because the distance between the pupil boundary and the outer boundary of the iris always ranges between \([0, 1]\).

Daugman describes this modeled coordinate system as the stretching of a homogeneous rubber sheet [22]. This model, see figure 4.5, assigns to every point in the \((x, y)\) domain a pair of coordinates \((r, \theta)\), where \(r\) is the radius mapped between \([0, 1]\) and \(\theta\) is an angle between \([0, 2\pi]\). This rubber sheet model (normalization process) will further be elaborated on in the following example. The algorithm described in this example makes use of an iris normalization method adopted from.

**Example 4.3.1**

Input picture \(I(x, y)\) and mask \(M(x, y)\), as described in the previous section, are used in this normalization example. First, a representation is created were both \(I(x, y)\) and \(M(x, y)\) have the iris center point \((x_0, y_0)\) centered in the middle of the image. These pictures are defined as \(I_c(x_c, y_c)\) and \(M_c(x_c, y_c)\). The graphical representation of \(I_c(x_c, y_c)\) and \(M_c(x_c, y_c)\) are shown in respectively 4.6(a) and 4.6(b). Both pictures are unscaled in comparison to the original input image, and they have a resolution of 240x240 pixels.

These two pictures in the image domain, with cartesian coordinates \((x, y)\), are translated to a...
polar domain \((r, \theta)\) according to the following equations:

\[
I_e(x, y) = I(x - x_0, y - y_0) \\
x_c' = \frac{\alpha}{r^*} \cos \theta \\
y_c' = \frac{\alpha}{r^*} \sin \theta \\
\alpha = r_i - r_p
\] (4.3)

Variable \(\alpha\) is the difference between the iris radius \(r_i\) and pupil radius \(r_p\), that has to be normalized to a fixed length for every iris region. Angle \(\theta\) is between \([0, 2\pi]\), and the value for \(r^*\) defines the normalized radius of the iris region. The input picture \(I_e(x, y)\), transformed to the normalized polar domain will be

\[
I_n(r, \theta) = I_e(x_c', y_c')
\] (4.4)

The same transformation is also performed on the mask \(M_e(x, y)\) to achieve \(M_n(r, \theta)\). The constant normalized radius \(r^*\) should be equal to the average iris radius length after examination of all iris pictures available in the database. As an example, the iris radius region \(\alpha\) in this input picture has a length of 65 pixels and is normalized to 70 pixels. The angular resolution is chosen to become two times the picture width (480 pixels), i.e. the entire iris circle is divided into 480 radial segments and translated to the same amount of horizontal pixels.

As mentioned earlier in this section, the bounding circle of the iris mostly does not have the same center point as the pupil’s bounding circle. The transformation representation reflects this displacement as a non-constant radius of the pupil boundary. As a consequence of the non constant radius, the unrolled image shows a wavy formed pupil boundary, instead of a perfect straight line in case the pupil and iris circle boundaries were centered. Figures 4.7(a) and 4.7(b) show the final mask and iris region.

### 4.3.3 Feature Extraction

The next step is to extract the features from the iris pattern. The iris pattern mostly has a dominated radial structure. This radial structure is used as a discriminating feature between different
irises. This is an important reason why the cartesian to polar coordinate transformation, as explained in the previous section, is implemented in common iris recognition systems. In polar coordinates the radial iris structure is transformed to a vertical dominated pattern. The polar form as well, is more suitable for rotation invariant analysis.

Gabor functions share many properties with the human visual system in terms of biological perception of texture. Filters based on wavelets localize frequency content to a spatial temporal position. The Gabor wavelet is ideally suited to analyse the texture of an iris. A Gabor filter consists of a Gaussian function, modulated by a sine. A Gabor filter attenuates several spatial frequencies and enhances others, depending on the orientation of the filter. Daugman uses 2D Gabor wavelets to extract features from the iris pattern. A 2D Gabor filter is represented by the following formula in the image domain $(x, y)$ [23]:

$$G(x, y) = e^{-\pi [(x-x_0)^2/w_0^2 + (y-y_0)^2/\beta^2]} e^{-2\pi i [u_0(x-x_0) + v_0(y-y_0)]}$$

(4.5)

Where $(x_0, y_0)$ specifies the position in the image, $(\alpha, \beta)$ specifies the effective length, and $(u_0, v_0)$ indicates modulation with spatial frequency $\omega_0 = \sqrt{u_0^2 + v_0^2}$. Figure 4.8 depicts a Matlab implementation of a 2D Gabor filter in the cartesian domain $(x, y)$ with its corresponding real and complex part. Daugman suggests to filter the iris in the polar domain $(r, \phi)$ instead in the Cartesian domain $(x, y)$. Transforming formula 4.5 to the polar domain results in equation [22].

$$G(r, \theta) = e^{-i\omega(\theta-\theta_0)} e^{-r/r_0} e^{i\omega^2/2} e^{i(\theta-\theta_0)^2/\beta^2}$$

(4.6)

Figure 4.9 shows this filter in the polar coordinates domain. Daugman uses a number of filters (filterbanks) with different sizes, orientation and frequencies, to extract the discriminating features of the iris pattern.

### 4.3.4 Quantization

Filtering the image with the 2D Gabor filter, results in complex valued output coefficients. These coefficients reflect a phasor in the complex plain. Each phasor is mapped to a location in one of

---

(a) Unrolled and normalized iris region

(b) Unrolled and normalized mask region

Figure 4.7: Normalized iris and mask regions. The initial iris radius $r_i$ is stretched from 65 to 70 pixels in this example.
the four quadrants of the complex plane. The location is determined by the phase between the phasor's real part and its imaginary part, see figure 4.10. Every quadrant in the complex plain represents two binary bits, in such a way, that neighboring quadrants have only one bit different. This means that every complex valued coefficient is quantized to two bits. All quantized coefficients together form the binary feature vector of an iris. Daugman uses the phase information of the complex valued coefficients to find the discriminating features of an iris, because amplitude information only gives information about luminance and imaging contrast values. This phase demodulation can be denoted as [23]:

\[
h_{\text{Re,Im}} = sgn_{\text{Re,Im}} \int_r \int_\phi I(r, \phi)e^{-\omega(t_0-\phi)}e^{-(r_0-r)^2/\alpha^2}e^{(l_{\text{Re}}-\phi^2)/\beta} r \, dr \, d\phi
\]

Where \( I(r, \theta) \) is the raw iris image in a polar coordinate system, the parameters \( \alpha \) and \( \beta \) are the 2D wavelet size parameters. The wavelet frequency is specified by \( \omega \), and \( (r_0, \theta_0) \) represents the polar coordinates of the iris region for which the phasor coordinates \( h_{\text{Re,Im}} \) are calculated. \( h_{\text{Re,Im}} \) can be explained as a complex valued bit whose real and imaginary parts are either 1 or 0, depending on the sign of the two dimensional integral.
After filtering the unrolled image with 2D Gabor wavelets, the complex valued output coefficients specify a phasor in the complex plane. The angle of every phasor determines the location in the four quadrant complex plane and is subsequently quantized to two bits of phase information.

After phase demodulation of the complex valued coefficient, the resulting binary feature vector consists of \( L = 2048 \) bits. This means that 1024 complex valued coefficients are quantized to a value of two bits. An equal number of \( L \) mask bits are generated to signify which bits of the 256 bytes binary feature vector are valid, i.e. are part of the iris region detected during the iris localization step described in section 4.3.1. These two 2048 bits vectors, signifying the binary quantized phase information and the corresponding mask part that excludes those binary information bits which are invalid, together form the so called IrisCode:

\[
\text{IrisCode} = [I \ M],
\]  

(4.8)

with

\[
I \in \{0, 1\}^L \quad \text{and} \quad M \in \{0, 1\}^L.
\]  

(4.9)

The enrollment template \( X \in \mathcal{X} \) and verification template \( Y \in \mathcal{Y} \), belonging to a user, can be denoted as a composition of the vectors \( I \) and \( M \):

\[
X = [X_I \ X_M] \quad \quad Y = [Y_I \ Y_M].
\]  

(4.10)

Thus,

\[
X_I, Y_I \in \{0, 1\}^L \\
X_M, Y_M \in \{0, 1\}^L.
\]  

(4.11)

### 4.4 Circular zones

One should expect that every pixel in the normalized unrolled iris, illustrated in figure 4.7(a), will be quantized to two bits using formula 4.7. The normalized unrolled iris has an radius of 70 pixels (vertical) and 480 pixels in the angular (horizontal) direction in this example. Mapping every pixel to two bits will lead to 8400 bytes, which is significantly higher than the 256 bytes template length produced by Daugman’s algorithm. Probably Daugman performs some dimension reduction to reduce the number of components in his IrisCode. In [22] is mentioned that the localized iris structure is divided in eight circular zones of analysis as is illustrated in figure

---

**Figure 4.10**: After filtering the unrolled image with 2D Gabor wavelets, the complex valued output coefficients specify a phasor in the complex plane. The angle of every phasor determines the location in the four quadrant complex plane and is subsequently quantized to two bits of phase information.
4.11. Probably he divides every circular zone in segments as well. When we assume that the iris is divided in eight radial parts, we can calculate the number of segments for each circular zone, using the template length of 2048 bits. We know that every filtered polar coordinate \((r, \theta)\) is demodulated to two bits. This implies that \(2048/2 = 1024\) polar coordinates were demodulated. Having eight circular zones yields to \(1024/8 = 128\) segments for every circular zone. The assumption has been made after implementing the template matching algorithm described in section 4.5.2.

4.5 Matching of binary templates

In contrast to many other biometric systems, the iris system deploys a Hamming Distance classifier, in which also the mask sequence is incorporated:

\[
HD = \frac{\| (X_I \oplus Y_I) \cap X_M \cap Y_M \|}{\| X_M \cap Y_M \|} \tag{4.12}
\]

Where \(X_I\) and \(Y_I\) denotes the binary information vectors for enrollment and verification, respectively. \(X_M\) and \(Y_M\) denote the corresponding mask parts and \(\oplus\) is defined as XOR. The operation \((X_I \oplus Y_I)\) indicates the location of those components pairs that differ from each other when comparing both vectors entirely. The boolean operator \(\cap\) ensures that none of the information bits are corrupted by any kind of noise or derived from a location in the eye outside the iris region.

4.5.1 Compensation for rotational differences

The template generation process does not take into account the rotational difference between the subject’s eye and the camera when capturing the eye in different sessions. Two iris templates, belonging to the same person, could be falsely rejected when no compensation is made for these rotational differences. Because these binary templates are not rotation invariant, the binary templates \(X\) and \(Y\) are relatively shifted. Circular shifting the bits in a template, will virtually rotate the binary iris representation along a rotation angle. Every circular shift size of \(\alpha\) bits along the iris template, will presumably rotate to the same angle proportional to \(\alpha\). Thus, a single orientated iris template \(X\) can be compared with a second template \(Y\) in multiple orientations. Furthermore, shifting the template bits with a certain shift size to the left (negative shift), will...
specify a virtual anticlockwise rotation of the eye. A right shift (positive shift) shall virtually rotate the eye in a clockwise direction. When shifting the bits in a template with \( \alpha \) positions, both the bits in the information vector and mask vector will shift with \( \alpha \) positions in the same direction.

Let us denote \( \alpha \) as a variable, indicating the shift size of a shifted template \( Y \). When \( \alpha = 0 \), the verification template \( Y \) is not shifted, i.e. representing a non rotational compensated template. The sign of \( \alpha \) indicates in which direction the bits are shifted. A negative value for \( \alpha \) means that template \( Y \) is shifted with a shift size of \( \alpha \) bits to the left. On the contrary, a positive value for \( \alpha \) indicates that the bits in \( Y \) are shifted \( \alpha \) bits to the right.

A multiple orientated template \( Y \) consists of the template \( Y \), shifted according to different values for \( \alpha \). Let us specify vector \( A \), containing all different values for variable \( \alpha \).

\[
A = \{-\alpha_{\frac{N_s}{2}}, \ldots, -\alpha_1, \alpha_0, \alpha_1, \ldots, \alpha_{\frac{N_s}{2}}\}, \quad \alpha_0 = 0
\]

(4.13)

The cardinality of vector \( A \) becomes:

\[
|A| = N_s + 1
\]

(4.14)

which is the number of shifts step \( N_s \), plus the non rotated (original) template \( Y \) when \( \alpha = 0 \). One can see in (4.13) that a template \( Y \) will be shifted as many times to the left as to the right.

The following equation represents a set \( Y_{shift} \), containing all circular shifted versions of \( Y \) with vector \( A \):

\[
Y_{shift} = \text{Shift}(Y, A)
\]

(4.15)

Where,

\[
Y_{shift} = \{Y_{\text{shift}}^1, Y_{\text{shift}}^2, \ldots, Y_{\text{shift}}^{N_s+1}\}
\]

(4.16)

For every shift step \( \alpha \), one can calculate the Hamming Distance between two templates, resulting in \( N_s + 1 \) scores of similarity. The set \( HD \) contains the HD values after matching template \( X \) with all \( N_s + 1 \) shifted versions of \( Y \).

\[
HD = \text{HamDist}(X, Y_{shift})
\]

(4.17)

Where,

\[
HD = \{HD^1, HD^2, \ldots, HD^{N_s+1}\}
\]

(4.18)

For the match that produces the lowest HD value in set \( HD \), it is assumed that both templates \( X \) and \( Y \) were correctly aligned towards each other. Out of this set of HD values, the match resulting in the lowest HD score is chosen as the best match, represented as follows:

\[
HD_{\text{min}} = \min(HD)
\]

(4.19)

Because of the varying number of valid components in a binary template, the number of common valid mask bits, denoted by \( \|X_M \cap Y_M\| \), varies as well, when matching a template pair. This is illustrated in figure 4.12. The left picture represents a distribution of common valid mask bits, when matching two templates from the same individual. One could interpret this distribution
that on average approximately one third of all available components do match, i.e. they are used for a Hamming Distance calculation between a template pair, derived from the same iris. The distribution on the right side visualizes this for interclass template comparisons. The minimum number of common valid mask bits for a specific genuine template pair, is around 500 components. This is a significant difference in comparison to the template length of 2048 components. Furthermore, both pictures reflect that there is a relatively broad range in the number of common valid components for intraclass and interclass template comparisons.

![Figure 4.12](image)

Figure 4.12: These two figures depict the distribution of common valid mask pairs for aligned templates. Figure (a) shows this distribution for aligned intraclass templates, (b) for aligned interclass templates.

### 4.5.2 Deriving parameter values for matching

The previous section described in which way two binary iris templates should be matched. When comparing the templates $X$ and $Y$, the complexity for computing the minimum Hamming Distance is proportional to shift size $\alpha$. Especially if one chooses to shift the verification template $Y$ bitwise for every new HD calculation, e.g. $A = \{-1024, -1023, \ldots, 0, \ldots, 1023, 1024\}$. In this example a template $X$ will be matched along the entire template $Y$ in their highest resolution representation. This type of matching would result in an unfavorable load to the iris recognition system. This section describes how the possibility to reduce this load without losing accuracy is investigated. Before reducing the computational complexity, two properties are used:

1. We do not have to compensate along the complete iris, i.e. compensate for a maximum of $360^\circ$ rotational difference between two irises. Or described in terms of binary templates: shift the template bits within a full range of 2048 bits. A more likely situation would be that people rotate, during enrollment or verification, their heads only slightly sidewards to the left or right, while capturing an iris.

2. The binary iris templates available in the database, are created by dividing the eyes in eight zones of analysis. See figure 4.11 in section 4.4. So we assume that there is no need to compare the irises in their highest resolution. In a real environment, the templates are
probably matched in several steps. In every step the bits are shifted a fixed number of bit positions (shift size) in a certain direction.

Within which range a person normally rotates its head in front of an iris capturing device had to be found out first. Consequently, a suitable number of \( N_s \) shift steps, very step \( n \) representing a shift amount of \( k \) bits, were searched.

**Finding the angular range of sidewards head rotations** This section describes the method to find out the maximum angle of head rotations. The finally obtained angular range is based on templates stored in the database. As mentioned earlier, a full range search would be the case when \(-1024 \leq \alpha \leq 1024\). This implies a virtual sidewards rotation of the eye along \( \pm 180^\circ \).

We computed repetitively \( Y_{\text{shift}} = \text{Shift}(Y, A) \) for every compared template pair belonging to the intraclass. The initial range for various values for \( \alpha \) had to be sufficiently high. Several tests indicated that if one chooses a search range \( A = \{-90, -89, \ldots, 0, \ldots, 89, 90\} \), this range was suitable enough.

**Finding the number of matching steps \( N_s \) with shift size \( k \)** Finding suitable values for \( N_s \) and \( k \) is done in a straightforward method.

1. Using a \( \pm 90 \) bit search range for \( \alpha \) results, according to equation (4.17), to 181 different HD values for every matched pair. One of the 181 candidate HD values represents the presumably correct alignment of template \( X \) and \( Y \).

2. According to equation (4.19), the lowest value in set HD for every matched pair had to be retained. The minimum value in HD implies that \( X \) and \( Y \) presumably were correctly aligned templates.

3. A histogram of all collected values for \( \alpha \), in situations that \( \alpha \) corresponds to a selected \( H_{D_{\text{min}}} \) values had to be created. This plot, illustrated in 4.13, indicates which maximum shift size \( \alpha \) had to be carried out, in order to align all templates belonging to the intraclass.

One can see that the collected values for \( \alpha \) differ within intervals of 16 bits. The maximum value for \( \alpha \) is 64 shift positions to the right (positive shift). Mostly, there was no need to rotate the verification template \( Y \) in multiple directions in order to align it to \( X \). One can conclude this from the fact that the majority of collected values for \( \alpha \) are zero. Visible in this plot are also the number of \( n \) shift steps that are carried out while matching two templates. These \( N_s \) shift steps can also be explained as the \( N_s \) different orientations wherein a template \( Y \) is matched with template \( X \). There are \( N_s = 8 \) different orientations in total. Illustrated by the eight vertical bars in the plot at shift size values \( \alpha \in \{-48, -32, -16, 0, 16, 32, 48, 64\} \). Shift size -64 is finally added to this set in order to have a symmetrical method of matching, giving nine virtual rotation directions for \( Y \). According to the constant interval of 16 bits between different values for \( \alpha \), we can conclude that at every step the template is shifted along 16 bit positions. All further research work after this observation had been carried out using these two values for \( N_s = 9 \) and \( k = 16 \).

**4.6 Modeling the imposter distribution**

In this section, the same steps as described in [23], were used to model the imposter distribution after matching iris templates. First, the distribution is modeled when matching templates in a
Figure 4.13: Histogram of collected shift sizes. Every retained shift size indicates the difference in bit positions between two templates belonging to the intraclass, in order to align them.

single orientation. The probability \( p \) that two arbitrary components of the binary iris template disagree, is approximately \( p = 0.5 \). Because of this fact the expected mean Hamming Distance value after different pairings would also be circa \( HD = 0.5 \). One can see that each comparison between two components is in essence a Bernoulli trail as is described in [22] and [23]. Note that there is dependency between the components (Bernoulli trail), as Daugman mentioned in these publications that there is correlation between components in a binary iris template.

The binomial distribution function is defined as:

\[
P(X = w) = f(w|N, p) = \frac{N!}{w!(N - w)!} p^w (1 - p)^{(1-w)}
\] (4.20)

The formula can be understood as the probability that a random variable \( X \) has exactly \( w \) successes after \( N \geq 0 \) trails. The success probability is \( 0 \leq p \leq 1 \). The parameter \( w \) is in this case the number of disagreeing bits, when comparing two binary iris templates.

In order to create a binomial distribution function, the parameter \( p \) and \( N \) have to be known. We presume that the probability \( p \) that an arbitrary component number \( x \) is set along a binary iris template is:

\[
p = \mathbb{P}(I(x) = 1|M(x) = 1) = 0.5 \quad , \quad (1 \leq x \leq L)
\] (4.21)

Knowing the variance \( \sigma^2 \) of the binomial distribution function, we can calculate \( N \):

\[
\sigma^2 = Np(1 - p)
\]

\[
N = \frac{p(1 - p)}{\sigma^2}
\] (4.22)

One can see \( N \) as a value that indicates that the equivalent of about \( N \) independent binary degrees of freedom typically remaining in a 2048 bits binary template exist. In [22] is suggested that there a lot of dependency between binary iris templates. Daugman stated in [22] that this
correlation is introduced by the 2D Gabor filter used to extract the features of an iris. Another form of dependency is the correlation inherently present in irises. The probability of two irises from different people agreeing is roughly stated at $1 : 2^N$.

Modeling the imposter distribution after matching templates in multiple orientations, makes use of the single orientation imposter distribution function. Let function $f_0(t)$ represent the binomial function defined in (4.20). This implies that $f_0(t)$ denotes the distribution function of collected HD values between template pairs belonging to the interclass, compared in a single orientation. Function $F_0(t)$ is the cumulative of $f_0(t)$ from 0 to $t$:

$$F_0(t) = \int_0^t f_0(\tau) d\tau = \sum_{i=0}^{t} \frac{N!}{i!(N-i)!} p^i (1-p)^{(N-i)}$$

(4.23)

Equivalently, $f_0(t)$ can be denoted as:

$$f_0(t) = \frac{d}{dt} F_0(t) = \frac{N!}{t!(N-t)!} p^t (1-p)^{(N-t)}$$

(4.24)

Function $F_0(t)$ signifies the probability of making a false match between two templates, compared in a single orientation, when decision threshold $t$ is applied. The probability of not making a false match between two different irises, is denoted as:

$$1 - F_0(t)$$

(4.25)

Given $N_S$ number of such independent tests, gives the probability of not making a false match between two different irises at $N_S$ different orientations, in combination with threshold criterium $t$:

$$F_{N_S}(t) = 1 - [1 - F_0(t)]^{N_S}$$

(4.26)

The associated probability density function $f_{N_S}(t)$ is specified as:

$$f_{N_S}(t) = \frac{d}{dt} F_{N_S}(t) = N_S f_0(t)[1 - F_0(t)]^{N_S-1}$$

(4.27)

This density function signifies an approximation for the imposter distribution.
Section 5

Implementation of Iris Recognition System

This chapter discusses the implementation of Daugman's method of comparing and authenticating binary iris templates without applying template protection. Publications [22] and [23] describe Daugman's method of authenticating people's irises. Especially with regard to rotation alignment of the iris templates, no exact and practical method to deal with this problem is described in these publications. However, the authentication system without template protection should function as a fundamental system and basic knowledge to develop an iris recognition system with template protection. Therefore, it is important to approach Daugman's method for iris authentication as correct as possible. In his publications Daugman provided several results after comparing a large amount of iris templates. The results obtained after simulating the authentication system, were compared with the published statistics. A significant difference in the achieved results compared to Daugman's results will most likely indicate a wrong approach. Also the simulation results for the unprotected iris authentication system can be compared with the template protected system, in order to conclude if there is performance loss. To test the system performance, the data set described in section 5.2 was used.

The last section in this chapter compares the results of the implemented method and the results described in Daugman's papers. The other sections explain results necessary to be able to verify if the implemented approach is correct.

5.1 Synthetic database

We generated a synthetic data set for testing purposes. It contains binary templates according to statistical data, provided in [23], with mean $\mu_{HD} = 0.11$ and $\sigma_{HD} = 0.065$ for the genuine distribution. In this setup, the mask vector to compute the fractional Hamming Distance, see equation (4.12), is not used. For that reason, the Hamming Distance is given by

$$HD = \frac{\|X_t \oplus Y_t\|}{L}$$

Every individual in the synthetic database has $N_Y = 5$ verification and $N_X = 1$ enrollment templates. The database consists of fifty users. The genuine and imposter distribution are shown left in figure 5.1. Both distributions are normal distributions, as the bit errors between two genuine templates are determined by gaussian noise. The right figure illustrates the FRR and
Figure 5.1: Performance results for the synthetic data set with $\mu_{HD} = 0.11$ and $\sigma_{HD} = 0.065$. Figure (a) depicts the genuine and imposter distribution, (b) the FRR and FAR after simulating the intra and interclass variation.

FAR. One can see that the FRR is zero at a maximum number of 424 bit errors ($HD = 0.29$) between two genuine templates. This is approximately the same as the maximum Hamming Distance $HD = 0.34$ for the genuine distribution mentioned in [23]. The imposter distribution has a mean of 1024 bits ($HD = 0.5$), indicating that bit pairs between two imposter templates do disagree with a probability of approximately $p = 0.5$. In the remaining part of this chapter, the Hamming Distance is calculated using equation (4.12).

5.2 The CASIA data set

In this section we introduce the data set used in our experiments. Furthermore, we study the statistical properties of the data set. The binary templates are derived from iris images available in the CASIA database [31]. It contains iris images of 108 people, where every person has in total seven iris images available. These iris images were captured in two different sessions, with a time interval of one month. Using the Daugman algorithm, described in the previous sections, most of the images were translated into binary feature vectors. Images with insufficient quality were omitted. A total of 504 iris images had been transformed to binary templates. People that only had one template available were deleted from the database. For statistical reasons, we needed more than one measurement per person. Therefore, these templates were also not used in the evaluations. The resulting data set was limited to 93 people, and for every individual at least two templates were available. Table 5.1 shows the number of users $N_u$ that have at least $N_{XY}$ binary templates available.

In order to artificially increase the number of template verifications, the original enrolled templates and verification templates can be swapped. Every permutation of templates will create a new set (split) of enrolled and verification templates and hence more reliable statistics. The
These images, taken from the CASIA database, illustrate some examples of unfavorable pictures. Figure 5.2(a) indicates a rotated eye, 5.2(b) bad focus and 5.2(c) extreme iris pattern occlusion.

\[
N_{mrSplits} = \binom{N_{XY}}{N_X} = \frac{N_{XY}!}{N_X!N_Y!}
\]

(5.2)

\[N_X\] denotes the number of enrolled templates, and \(N_Y\) the number of verification templates available for a user and \(N_X + N_Y = N_{XY}\). In our experiments, shown later in this report, we use three different experimental setups indicated with A, B and C, see table 5.2. The number of matches between template pairs belonging to the intraclass in the different database settings, becomes \(N_U \cdot N_X \cdot N_Y \cdot \binom{N_{XY}}{N_X}\) and for the interclass it holds \(N_U \cdot (N_U - 1) \cdot N_X \cdot N_Y \cdot \binom{N_{XY}}{N_X}\) comparisons.

Figure 5.3 depicts that on average, in each template, less then half of the total available amount of 2048 components are indicated as valid components. On average, 1090 bits of a binary template are valid. In [23] it is suggested that there is a lot of correlation between components by comparing the iris template length \(L\) with the derived degrees of freedom \(N\). This is probably not an assumption which is always true. The figure shows that none of the templates derived
from the CASIA database has 2048 valid components. Comparing the average amount of valid components (1090) in a binary template probably gives a better understanding about the degree of correlation.

![Histogram of number of valid components](image)

**Figure 5.3:** Distribution of the number of valid components in each template. This histogram is derived after observing all in the database available templates.

### 5.2.1 Rotation invariant approach

Although in the final system there could be a rotational misalignment between enrollment and authentication template, we ignored this at first. This means that in each authentication, only one HD is evaluated. The resulting intra and inter-class distribution and corresponding performance curves are given in figure 5.4. The imposter distribution has \( \mu_{HD_{imp}} = 0.5 \) and \( \sigma_{HD_{imp}}^2 = 6.76 \times 10^{-4} \). With the use of these parameter values we can approximate this distribution as described in section 4.6, resulting in \( N = 370 \) degrees of freedom. The relatively high amount of overlap between both distributions is caused by the variance in rotation between the original iris images. Hence the corresponding EER is only 7.2%.

The CASIA data set does not contain information on rotational alignment. Therefore, the templates are matched according to the algorithms explained in section 4.5.1 and 4.5.2. The statistical behavior when incorporating the rotation compensation is shown in figure 5.4, using the CASIA data set in configuration A. A template \( X \) is compared to 9 rotational directions for verification template \( Y \). These rotational directions are obtained by shifting the bits of verification binary template \( Y \). The 9 different shift sizes are specified in set \( A \). Set \( A \) is denoted by \( A = \{-64, -48, -32, -16, 0, 16, 32, 48, 64\} \).

In Figure 5.5(a) can be seen that both intra and interclass distributions do not significantly overlap each other. In other words, both the intra and interclass are separable. Figure 5.5(b) illustrates the FRR and FAR. For this configuration, the EER approximates 0%, which means no genuine person is falsely rejected and no imposter is falsely accepted. The solid line indicates a function that approximates the genuine distribution. In this case there is no known distribution function...
Figure 5.4: Performance results for the CASIA data set in setup A without rotation compensation. Figure (a) shows the genuine and imposter distribution, (b) the FRR and FAR.

that models the genuine distribution.

Also the imposter distribution becomes narrower when compensating for rational differences. The mean HD value of the imposter distribution is reduced. The approximation of the imposter distribution is modeled according to the theory explained in section 4.6 and represented in figure 5.5(a) with a dashed line. In this case, the probability density function $f_0(x)$ and accumulative function $F_0(x)$ are determined using the parameter values $N = 370$ and $p = 0.5$. However, filling these values for $N$, $p$ and $N_S = 9$ in equation 4.27, does not give a satisfying approximation for $f_{N_S}(t)$. This probably has to do with the fact that there are less (just two) retained minimum HD values, in the situation were $\alpha = \pm 64$. This observation is also visualized in figure 4.13. The plotted approximation for $f_{N_S}(x)$ in figure 5.5(a), is according to $N_S = 7$, i.e. neglecting the retained orientations were $\alpha = \pm 64$.

5.3 Statistics of system accuracy

In this section we summarize the statistical results for both the rotation compensated and non-compensated system as described in the previous section. The statistics obtained after simulating the system performance using the CASIA database, are provided in table 5.3. In order to have an indication if a (presumably) correct method for template matching is implemented, especially with regards to the rotation invariant system, the simulation results are compared to statistics provided by Daugman. These statistics [23], presented in the right part of the table, were gained in several tests$^1$. One can conclude, after comparing both statistical results, that the implemented approach on ration alignment is probably correct, as there is a minor difference in the results.

$^1$Daugman compared 7070 different pairs of same-iris templates and 9.1 million templates, belonging to different irises.
Figure 5.5: Performance results for the CASIA data set in setup A with rotation compensation. Figure (a) shows the genuine and imposter distribution, (b) the FRR and FAR.

Table 5.3: This table indicates the statistics after simulating the performance with the CASIA database for either the rotation uncompensated as the compensated system. Furthermore statistics on rotation compensation provided in [23] are shown (Daugman section).
Section 6

Iris Recognition System with Template Protection

In this chapter we discuss how the template protection system based on helper data, explained in chapter 3, can be used in an iris recognition system such as described in the previous section.

6.1 General architecture

This section describes the architecture when both schemes are combined, see figure 6.1. The left and right part of the figure indicate the enrollment and verification phase, respectively. In the upper part of this figure, one can see the standard unprotected iris recognition system. The unprotected iris recognition system with rotation compensation as described in section 4.5.1, has been taken as a reference model to explore the feasibility of the iris template protection scheme. The output of this system is a feature vector $X_I$ and a mask vector $X_M$. These vectors are used as input data for the template protection system.

For all systems it holds that the combined biometric authentication scheme and theoretical template protection model only can have a satisfying performance, when the noise difference between an enrollment template $X$ and verification template $Y$ is within certain limits. In principle the template protection model adds redundant information to template $X$ and in the verification

© Koninklijke Philips Electronics N.V. 2006
phase, it tries to correct all errors due to the noise difference between both templates. The noise difference between iris templates might be caused due to:

- Covering parts of the iris region, i.e. occluding eyelashes and eyelids.
- Different light conditions between captured iris images.
- Hard contact lenses that give boundary artifacts in the iris region.
- Specular reflections. While capturing the eye in the enrollment or verification phase, the eye is illuminated with a light source. This can cause light reflections visible in the iris region.
- Normalization of the iris region. As illustrated in 6.2, the iris pattern changes when different light intensities are used in either the enrollment or verification session. Although the iris is normalized, stretching the iris pattern with Daugman’s rubber sheet model might give an inappropriate normalized representation.

![Figure 6.2: Two similar eyes where both surfaces of the iris region differ.](image)

### 6.1.1 Challenges

In case there is no noise difference between the enrollment and verification templates, only hashing the biometric measurements during both session would be feasible to ensure a person’s privacy. In case the hashed biometrics are the same, a positive match is registered. Obviously, it is very likely that noise influences a biometric measurement, resulting in different templates during enrollment and verification and this straightforward method is not possible. Moreover, different iris biometrics $X$ and $Y$ would give different hashed outputs. This means that with high probability the two hashed biometrics will not match.

Furthermore, there are some significant differences between the unprotected iris authentication system and other biometric systems, i.e. fingerprints and face recognition. These differences also induce different approaches and solutions for the finally designed iris template protection system. There is a mask bit assigned to every component in an iris template, indicating if the corresponding bit may be used in authentication. Integrating mask vectors into the template protection system is not a straightforward task. The enrolled information feature vector components are scrambled and encoded, therefore mask vectors cannot be used in a similar way as in equation (4.12). A simple approach would be to ignore the mask vectors while matching a template pair, giving

$$H_D = \frac{\|X_I \oplus Y_I\|}{2048} \quad (6.1)$$
This means that possible errors due to noise are not indicated by mask vectors. The impact on the Hamming Distance scores when using this classifier is depicted in figure 6.3. One can see that the performance has become worse compared to figure 5.5, which reflects the system performance of the unprotected iris recognition. Matching templates without the mask vectors gives raise to the FRR and FAR, because the system has to manage more matching bit errors. For the intraclass it holds $\mu_{HD} = 0.21$ and for the interclass $\mu_{HD} = 0.40$. The EER = 4.2% at 619 bit errors. This implies a significantly decreased class separability. The ECC should correct at least 0.3% of the bit errors to achieve the EER. Therefore, one of the major challenges for the iris template protection system is to incorporate the mask vectors in the error correction scheme.

![Figure 6.3](image)

(a) Genuine and imposter distribution  
(b) FRR and FAR

Figure 6.3: Performance when matching templates without using the mask vectors.

Moreover, the template protection scheme adapted to iris data should also deal with possible rotational differences between iris templates in authentication.

### 6.2 Template protection for iris systems

In this section, we address the challenges raised in the previous section. Figure 6.4 gives a detailed schematic. The helper data $W_1$ and $W_2$ are stored in the database. A one-way hashed version of a random generated secret $S$ that refers to the user, is also stored in the database. At verification, a set of candidate secrets is derived, subsequently hashed and compared to the stored hashed secret. The variables used in this scheme are specified as:

- $X_I \in \{0,1\}^L$, $X_{rel} \in \{0,1\}^L$, $X_{All} \in \{0,1\}^L$, $C, C' \in \{0,1\}^N$, $W_1 \in \mathbb{N}^N$,
- $X_M \in \{0,1\}^L$, $X_{Mrel} \in \{0,1\}^L$, $X_{Mal} \in \{0,1\}^L$, $S, S' \in \{0,1\}^K$,
- $X = [X_I \; X_M]$, $X_{Mrel} = [X_{rel} \; X_{Mrel}]$, $X_{Mal} = [X_{All} \; X_{Mal}]$,
- $Y_I \in \{0,1\}^L$, $Y_{shift} \in \{0,1\}^L$, $W_I \in \mathbb{N}^N$,
- $Y_M \in \{0,1\}^L$, $Y_{Mshift} \in \{0,1\}^L$, $W_2 \in \{0,1\}^N$,
- $Y = [Y_I \; Y_M]$, $Y_{shift} = [Y_{shift} \; Y_{Mshift}]$, $Z, Z' \in \{0,1\}^N$,

The following scenario’s for enrollment and verification of user $m$ explains the functionality of the scheme in more detail:

© Koninklijke Philips Electronics N.V. 2006
Figure 6.4: Detailed schematic of iris template protection scheme. The dashed block W on the left side (enrollment) creates helper data W₁ and W₂. Subsection R contains processes for rotation alignment, finding reliable components and reliable component subset selection. On right side (verification), block S' generates candidate secrets. Subsection R' contains functional blocks in order to shift the verification templates and composition of the vectors Z'.

Enrollment

1. From the individual, a number of $N_X$ measurements from the same iris are made. The features are extracted and quantized. This gives an enrollment set $X$. As every binary feature vector has an information and mask vector, set $X$ is divided in $X_{I,m} = \{X_{I,m}^1, X_{I,m}^2, \ldots, X_{I,m}^{N_X}\}$ containing the information vectors and set $X_{M,m} = \{X_{M,m}^1, X_{M,m}^2, \ldots, X_{M,m}^{N_X}\}$ that contains the mask vectors.

2. Helper data $W₁$ is derived from the enrollment sets $X_{I,m}$ and $X_{M,m}$, and stored in the database. Helper data $W₁$ contains the indices of those components in the enrollment set that are reliable. How reliable components are estimated will be explained in section 7.2.

3. Vector $Z$ is composed of those reliable component bit values indicated by helper data $W₁$.

4. A secret $S$ is randomly generated and used as a reference to user $m$.

5. An error correction code (ECC) encodes secret $S$ to code word $C$. Redundant information is added to secret $S$, because this enables error correction during verification. The number of errors that can be corrected is restricted and is determined by the error correction capability that is related to the configured ECC.

6. Code word $C$ is added to $Z$ by means of an Exor operation. This is specified as $W₂ = C \oplus Z$. The resulting vector $W₂$ is stored in the database.

7. The secret $S$ is hashed and stored in the database. Since its is cryptographically scrambled, $h(S)$ cannot be traced to its origin. Therefore, privacy is ensured.

In enrollment step 5, it is mentioned that secret $S$ is encoded by means of an error correction code. The BCH error correction code is used in this thesis. Appendix A provides a detailed specification of the BCH code. In the list of the variables used in figure 6.4, one can see that the secret $S$ has a length of $K$ bits. Table A.1 reflects the code word length $N$, message length $K$.
and error correction capability of \( T \) bits. Obviously, the greater the size of the secret \( S \), the more secure the system is. A secret size of \( K = \pm 60 \) bits offers privacy protection to the database user. This implies a code rate of \( N/K \) for the template protection system. An examination of the BCH table indicates that the more errors the system has to correct, specified by \( T \), the shorter the corresponding secret size \( K \) should be to enable correct decoding of the guessed secret \( S' \).

**Verification** During the verification session, for each user \( m \) the scenario is as follows:

1. The biometric of the user \( m \) is measured, the features are extracted and quantized into two verification vectors \( Y_m = [Y_{1m}, Y_{M_m}] \), where \( Y_{1m} \) is equal to the feature vector and \( Y_{M_m} \) the mask.

2. Helper data \( W_1 \) extracted from the claimed identity is selected from the database. With use of helper data \( W_1 \), a set of nine vectors \( Z' = \{Z'_n\}_{n=1}^9 \) and \( P = \{P'_n\}_{n=1}^9 \) is created. Helper data \( W_1 \) selects for every rotation compensated \( Y_{1m} \) and \( Y_{M_m} \), the components that are specified in \( W_1 \). In essence, every version of the virtual rotation compensated template \( Y_{1m} \) corresponds to a \( Z'_n \). The accompanying mask vector \( Y_{M_m} \) corresponds to a \( P'_n \). This procedure will be explained in more detail in section 6.4.

3. Helper data \( W_2 \) of user \( m \) is selected from the database and added individually to every vector in \( Z' \), by means of an XOR function \( \oplus \), giving: \( C' = \{C'_n\}_{n=1}^9 = W_2 \oplus \{Z'_n\}_{n=1}^9 \).

4. Every vector in the set \( C' \) is decoded individually to a candidate secret \( S'_n \), giving nine candidate secrets \( S' = \{S'_n\}_{n=1}^9 \).

5. All candidate secrets \( S' \) are individually hashed, denoted as \( h(S') = h(\{S'_n\}_{n=1}^9) \). The stored \( h(S) \), which refers to user \( m \) is compared to the nine candidate one way hashed secrets \( h(S') \). The system declares a positive match if and only if there is at least one hashed secret \( h(S'_n) \) in the set \( h(S') \) similar to \( h(S) \).

6. In case of a positive match, the system decides that templates \( X \) and \( Y \) have been derived from the same iris.

The algorithm described in this section is reflected in figure 6.5. In this figure, the dashed circles define all distinct code word regions used in the error correction code. The boundary of the coding regions specifies the error correction capability. The biometric measurement \( X \) of individual \( m \) made during enrollment, illustrated as the dashed vector in the figure, refers by means of helper data \( W \) to a unique encoded secret. During verification of the same user \( m \), a binary template \( Y \) is generated of the iris and shifted along multiple shift sizes, inducing multiple virtual directions, denoted by the vectors \( \{Y_1, Y_2, \ldots, Y_{N_m=9}\} \). The same helper data \( W \) that was derived during enrollment is used to retrieve the code region specified for user \( m \). Every combination of helper data \( W \) and \( Y \) assembles a codeword. Only in case the combination of vector helper data \( W \) and one of the vectors \( Y_n \) derived for user \( m \) coincides in the same coding region as the encoded secret of user \( m \), this individual is accepted. The vector \( Y_n \) that together with \( W \) refers to the encode secret, can be interpreted as an rotation compensated (aligned) pair \( X \) and \( Y \).
6.3 Classification in Hamming Distances

The degree of similarity between an enrollment template and a verification template in the unprotected iris recognition system, is computed according to formula (4.12), which results in a Hamming Distance. In the iris template protection system as well, a calculation for the degree of similarity between an enrollment and verification template should be made, to be able to analyze the system performance. It is probably not possible to adopt the HD calculation between two templates within the iris template protection system. This section explains why.

Figure 6.6 reflects the fundamental part of the template protection system that is shown in figure 3.1. We left out the one-way hash function, as it in essence does not contribute to the decision criterium. The dashed arrows in the figure, indicate that we basically compare none hashed secrets in order to decide whether there is a positive or negative match. In more detail:
During verification of a person \( m \), code word \( C' \) is constructed as

\[
C' = W_2 \oplus Z' \tag{6.2}
\]

Helper data \( W_2 \) is specified by an Exor function between codeword \( C \) and \( Z \), giving

\[
W_2 = C \oplus Z \tag{6.3}
\]

During verification, the derived vector \( Z' \) can be modeled as a noised version of \( Z \). Thus,

\[
Z' = Z \oplus \mathcal{N} \tag{6.4}
\]

When we substitute vector \( Z' \) from equation (6.4) and helper data \( W_2 \) specified (6.3) into equation (6.2), codeword \( C' \) can also be specified as:

\[
C' = C \oplus Z \oplus Z \oplus \mathcal{N} = C \oplus N \tag{6.5}
\]

The decoding codeword \( C' \) will give the guessed secret \( S' \):

\[
S' = \text{DEC}(C') = \text{DEC}(C \oplus N) \tag{6.6}
\]

A person is accepted \( (S = S') \), whenever the noise difference \( \mathcal{N} \) between \( C \) and \( C' \) is below the ECC error correction capacity, i.e. \( \mathcal{N} \leq T \).

### 6.3.1 Bit Error Rate

To analyze the system performance of the template protection system, observing the number of bit errors between \( C \) and \( C' \) could give an indication about the system’s FAR and FRR as a function of the bit error rate (BER). Moreover, the noise factor \( \mathcal{N} \) in equation (6.6) is equal to the noise factor in equation (6.4). This implies that observing the amount of bit errors between \( Z \) and \( Z' \) is also sufficient to analyse the system performance, as the bit errors between codewords \( C \) and \( C' \) is in fact introduced due to the noise difference between \( Z \) and \( Z' \). This defines the BER for the template protection system:

\[
\text{BER} = \frac{\|C \oplus C'\|_N}{N} = \frac{\|Z \oplus Z'\|_N}{N} = \frac{\|N\|}{N} \tag{6.7}
\]

\( N \) is the codeword length of the BCH error correction code. However, in the template protection scheme, the information vector \( Z \) is Exored with \( C \), to ensure that impersonation is impossible. This means that real bit values of vector \( Z \) are unknown during the verification session.

### 6.3.2 Hamming Distance

Applying Daugman’s HD computation used to analyse the degree of similarity between a pair of iris templates as specified in formula (4.12) on the reliable component vectors \( Z \) and \( Z' \), would result in the following equation

\[
\text{HD} = \frac{\|(Z \oplus Z') \cap P \cap P'\|}{\|P \cap P'\|} \tag{6.8}
\]

Vector \( P \) contains the mask bits of the reliable components derived in enrollment set \( X \). Vector \( P' \) contains the validity of the selected components in \( Y \), specified by helper data \( W_1 \). Although
not shown in figure 6.6, both vectors $P$ and $P'$ could be known at the verification phase. One can retrieve $P$ in verification, since the indices defined in $W_1$ refer to valid reliable components during enrollment. In equation (6.8) is visible that the information vector $Z$ at the verification session should be known to enable a HD computation. Since helper data $W_1$ and $W_2$ and $h(S)$ are stored, the vector $Z$ is not available during verification. Therefore, HD computation between $Z$ and $Z'$ using both vectors $P$ and $P'$ is not possible. In other words computing (6.8) does not give an indication about the amount of bit errors between codeword $C$ and $C'$, which is in essence the decisive factor for a positive or negative match.

The following example explains this more.

**Example**

**Enrollment** Enrollment set $X$ ensues from several measurements of person $m$, giving $N_x$ iris templates $X$. We assume that all enrollment templates are aligned. A template $X = [X_I \quad X_M]$ for example has the following 12 bits information and mask templates:

$$X_I = [1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0]$$

$$X_M = [1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0]$$

Suppose in $X$, the components numbers $\{1,2,4,6,7,9,10,12\}$ are judged as being reliable components. The indices of the reliable components are stored in the database as helper data $W_1 = [1 \quad 2 \quad 4 \quad 6 \quad 7 \quad 9 \quad 10 \quad 12]$

The reliable information components in template $X$ compose vector $Z$

$$Z = [1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0]$$

Vector $P$, containing the reliable mask components of $X$

$$P = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$$

These are all valid components (set), as they all comply with the reliable component selection criteria. A more detailed explanation of how reliable components are estimated, is explained in section 7.2.

**Verification** During verification of the same user $m$, template $Y = [Y_I \quad Y_M]$ is generated. We assume that the verification template $Y$ is aligned with the enrolled template $X$. The measured information and mask vectors are for example

$$Y_I = [1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1]$$

$$Y_M = [1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1]$$

Helper data $W_1$ is taken from the database and the system selects the components indicated by $W_1$ in $Y$. Then vector $Z'$, with selected information components, should be

$$Z' = [1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1]$$

Vector $P'$ containing mask components becomes

$$P' = [1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$$
When we apply equation (6.7) on the mentioned scenario above, we have the following computation for the BER:

\[
\text{BER} = \frac{\left\| [0 0 1 0 0 0 1 1] \right\|}{8} = \frac{3}{8} = 0.375
\]

When we compute the Hamming Distance between the reliable component vectors using equation (6.8) gives for \((Z \oplus Z') = [0 0 1 0 0 0 1 1]\) and \(P \cap P' = [1 1 0 1 1 1 1 1]\).

Thus,

\[
\text{HD} = \frac{\left\| [0 0 0 0 0 0 1 1] \right\|}{\left\| [1 1 0 1 1 1 1 1] \right\|} = \frac{2}{7} = 0.29
\]

Clearly, the results for the HD computation and BER are different: use of the mask information guarantees a better performance. An attempt to exploit the fact that both mask vectors \(P\) and \(P'\) are available during verification of an individual and give rise to the system performance is explained in section 6.6.

### 6.4 Candidate secrets

In section 6.2 we explain that during verification of an individual, nine candidate secrets are generated. In an unprotected rotation invariant system, a verification template \(Y\) is matched in different orientations against template \(X\). The number of different rotations\(^1\) \(N_S\) and the range in which people rotate their heads in terms of bit shifts, is adopted from the unprotected iris recognition system with rotation compensation, as described in section 4.5.2.

This section explains in more detail how the iris template protection system deals with rotational differences between irises in enrollment and verification of an individual. Figure 6.7 reflects the

![Figure 6.7: Simplified version of template protection scheme for rotation invariant iris recognition](image)

fundamental part of the template protection system that is shown in figure 6.4. We leave out the one-way hash functions as they in essence do not contribute to the decision criterium. The dashed lines in the figure indicates that we basically compare non hashed secrets in order to

---

\(^1\)In section 4.5.2 is explained how the number of virtual iris rotations \(N_S = 9\) and bit shifts (maximally 64 bits toward left and right) was being retrieved.
decide whether there is a positive or negative match. The decisive factor for registering a positive or negative match is whether the amount of noise between \( Z \) and \( Z' \) is below the error correction capability of the used ECC.

The following example gives an enrollment and verification scenario for the rotation invariant iris template protection system.

**Example**

**Enrollment** For user \( m \), in the same way as described in the example in section 6.3, vector \( Z \) and helper data \( W_1 \) are derived. Vector \( W_1 \) is stored in the database together with the random generated secret \( S \). After BCH encoding of secret \( S \) to code word \( C \) with a length of \( N \) bits, helper data \( W_2 \) is given by

\[
W_2 = C \oplus Z
\]

(6.9)

Helper data \( W_2 \) is also stored in the database.

**Verification** For the same user \( m \):

1. Template \( Y = [Y_I \ Y_M] \) is generated after measuring the iris.

2. A set of shifted versions of template \( Y \) is created according to \( Y_{\text{shift}} = [Y_{\text{shift}}, Y_{\text{shift}} \ldots Y_{\text{shift}}]_n \) = Shift(\( Y \), \( A \)), where \( A \) is a set of different values \( \alpha \). Each value for \( \alpha \) specifies the shiftsize of a binary template. Vector \( Y_{\text{shift}} \) also has an information and mask part, according to

\[
Y_{\text{shift}} = [Y_{\text{shift}} \ Y_{\text{mask}}]
\]

The different values for \( \alpha \) are \( A = \{-64, -48, -32, -16, 0, 16, 32, 48, 64 \} \).

3. In every template \( Y_{\text{shift}} \), the components specified by \( W_1 \) are selected. Helper data \( W_1 \) selects the components in either \( Y_{\text{shift}} \) giving \( Z'_n \) and \( Y_{\text{mask}} \) resulting in \( P'_n \). In other words: the locations of the reliable components, determined at the enrollment phase, are again selected during authentication in different virtual orientations of the iris, giving nine different shifted versions vector \( Z'_n \) and shifted mask \( P'_n \). All vectors \( Z'_n \) are assembled in set \( Z' \). The corresponding mask vectors \( P'_n \) in \( P' \). We know that an iris does not rotate more than 64 bit shifts towards the left or right in steps of 16 bits. This implies that there is at least one vector \( Z'_n \) in the set \( Z' \), which is aligned with \( Z \), when \( Z \) and \( Z' \) are derived from the same iris. Every vector \( Z' \) in the set \( Z' \) can be modeled as the vector \( Z \) derived during enrollment added with noise \( N'_n \). In binary vector notation \( Z'_n = Z \oplus N'_n \). This noise factor \( N'_n \) is different for every rotation step. It might become even higher in value than the noise factor between a pair of \( Z \) and \( Z' \) derived from two different iris. This is illustrated in figure 5.4(a) as the genuine distribution goes beyond the imposter distribution when no rotation alignment is performed.

4. Each vector \( Z'_n \) is added to vector \( W_2 \). This addition results in a code word \( C'_n \), specified as \( C'_n = W_2 \oplus Z'_n \). Suppose every vector \( Z'_n \) represents a row vector in array \( Z' \). Then the set of code words \( C' \) becomes

\[
C' = W_2 \oplus Z' = \begin{bmatrix}
C \oplus Z \oplus Z \oplus N_1 \\
\vdots \\
C \oplus Z \oplus Z \oplus N_n \\
\vdots \\
C \oplus Z \oplus Z \oplus N_{N_5}
\end{bmatrix} = \begin{bmatrix}
C \oplus N_1 \\
\vdots \\
C \oplus N_n \\
\vdots \\
C \oplus N_{N_5}
\end{bmatrix}
\]

(6.10)
5. Every row in array $C'$ is decoded, which results in a candidate secret $S'_n$. This is defined as

$$\text{DEC}(C') = S' = \begin{bmatrix} S'_1 \\ \vdots \\ S'_{2} \\ \vdots \\ S'_{N_{S}} \end{bmatrix}$$ (6.11)

6. One out of these nine candidate secrets refers to a correctly compensated rotational difference between both captured irises, during enrollment and the authentication phase. If there is at least one vector $Z'_n$ in the set $Z'$ from which the number of disagreeing bits compared with $Z$ is less than the error correction capability, individual $m$ would be correctly authenticated, i.e. there is $N'_m \leq T$. How this affects the FRR and FAR is explained in section 6.5.

### 6.5 System performance analysis

This section describes how the performance of the iris template protection system was analyzed. In section 6.3 we wrote that by making use of equation (6.7), the BER between binary vectors $Z$ and $Z'$ belonging to intra and interclass is calculated. In every authentication attempt, the system calculates nine times a BER value between $Z$ and $Z'$, as nine different variants of $Z'$ were produced. Collecting all scores of similarity after simulating the intra and interclass variance, gives the typical FRR and FAR curves shown in figure 6.8(a). The EER equals almost 50%, which corresponds to random rejection and acceptance of templates. However, there is always one vector $Z'_n$ in the set $Z'$, which corresponds to a correct alignment with vector $Z$. The aligned pair $Z$ and $Z'_n$ is found by selecting the minimum BER between $Z$ and all vectors in $Z'$. When collecting only the minimum BER scores, the plots will not be disturbed by superfluous BER values, when simulating the intra and interclass variance. Figure 6.8 illustrates the FAR and FAR in both situations. In this simulation vector $Z$ is composed after selecting all reliable components that are available in the enrollment set $X$ for every user $m$, i.e. $N=2048$. All following BER distributions in this thesis, FRR and FAR plots are accomplished after selecting the minimum BER value between $Z$ and $Z'$ for either the intra and interclass.

**Remarks** Suppose one measures the rotation angle between the iris and the camera during enrollment, and stores this as additional information in the database. During verification, one could use this information to retrieve a measurement with both irises aligned. This can be done by taking a picture with both eyes visible of the person in front of the camera and determine where the center points of both irises are located. By drawing an imaginary line between the iris center points of both eyes and a horizontal line through the camera lens, one can determine the angle. Knowledge of the rotation difference between both sessions would imply that no rotation compensation is necessary, and just one secret should be retrieved.

### 6.6 Composition of vectors $Z'$

Confidential
Figure 6.8: Both figures portrays the FRR and FAR when computing the BER between vectors $Z$ and $Z'$ belonging to the intra and interclass. The system performance illustrated in the left contains BER computation between vector $Z$ and all rotation compensated vectors $Z'$. The picture on the right reflects the performance when only the BER is calculated between $Z$ and corresponding aligned $Z'$ without using the mask vectors $P$ and $P'$. This is done by selecting only the minimum BER between $Z$ and all rotation compensated vectors $Z'$. In this simulation $N=2048$ and database configuration $A$. 
Section 7

Reliable Component Selection

This chapter discusses how reliable components are selected during enrollment and how they affect the system performance.

7.1 Rotation alignment

All binary templates that are generated from an individual’s iris during enrollment, are observed in order to find reliable components in its enrollment set $X$. An individual has a number of $N_X$ templates present in its enrollment set. However, the user in front of the camera could move its head between the measurements, which eventuates in rotational differences between templates. To enable a selection of reliable components derived from enrollment set $X$, the binary templates should be aligned towards each other\(^1\). The algorithm to align the templates is as follows:

1. Take the first template $X^1$ in the enrollment as reference template.

   Continue for each $p$ that refers to the remaining templates in the set $\{X^p\}_{p=2}^{N_X}$:

2. Compose a set with shifted versions of $X^p$ denoted as $X_{shift}^p = \text{Shift}(X^p, A)$, where $A$ contains different values $\alpha$, specifying the shift size of a binary template. The different values for $\alpha$ are $A = \{-64, -48, -32, -16, 0, 16, 32, 48, 64\}$.

3. Compute the set of Hamming Distance values $HDP$ between reference $X^1$ and $X_{shift}^p$ defined as $HDP = \text{HamDist}(X^1, X_{shift}^p)$.

4. Select the minimum Hamming Distance $H_{D_{min}}^p$ in $HDP$, specified as $\text{min}(HDP)$.

5. Determine which shift size value $\alpha$ in $A$ induced $H_{D_{min}}^p$.

6. Shifting $X^p$ with $\alpha$ bit positions that causes $H_{D_{min}}^p$, results in an alignment of $X^1$ with $X^p$. This aligned vector is defined as $X_{al}^p = \text{Shift}(X^p, \alpha)$.

This alignment process composes a set $X_{al} = \{X_{al}^p\}_{p=1}^{N_X}$ in which all initial non aligned templates in enrollment $X$ are aligned to reference template $X^1 = X_{al}^1$.

\(^1\)This procedure is implemented in functional block “Rotation Alignment” from figure 6.4.

© Koninklijke Philips Electronics N.V. 2006 53
7.2 Reliable components

This section described the definition of reliable components in enrollment set X and how they affect the system performance compared with the unprotected iris recognition system\(^2\). The initial starting point for determining which components are reliable is the set with aligned enrollment templates \(X_{\text{al}}\). Furthermore an aligned template contains an information vector \(X_{\text{al}}\) and mask vector \(X_{\text{M,al}}\). The definition of a reliable component becomes:

\[
X_{\text{Irel}}(k) \land X_{\text{M,rel}}(k) = \begin{cases} 
0 \land 1, & X_{\text{al}}^{p}(k) = 0 \land X_{\text{M,al}}^{p}(k) = 1 \text{ for all } 1 \leq p \leq N_X \\
1 \land 1, & X_{\text{al}}^{p}(k) = 1 \land X_{\text{M,al}}^{p}(k) = 1 \text{ for all } 1 \leq p \leq N_X 
\end{cases}
\]

Where variable \(k\) denotes the \(k\)th component in \(X_{\text{Irel}}\) and varies between \(1 \leq k \leq L\).

Less formally stated, all components at position \(k\) in the information part of the aligned templates should be either 0 or 1, moreover all corresponding mask bits at position \(k\) must be 1. After determining which components are reliable, \(X_{\text{rel}} = [X_{\text{Irel}}\ X_{\text{M,rel}}]\) is created for user \(m\), containing all reliable components in enrollment set \(X\).

The selection of reliable components in the enrollment templates reduces the number of valid components, i.e. components that refer to the iris region. Comparing figures 5.3 and 7.1(a) gives an indication of this. Moreover, the more templates are used in the enrollment set the less, reliable components are found. This is indicated in figure 7.1. Table 7.1 gives the minimum, maximum and mean number of reliable components in the enrollment set observed for database A, B and C respectively. Comparison of the figures 7.2 and 5.5 gives a better understanding how reliable templates affect the system performance. The use of reliable templates instead of normal binary iris templates, decreases the intraclass variance and increases the interclass variance. Furthermore, the imposter distribution is shifted further towards the left, although both classes are separable. Hence EER = 0%, which is the same as using regular templates.

<table>
<thead>
<tr>
<th>Nmr. Rel. Comp</th>
<th>DB A</th>
<th>DB B</th>
<th>DB C</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>294</td>
<td>194</td>
<td>232</td>
</tr>
<tr>
<td>max</td>
<td>898</td>
<td>800</td>
<td>722</td>
</tr>
<tr>
<td>mean</td>
<td>539</td>
<td>455</td>
<td>411</td>
</tr>
</tbody>
</table>

Table 7.1: This table indicates the minimum, maximum and mean number of reliable components in the enrollment set for database A, B and C.

7.3 Reliable component subset selection

The iris template protection scheme can operate in different values for parameter \(N\). The value of \(N\) determines how many reliable components are selected to compose vector \(Z\) and in addition, it configures the ECC code word length. \(N\) also influences the system performance in terms of FRR and FAR. In particular, the operation secret size \(S\) and error correction capability \(T\) are also dependent of the selected \(N\).

This section describes which strategies were implemented in order to find an appropriate subset

\(^2\)Determining reliable components in the enrollment set is implemented in functional block "Reliable Component Selection" from figure 6.4.
Figure 7.1: Figure 7.1(a), 7.1(b) and 7.1(c) shows the distribution of the number of reliable component in the enrollment set, observed for database A, B and C, respectively.

of reliable components, when there are more than \( N \) reliable components available in the enrollment set \( X \) for an individual \( m \). This reliable component subset selection is visible in figure 6.4\(^3\). This module selects \( N \) reliable components in reliable template \( X_{rel} \), which is provided by the preceding module, using different strategies and statistics derived from the entire enrolled population. The indices of the selected reliable components in \( X_{rel} \) are stored in \( W_1 \). The selected reliable components themselves, construct vector \( Z \). Possible values for \( N \) with regards to the BCH error correction code and the maximum iris template length of \( L=2048 \) could be 2047, 1023, 511, 255 and 127 bits. The explored strategies are based on:

1. Random selection
2. The probability a mask component is valid
3. Discriminating components
4. Signal to noise ratio

\(^{3}\)Indicated by functional block "Select N Reliable components".
Figure 7.2: Both figures illustrate the intraclass and interclass variation, after comparing reliable templates with verification templates.

5. Dependency (correlation) between different components

If the number of reliable components is lower than the configured $N$, the vector $Z$ is zero-padded. This holds for all selection strategies mentioned above. Furthermore, the performance for every subset selection strategy is provided for database configurations A. Some database selection strategies are extended with the performance with database configurations B and C. In addition, the system performance tests for different values of $N$, have been computed for both alternative methods of composition on set $Z'$. Section 6.6 provides a more detailed explanation on both alternatives.
7.3.1 No reliable component selection

A straightforward way to protect the iris templates would be if \( Z = X_I \) and \( W_I = X_M \), i.e. directly forwarding the templates generated by Daugman's method into the template protection system, without selecting the reliable components in the enrollment set \( X \) for an individual. Figure 7.3 illustrates the system performance when adopting this approach for database A. Mask vectors are ignored in this test, i.e only the information parts of genuine and imposter templates are matched. After simulating this approach, the EER = 4.2% at a number of 619 bit errors. One can conclude that refraining from reliable component selection leads to a poor system performance compared with Daugman's unprotected iris recognition system. In the last mentioned system, both the intra class and interclass are better distinguishable, as can be seen in figure 5.5.

![Figure 7.3: Performance of template protection scheme without reliable component selection. Database A.](image)

© Koninklijke Philips Electronics N.V. 2006
7.3.2 Use all available reliable components

The selection of reliable components in enrollment set X for an individual that has access to the database, affects the system performance positively. Figure 7.4(a) compared to figure 7.3 depicts a reduced number of errors when a selection of reliable components in X is performed. Although both the intra and interclass are better distinguished with this method, there is still an overlap in both classes since the EER = 1.3% at 125 bit errors. The remaining figures 7.4(b) and 7.4(c) show that there is a gain in the FRR when more enrollment templates are used to find reliable components.

![Graphs showing FRR and FAR curves for databases A, B, and C.](image)

Figure 7.4: FRR and FAR curves for Alternative 1 and 2 when all available reliable components in enrollment set X are selected for composition of vector Z. Performance is observed for database A, B and C.
7.3.3 Random selection

With this strategy, \( N \) reliable components in the range of 1 to \( L \) in reliable template \( X_{rel} \) produced for user \( m \), are selected randomly. All reliable components have an equal probability to be chosen.

**Simulation results** In the previous section, no reliable component subset selection is performed while determining the system's FRR and FAR. The FRR and FAR for different values of code word length \( N \) and both alternatives are illustrated in figures 7.5(a) and 7.5(b). One can see in both figures that the intra and interclass are well separated only in case a configuration of \( N=255 \) is selected. This means that when \( N=255 \), neither a genuine template pair was falsely rejected nor an imposter template pair was falsely accepted. For this reliable component subset selection method and all in this section described remaining methods, the simulation results for \( N=255 \) are used to give an indication of the system's behavior in terms of error correction capability \( T \) and secret size \( K \). The motivation why \( N=255 \) was chosen to signify the difference between experiments, is that in this configuration almost no zero padding has to be performed in order to compose vector \( Z \). In essence, zero padding eventuates in a varying number of elements in helper data \( W_i \). A constant number of elements for all created helper data \( W_i \) vectors, would probably result in more reliable simulation statistics. Figure 7.6 shows the mean reliable component value of feature vector \( X \) when performing a random subset selection for \( N = 255 \) on database A.

Table 7.2 gives for each value of \( N \) the maximum number of observed errors between two templates, after comparing all possible intraclass templates with database configuration A. Likewise, the minimum number of observed errors for all interclass template comparison is provided in this table. Both observations are done for Alternative 1 and 2. The difference between the minimum and maximum number of observed errors gives an indication about the intra and interclass.
Figure 7.6: This figure shows for database A, the mean value of random selected reliable components in feature vector X when N = 255.

terclass separability. In addition, the EER is given for every simulated N.

<table>
<thead>
<tr>
<th>Random Selection</th>
<th>N=127</th>
<th>N=255</th>
<th>N=511</th>
<th>N=2048</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alt. 1</td>
<td>Alt. 2</td>
<td>Alt. 1</td>
<td>Alt. 2</td>
</tr>
<tr>
<td>Max Error Intra-class</td>
<td>38</td>
<td>33</td>
<td>75</td>
<td>61</td>
</tr>
<tr>
<td>Min Error Inter-class</td>
<td>35</td>
<td>28</td>
<td>75</td>
<td>70</td>
</tr>
<tr>
<td>EER %</td>
<td>0.08</td>
<td>0.05</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.2: This table reflects for different values of N the maximum and minimum number of observed errors for respectively the intra and interclass using database A. The last row of this table reflects the EER for different N.

Table 7.3 highlights for N = 255 the FRR and FAR as a function of different secret sizes K and error correction capabilities T for all database configurations. A favorable secret size of K \approx 60 gives an undesirably high FRR. Even though both class are separable, like in Daugman’s system, choosing an error correction capability T = 63 bits, which enables error correction of all observed errors, results in an extreme low secret size of K = 9 bits.
<table>
<thead>
<tr>
<th>N=255</th>
<th>DB A</th>
<th>DB B</th>
<th>DB C</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>T</td>
<td>FRR%</td>
<td>FAR%</td>
</tr>
<tr>
<td>63</td>
<td>30</td>
<td>12.02</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>31</td>
<td>10.81</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>42</td>
<td>2.66</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>43</td>
<td>2.42</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>1.69</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>47</td>
<td>1.53</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.3: FRR and FAR as function of different error correction capabilities T and corresponding secret size K for database A, B and C. Observations are done with system Alternative 2.
7.3.4 Probability of a valid mask component

The probability of a valid component is not uniformly distributed along the entire iris template. Presumably, occluding eyelids, eyelashes etc. reduces this probability at certain locations in the iris template. For each \( k^{th} \) position in the binary template, the probability that this component is valid is estimated according to the following equation:

\[
P_X(k) = \mathbb{P}(X_{M_m}^p(k) = 1) = \frac{1}{N_U N_X} \sum_{n=1}^{N_U} \sum_{p=1}^{N_X} (X_{M_m}^p(k))
\]  

(7.1)

Figure 7.7 reflects this probability at each component position in a binary template. Especially in the middle section of a binary template, this probability is low. A random selection can still select reliable components in this region. The use of function \( P_X \) makes it possible to select those \( N \) reliable components, that are most likely valid. This probability also holds during verification. This subset selection reduces the risk that components which are not valid in both vectors \( Z \) and \( Z' \) are compared. This possibly decreases the probability of a disagreeing information component pair in \( Z \) and \( Z' \).

Simulation results  The system performance is depicted in figure 7.8 and tables 7.5 and 7.4. Figure 7.9 shows for database A and \( N = 255 \) the mean reliable component value of feature vector \( X \) when performing the subset selection strategy described in this section. There are hardly any components selected in the middle part along the template, as the probability of a valid bit in this region is low.
Figure 7.8: Simulated FRR and FAR curves for different values of $N$ and database A, when selecting those components in enrollment set $X$, that have the highest probability of being valid also during the verification phase. Figure 7.8(a) shows the system performance for Alternative 1, figure 7.8(b) depicts this for Alternative 2. Database A.

![Graph](image)

(a) Alternative 1

(b) Alternative 2

Table 7.4: This table reflects for different values of $N$ the maximum and minimum number of observed errors for respectively the intra and interclass using database A. The last row of this table reflects the EER for different $N$.

<table>
<thead>
<tr>
<th>Probability Selection</th>
<th>N=127</th>
<th>N=255</th>
<th>N=511</th>
<th>N=2048</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alt. 1</td>
<td>Alt. 2</td>
<td>Alt. 1</td>
<td>Alt. 2</td>
</tr>
<tr>
<td>Max Error Intraclass</td>
<td>40</td>
<td>31</td>
<td>69</td>
<td>59</td>
</tr>
<tr>
<td>Min Error Interclass</td>
<td>30</td>
<td>25</td>
<td>75</td>
<td>70</td>
</tr>
<tr>
<td>EER %</td>
<td>0.2</td>
<td>0.03</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 7.9: This figure shows for database A and $N = 255$, the mean reliable component value of feature vector $X$ that has the highest probability to be valid.

© Koninklijke Philips Electronics N.V. 2006

63
<table>
<thead>
<tr>
<th>K</th>
<th>T</th>
<th>FRR%</th>
<th>FAR%</th>
<th>FRR%</th>
<th>FAR%</th>
<th>FRR%</th>
<th>FAR%</th>
</tr>
</thead>
<tbody>
<tr>
<td>63</td>
<td>30</td>
<td>10.97</td>
<td>0</td>
<td>4.96</td>
<td>0</td>
<td>1.71</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>31</td>
<td>9.52</td>
<td>0</td>
<td>4.68</td>
<td>0</td>
<td>1.14</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>42</td>
<td>2.26</td>
<td>0</td>
<td>1.56</td>
<td>0</td>
<td>0.09</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>43</td>
<td>1.69</td>
<td>0</td>
<td>1.27</td>
<td>0</td>
<td>0.09</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>1.13</td>
<td>0</td>
<td>0.85</td>
<td>0</td>
<td>0.09</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>47</td>
<td>0.89</td>
<td>0</td>
<td>0.49</td>
<td>0</td>
<td>0.09</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.5: FRR and FAR as a function of different error correction capabilities $T$ and corresponding secret size $K$ for database A, B and C. Observations are done with system alternative 2.
7.3.5 Discriminating components

The idea behind this strategy is to select those \( N \) reliable component for \( m \) which differ the most, compared to the corresponding average value of that component in the database. The average value for each component \( k \) along the entire template is computed as follows

\[
A_X(k) = \frac{\sum_{m=1}^{N_c} \sum_{p=1}^{N_X} (X_{cm}^m(k) | X_{M_m}^p(k) = 1)}{\sum_{m=1}^{N_c} \sum_{p=1}^{N_X} X_{M_m}^p(k) = 1}
\]  

(7.2)

The corresponding plot is shown in figure 7.10. One can see that, especially in the middle section of this plot, the average values vary significantly. The average value for each component is based on a varying number of valid components. As the probability of a valid mask component is low in middle region of an iris, a small number of valid information components was used to estimate which average value this component has. If there is a distinct difference between the actual value

![](image)

Figure 7.10: Average value computed for all components

of a reliable component \( k \) for user \( m \) and the average value of that component in the database, component \( k \) is eligible as being a discriminant component for user \( m \). The selection procedure of the \( N \) reliable component is as follows. For user \( m \) the difference between the actual value and the average value for each component is computed, giving

\[
D_X(k) = |X_{rel}(k) - A_X(k)|, \quad \text{for all } (1 \leq k \leq L)
\]

(7.3)

Afterwards, the reliable components in \( X_{rel} \) that correspond to the \( N \) highest values in \( D(k) \) are selected in order to compose \( Z \).

Simulation results The system performance is depicted in figure 7.11 and tables 7.7 and 7.6. Figure 7.12 shows for database A and \( N = 255 \) the average reliable component values when the discriminant components are selected in the enrollment set. The uniqueness or discriminating property of the selected reliable components is translated in more extreme values, i.e. components that have a mean value of 0 or 1.
Figure 7.11: Simulated FRR and FAR curves for different values of $N$ and database A, when selecting discriminating components in enrollment set $X$. Figure 7.11(a) shows the system performance for alternative 1, figure 7.11(b) depicts this for alternative 2.

<table>
<thead>
<tr>
<th>Discriminant Selection</th>
<th>$N$=127</th>
<th>$N$=255</th>
<th>$N$=511</th>
<th>$N$=2048</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alt. 1</td>
<td>Alt. 2</td>
<td>Alt. 1</td>
<td>Alt. 2</td>
</tr>
<tr>
<td>Max Error Intraclass</td>
<td>40</td>
<td>28</td>
<td>72</td>
<td>59</td>
</tr>
<tr>
<td>Min Error Interclass</td>
<td>28</td>
<td>10</td>
<td>75</td>
<td>29</td>
</tr>
<tr>
<td>EER %</td>
<td>0.17</td>
<td>1.2</td>
<td>0</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 7.6: This table reflects for different values of $N$ the maximum and minimum number of observed errors for respectively the intra and interclass using database A. The last row of this table reflects the EER for different $N$.

Figure 7.12: This figure shows, for database A and $N = 255$, the mean reliable component value of feature vector $X$ when discriminant components in the enrollment set are chosen.
<table>
<thead>
<tr>
<th>N=255</th>
<th>DB A</th>
<th>DB B</th>
<th>DB C</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>T</td>
<td>FRR%</td>
<td>FAR%</td>
</tr>
<tr>
<td>63</td>
<td>30</td>
<td>4.59</td>
<td>0.01</td>
</tr>
<tr>
<td>55</td>
<td>31</td>
<td>4.19</td>
<td>0.02</td>
</tr>
<tr>
<td>47</td>
<td>42</td>
<td>0.56</td>
<td>0.69</td>
</tr>
<tr>
<td>45</td>
<td>43</td>
<td>0.56</td>
<td>0.84</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>0.48</td>
<td>1.19</td>
</tr>
<tr>
<td>29</td>
<td>47</td>
<td>0.24</td>
<td>1.72</td>
</tr>
</tbody>
</table>

Table 7.7: FRR and FAR as a function of different error correction capabilities T and corresponding secret size K for database A, B and C. Observations are done with system alternative 2.
7.3.6 Signal to Noise Ratio

The selection of the components with the highest Signal to Noise Ratio (SNR) is described in this section. This strategy is adopted from the earlier implemented fingerprint template protection system described in [21]. This SNR selection is adapted to iris templates. The SNR for each component is derived as follows

\[
\text{SNR}_X(k) = \frac{\sigma_X^2(k)}{\sigma_N^2(k)}
\]  

(7.4)

It is important to take only those components into account, that are valid indicated by the mask vector of a template. The mean value for each position \(k\) in enrollment template \(X\) for user \(m\) is calculated as:

\[
X_{\mu,m}(k) = \frac{1}{X_{V,m}} \sum_{p=1}^{N_x} \left( X_{I_m^p}(k) | X_{M_m^p}(k) = 1 \right)
\]

(7.5)

where,

\[
X_{V,m}(k) = \sum_{p=1}^{N_x} X_{M_m^p}(k)
\]

(7.6)

For user \(m\), it sums all valid information components in the enrollment set at position \(k\), and is normalized with the number of valid components at position \(k\), denoted with \(X_{V,m}(k)\).

Now the problem occurs that it might be possible that there are components in the enrollment set for user \(m\), which were never valid. For these components, no mean value can be calculated. These components are kept out of consideration while computing \(\sigma_X^2(k)\). Thus,

\[
\sigma_X^2(k) = \frac{1}{N_U - N_V(k)} \sum_{m=1}^{N_U} \left( (X_{\mu,m}(k) - A_X(k))^2 | X_{V,m}(k) > 0 \right),
\]

(7.7)

where \(N_V(k)\) specifies the number of users in the database, that have at least one valid component in their enrollment set at position \(k\).

The noise variance for each component is calculated by means of

\[
\sigma_N^2(k) = \frac{1}{N_U} \sum_{m=1}^{N_U} \sum_{p=1}^{N_x} \left( (X_{I_m^p}(k) | X_{M_m^p}(k) = 1 \right) - A_X(k))^2
\]

(7.8)

\[
\sum_{m=1}^{N_U} \sum_{p=1}^{N_x} X_{M_m^p}(k)
\]

Figure 7.13 depicts the SNR value at each component position. After calculating \(\text{SNR}_X(k)\), the \(N\) reliable components with the highest SNR value are selected in order to compose vector \(Z'\).

**Simulation results** The system performance is depicted in figure 7.14 and tables 7.9 and 7.8. Figure 7.15 shows for database A and \(N = 255\), the mean value of selected reliable components that have the highest SNR values.
Figure 7.13: SNR value for each component.

<table>
<thead>
<tr>
<th>SNR Selection</th>
<th>N=127</th>
<th>N=255</th>
<th>N=511</th>
<th>N=2048</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alt. 1</td>
<td>Alt. 2</td>
<td>Alt. 1</td>
<td>Alt. 2</td>
</tr>
<tr>
<td>Max Error Intraclass</td>
<td>39</td>
<td>41</td>
<td>76</td>
<td>85</td>
</tr>
<tr>
<td>Min Error Interclass</td>
<td>32</td>
<td>23</td>
<td>74</td>
<td>65</td>
</tr>
<tr>
<td>EER %</td>
<td>0.16</td>
<td>0.18</td>
<td>9x10^-3</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 7.8: This table reflects for different values of N, the maximum and minimum number of observed errors for respectively the intra and interclass using, database A. The last row of this table reflects the EER for different N.

Figure 7.14: Simulated FRR and FAR curves for different values of N and database A, when the components with the highest signal to noise ratio in enrollment set X are selected. Figure 7.14(a) shows the system performance for alternative 1, figure 7.14(b) depicts this for alternative 2.
Figure 7.15: This figure shows for database A and $N = 255$ the mean reliable component value of feature vector $X$ having the highest SNR values in the enrollment set.

Table 7.9: FRR and FAR as function of different error correction capabilities $T$ and corresponding secret size $K$ for database A, B and C. Observations are done with system alternative 2.
7.3.7 Dependency between components

As mentioned in section 4.6, there is dependency between binary iris templates. This section describes an attempt to explore the correlation between different iris templates. The correlation is explored by trying to find those components that are most independent from each other. The dependency between components is estimated by computing the correlation coefficient between every possible pair of components.

Suppose we have a matrix \( D \), each row in \( D \) represents a unique enrollment template that is present in the database. We split the corresponding information and mask parts in separate matrices \( D_I \) and \( D_M \). The number of components in each template is \( L = 2048 \) and the total number of templates present in the database is \( T = N_U N_X \). Thus,

\[
D_I = \begin{pmatrix}
X_I^1 \\
X_I^2 \\
\vdots \\
X_I^T
\end{pmatrix}, \quad D_M = \begin{pmatrix}
X_M^1 \\
X_M^2 \\
\vdots \\
X_M^T
\end{pmatrix}
\]  

(7.9)

This means that each matrix has \( T \) rows and \( L \) columns. Each column in \( D_I \) is considered as a sequence of binary numbers, related to the \( T \) different values for a component. Let us denote \( x_i^t \) the \( k \)th component in the \( t \)th information vector \( X_I \) in the database \( D_I \). We want to explore the dependency between each possible column (iris component) pair, by means of computing the correlation coefficient for all possible combinations of two components. When we assume that all mask components in \( D_M \) are valid, the computation of the correlation coefficients between two different components \( x_i \) and \( x_j \), is as follows.

For the \( T \) available templates, we calculate the mean value for every \( k \)th component as

\[
\mu_{x_k} = E(x_k) = \frac{1}{T} \sum_{i=1}^{T} x_i^k
\]  

(7.10)

The variance for every \( k \)th component becomes

\[
\sigma_k^2 = E(x_k - \mu_{x_k})^2 = \frac{1}{T-1} \sum_{i=1}^{T} (x_i^k - \mu_{x_k})^2
\]  

(7.11)

The covariance between \( x_i \) and \( x_j \) is given as

\[
\sigma_{ij}^2 = E[(x_i - \mu_{x_i})(x_j - \mu_{x_j})]
\]  

\[
= \frac{1}{T-1} \sum_{i=1}^{T} (x_i^k - \mu_{x_k})(x_j^k - \mu_{x_j})
\]  

(7.12)

As there are \( L \) different components in an iris template, the covariance matrix \( C_X \) between any of them becomes

\[
C_X = \begin{pmatrix}
\sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1L}^2 \\
\sigma_{21}^2 & \sigma_{22}^2 & \cdots & \sigma_{2L}^2 \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{L1}^2 & \sigma_{L2}^2 & \cdots & \sigma_{LL}^2
\end{pmatrix}
\]  

(7.13)
The correlation coefficient between two components can be written as

$$\rho_{ij} = \frac{\sigma_{ij}^2}{\sigma_i \sigma_j}$$  \hspace{1cm} (7.14)

When this is done for all possible combinations of two components, the correlation matrix \( R_x \) becomes

$$R_x = \begin{pmatrix}
\rho_{11} & \rho_{12} & \cdots & \rho_{1L} \\
\rho_{21} & \rho_{22} & \cdots & \rho_{2L} \\
\vdots & \vdots & \ddots & \vdots \\
\rho_{L1} & \rho_{L2} & \cdots & \rho_{LL}
\end{pmatrix}$$  \hspace{1cm} (7.15)

Doing the above correlation coefficient computations when taking the mask templates in \( D_M \) into account, gives some difficulties. A component \( x_i \) can be (in)valid, as well as the second observed component \( x_j \) and it should be ignored while calculating the correlation coefficient between \( x_i \) and \( x_j \). Only components \( x_i \) and \( x_j \), which are both valid in the same template \( t \), are taken into account. The sequence length \( T \) is no longer legitimate when one or both components \( x_i \) and \( x_j \) are invalid in a template \( t \), as the covariance is computed by equation (7.12). This implies that the correlation coefficients \( \rho_{ij} \) in matrix \( R_x \) are computed for different sequence lengths. This also means that not every computed \( \rho_{ij} \) is representative, as it could be possible that there are too less valid mask component matches between two compared components \( x_i \) and \( x_j \). This especially holds when computing the correlation coefficient between two components that are located in the middle section of a template. As is indicated in figure 7.7, the probability is low that a component is valid in this region.

The mean value, variance and covariance, represented by equations (7.10) to (7.12) are computed using only valid components. Furthermore, when all valid components \( x_k \) for the \( T \) observed templates have the same binary value, the variance \( \sigma_k^2 \) becomes zero. This results in equation (7.14) into a division by zero, when the standard deviation \( \sigma_i \) or \( \sigma_j \) appears to be zero. In this situation, the correlation coefficient is chosen to become \( \rho_{ij} = 1 \). Figure 7.16(a) illustrates the correlation matrix \( R_x \) after computing the correlation coefficient between every possible component pair. Some vertical and horizontal lines are visible in the figure for components located at approximately \( 800 \leq k \leq 1200 \). For these component positions it holds that there were not enough valid components to calculate a correlation coefficient for, or have zero variance. Figure 7.16(b) depicts the correlation coefficients between the first 800 components. Both figures reflect that there is correlation between the binary valued components, although in a relatively short range for \( k \pm 75 \).

**Simulation results**  The system performance is depicted in figure 7.17 and tables 7.11 and 7.10. Figure 7.18 shows the mean value of selected reliable components that have the least correlation in the enrollment set. This experiment uses the algorithm, explained in section 7.3.8. In this algorithm, the relative importance of the accompanying functions \( SNR_X \), \( P_X \) and \( D_X \), in order to create a subset of reliable components, are configured according to the weighed factors \( \alpha_1 \), \( \alpha_2 \) and \( \alpha_3 \), see equations (7.16) and (7.17). When all weighted factors are set to 0, except for \( \alpha_4 = 1 \), one selects only the components that depend on correlation matrix \( R_x \), i.e. leaving the above mentioned functions out of consideration. In this way, every new in step 3 selected component \( x_k \), has the lowest correlation with the previously selected components.
Figure 7.16: These figures illustrate the correlation between the components. Every pixel in this figure illustrates a correlation coefficient between a component pair $x_i$ and $x_j$. The pixel color range is black ($\rho_{ij} = -1$) to white ($\rho_{ij} = 1$).
Figure 7.17: Simulated FRR and FAR curves for different values of $N$ and database A, when selecting in enrollment set $X$ the components with the lowest mutual correlation. Figure 7.20(a) shows the system performance for alternative 1, figure 7.20(b) depicts this for alternative 2.

<table>
<thead>
<tr>
<th>Correlation Selection</th>
<th>N=127</th>
<th>N=255</th>
<th>N=511</th>
<th>N=2048</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alt. 1</td>
<td>Alt. 2</td>
<td>Alt. 1</td>
<td>Alt. 2</td>
</tr>
<tr>
<td>Max Error Intraclass</td>
<td>40</td>
<td>31</td>
<td>76</td>
<td>65</td>
</tr>
<tr>
<td>Min Error Interclass</td>
<td>35</td>
<td>30</td>
<td>84</td>
<td>75</td>
</tr>
<tr>
<td>EER %</td>
<td>0.08</td>
<td>0.02</td>
<td>0</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Table 7.10: This table reflects for different values of $N$ the maximum and minimum number of observed errors for respectively the intra and interclass using database A. The last row of this table reflects the EER for different N.

Figure 7.18: This figure shows for database A and $N = 255$ the mean value of selected reliable components that have the least correlation in the enrollment set.
### Correlation Selection

<table>
<thead>
<tr>
<th>N=255</th>
<th>DB A</th>
<th></th>
<th></th>
<th>DB B</th>
<th></th>
<th></th>
<th>DB C</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>K</td>
<td>T</td>
<td>FRR%</td>
<td>FAR%</td>
<td>FRR%</td>
<td>FAR%</td>
<td>FRR%</td>
<td>FAR%</td>
<td>FRR%</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>---</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>63</td>
<td>30</td>
<td>16.53</td>
<td>0</td>
<td>7.59</td>
<td>0</td>
<td>4.31</td>
<td>0</td>
<td>4.31</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>31</td>
<td>15.24</td>
<td>0</td>
<td>6.31</td>
<td>0</td>
<td>3.15</td>
<td>0</td>
<td>3.15</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>42</td>
<td>2.98</td>
<td>0</td>
<td>1.56</td>
<td>0</td>
<td>0.85</td>
<td>0</td>
<td>0.85</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>43</td>
<td>2.66</td>
<td>0</td>
<td>1.49</td>
<td>0</td>
<td>0.74</td>
<td>0</td>
<td>0.74</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>2.17</td>
<td>0</td>
<td>1.35</td>
<td>0</td>
<td>0.67</td>
<td>0</td>
<td>0.67</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>47</td>
<td>1.69</td>
<td>0</td>
<td>0.99</td>
<td>0</td>
<td>0.30</td>
<td>0</td>
<td>0.30</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.11: FRR and FAR as function of different error correction capabilities T and corresponding secret size K for database A, B and C. Observations are done with system alternative 2.
7.3.8 Combining several strategies

The problem with especially strategy 7.3.5, 7.3.6 and 7.3.7, is that the statistics computed for every component position \( k \) depend on the number of available valid components at position \( k \). Combining several strategies would possibly benefit from each singular subset selection strategy.

Finding a subset of reliable components when combining several strategies, is based on an algorithm explained in [32]. The subset selection strategy in this section, combines the functions \( SNR_x, P_x \) and \( D_x \) with incorporate correlation information provided in correlation matrix \( R_x \). The algorithm is as follows:

1. Select one of the above mentioned functions. In this algorithm, we choose \( P_x(k) \). Rank \( P_x(k), k = 1, 2, \ldots, L \) in a descending order and choose the component with the highest value. Let us say this is \( X_{i_1} \).

2. Choose the the second component \( X_{i_2} \) for which

\[
i_2 = \max_j \left\{ \alpha_1 SNR_x(j) + \alpha_2 P_x(j) + \alpha_3 D_x(j) - \rho_{i_1} \right\}, \quad \text{for all } j \neq i_1
\]

(7.16)

Parameters \( \alpha_1, \alpha_2, \alpha_3 \) and \( \alpha_4 \) are weighed factors that determine the relative importance of the four terms. This means for the selection of the next \( i_2 \), that we do not only take the selection criteria into account, but also the correlation of the already chosen component \( i_1 \).

3. Select \( X_{i_k}, k = 3, \ldots, N \), so that

\[
i_k = \max_j \left\{ \alpha_1 SNR_x(j) + \alpha_2 P_x(j) + \alpha_3 D_x(j) - \frac{\alpha_4}{k-1} \sum_{r=1}^{k-1} \rho_{i_r} \right\}, \quad \text{for } j \neq i_r, \quad r = 1, 2, \ldots, k - 1
\]

(7.17)

That is, the average correlation with all previously selected components is taken into account.

In figure 7.19, one can see for different \( N \), the mean value and at which positions along the entire range, reliable components have been selected, after performing the subset selection method described in this section. For \( N = 127 \), none of the reliable components were selected in the middle region of the reliable templates. The statistics used as input for the cost function, decide that the components at the begin and end part are more favorable. When \( N \) is increased the initially less favorable reliable components should be selected as well. Figure 7.19(d) reflects the position and corresponding average value of all reliable components, used by the iris template protection system, when i.e. no subset selection is performed.
Simulation results database A  This section reflects the system performance after applying the reliable component selection algorithm, described above, and is performed when all weighed factors $\alpha$ are set to an equal weight of $\frac{1}{4}$. The system performance is depicted in figure 7.20 and tables 7.12 and 7.13. Even better results might be obtained by optimizing the algorithm by assigning different values for $\alpha_1$ to $\alpha_4$ at each new simulation. Although many experiments on this did not give a significant performance gain.

A disadvantage for this combined subset selection strategy, as well as the previously explained method is its high complexity. Computing the correlation matrix $R_X$ has a complexity of $O(L^2)$. Step 3 in the algorithm has a complexity of $O(N^2)$.

<table>
<thead>
<tr>
<th>Combined Selection</th>
<th>N=255</th>
<th></th>
<th>Combined Selection</th>
<th>N=511</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>T</td>
<td>FRR %</td>
<td>FAR %</td>
<td>K</td>
</tr>
<tr>
<td>63</td>
<td>30</td>
<td>6.53</td>
<td>0</td>
<td>93</td>
</tr>
<tr>
<td>55</td>
<td>31</td>
<td>6.13</td>
<td>0</td>
<td>85</td>
</tr>
<tr>
<td>47</td>
<td>42</td>
<td>1.23</td>
<td>0</td>
<td>76</td>
</tr>
<tr>
<td>45</td>
<td>43</td>
<td>1.13</td>
<td>0</td>
<td>67</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>0.73</td>
<td>0</td>
<td>58</td>
</tr>
<tr>
<td>29</td>
<td>47</td>
<td>0.40</td>
<td>0</td>
<td>49</td>
</tr>
</tbody>
</table>

Table 7.12: FRR and FAR as function of different error correction capabilities $T$ and corresponding secret size $K$ for database A. Table (a) shows this for $N = 255$ and table (b) for $N = 511$.

<table>
<thead>
<tr>
<th>Combined Selection</th>
<th>N=127</th>
<th>N=255</th>
<th>N=511</th>
<th>N=2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Error Intraclass</td>
<td>Alt. 1</td>
<td>Alt. 2</td>
<td>Alt. 1</td>
<td>Alt. 2</td>
</tr>
<tr>
<td>29</td>
<td>23</td>
<td>59</td>
<td>51</td>
<td>140</td>
</tr>
<tr>
<td>Min Error Interclass</td>
<td>26</td>
<td>21</td>
<td>70</td>
<td>68</td>
</tr>
<tr>
<td>EER %</td>
<td>$5 \times 10^{-3}$</td>
<td>$3 \times 10^{-3}$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.13: This table reflects for different values of $N$ the maximum and minimum number of observed errors for respectively the intra and interclass using database A. The last row of this table reflects the EER for different $N$. 

© Koninklijke Philips Electronics N.V. 2006
Figure 7.19: These figures show for different $N$ the mean reliable component value. Only those component positions that were selected by helper data $W_i$, are indicated after a subset selection was performed. Figure 7.19(d) indicates no subset selection, as $N = 2048$. In this case all, components that are reliable are confined by helper data $W_i$. Furthermore this selection strategy is performed on database A.
Figure 7.20: Simulated FRR and FAR curves for different values of $N$ and database A, when selecting in enrollment set $X$ the components according to a combination of several selection strategies. Figure 7.17(a) shows the system performance for alternative 1, figure 7.17(b) depicts this for alternative 2.
Simulation results database B  The system performance, according to the subset selection strategy described in this section 7.3.8, is visualized in figure 7.21 and tables 7.14 and 7.15. Comparing table 7.14 with table 7.12, indicates an improvement of the FRR. Note that both tables are obtained after simulating the system performance using the same subset selection strategy. Apparently, using more enrollment templates to determine the reliable components that refer to an individual, reduces the noise factor, i.e. less errors have to be corrected.

The S-shaped FAR-curve visible in figure 7.21 for \( N = 511 \), is probably caused by the effect that too much zero padding was necessary in order to compose vector \( Z \). The previous database configuration required less zero padding for \( N = 511 \), giving a steeper and smoother FAR-curve.

![Figure 7.21](image)

(a) Alternative 1. 
(b) Alternative 2.

Figure 7.21: Simulated FRR and FAR curves for different values of \( N \) and database B, when selecting in enrollment set \( X \) the components according to a combination of several selection strategies. Figure 7.17(a) shows the system performance for alternative 1, figure 7.17(b) depicts this for alternative 2.

<table>
<thead>
<tr>
<th>Combined Selection</th>
<th>N=255</th>
<th>Combined Selection</th>
<th>N=511</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K )</td>
<td>( T )</td>
<td>FRR %</td>
<td>FAR %</td>
</tr>
<tr>
<td>63</td>
<td>30</td>
<td>3.97</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>31</td>
<td>3.55</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>42</td>
<td>0.85</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>43</td>
<td>0.57</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>0.43</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>47</td>
<td>0.28</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.14: FRR and FAR as function of different error correction capabilities \( T \) and corresponding secret size \( K \) for database B. Table (a) shows this for \( N = 255 \) and table (b) for \( N = 511 \).
<table>
<thead>
<tr>
<th>Combination Selection</th>
<th>N=127</th>
<th>N=255</th>
<th>N=511</th>
<th>N=2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alt. 1</td>
<td>Alt. 2</td>
<td>Alt. 1</td>
<td>Alt. 2</td>
<td>Alt. 1</td>
</tr>
<tr>
<td>Max Error Intraclass</td>
<td>32</td>
<td>25</td>
<td>66</td>
<td>60</td>
</tr>
<tr>
<td>Min Error Interclass</td>
<td>28</td>
<td>22</td>
<td>66</td>
<td>66</td>
</tr>
<tr>
<td>EER %</td>
<td>0.04</td>
<td>0.03</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.15: This table reflects for different values of $N$ the maximum and minimum number of observed errors for respectively the intra and interclass using database B. The last row of this table reflects the EER for different $N$. 
Simulation results database C: The system performance, according to the subset selection strategy described in section 7.3.8, is visualized in figure 7.22 and tables 7.16 and 7.17. Comparing tables 7.12, 7.14 and 7.16 gives an indication of the finally achieved system performance for the combined subset selection strategy. An examination of these tables makes clear that using more templates in order to find the reliable components in an enrollment set $X$ improves the system performance. The best achieved system performance when $N = 255$ is visualized in table 7.16(a). For $N = 255$, zero padding was necessary for 3% of all observed vectors $Z$. The S-shaped FAR curve for $N = 511$ becomes more intense, as in this simulation the majority for each individual generated vector $Z$ is padded with zeros. For $N = 511$, zero padding was necessary for 77% of all observed vectors $Z$. The need for much zero padding is reflected in figure 7.1(c). The corresponding FRR based on alternative 2 shown in table 7.16(b) is promising. However, the lack of enough reliable components to compose the vectors $Z$ with this database configuration might dissemble this outcome. Probably only when a database could be used, containing iris images which after transformation to iris templates provides sufficient enough reliable components to enable for each individual at least 511 reliable components when $N_x = 5$, eventuates in a reliable judgement about the in table 7.16(b) shown results. This demands a high quality standard on the captured images during enrollment. The in this thesis used CASIA database does not offer such iris images.

Figure 7.22: Simulated FRR and FAR curves for different values of $N$ and database $C$, when selecting in enrollment set $X$ the components according to a combination of several selection strategies. Figure 7.17(a) shows the system performance for alternative 1, figure 7.17(b) depicts this for alternative 2.
### Table 7.16: FRR and FAR as function of different error correction capabilities $T$ and corresponding secret size $K$ for database C. Table (a) shows this for $N=255$ and table (b) for $N=511$.

<table>
<thead>
<tr>
<th>Combined Selection</th>
<th>N=255</th>
<th>Combined Selection</th>
<th>N=511</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$</td>
<td>$T$</td>
<td>FRR %</td>
<td>FAR %</td>
</tr>
<tr>
<td>63</td>
<td>30</td>
<td>1.24</td>
<td>0.00</td>
</tr>
<tr>
<td>55</td>
<td>31</td>
<td>0.76</td>
<td>0.00</td>
</tr>
<tr>
<td>47</td>
<td>42</td>
<td>0.09</td>
<td>0.00</td>
</tr>
<tr>
<td>45</td>
<td>43</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>37</td>
<td>45</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>29</td>
<td>47</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### Table 7.17: This table reflects for different values of $N$ the maximum and minimum number of observed errors for respectively the intra and interclass using database C. The last row of this table reflects the EER for different $N$.

<table>
<thead>
<tr>
<th>Combined Selection</th>
<th>N=127</th>
<th>N=255</th>
<th>N=511</th>
<th>N=2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Error Intraclass</td>
<td>23</td>
<td>19</td>
<td>45</td>
<td>43</td>
</tr>
<tr>
<td>Min Error Interclass</td>
<td>30</td>
<td>22</td>
<td>72</td>
<td>63</td>
</tr>
<tr>
<td>EER %</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

© Koninklijke Philips Electronics N.V. 2006
7.4 Overall evaluation of simulation results

Experiments demonstrated that the best system performance is achieved by a combination of different reliable component subset selection strategies, using $N_X = 5$ enrollment templates. The obtained simulation results when selecting 127 or 255 out of the available reliable components in the enrollment set $X$, indicate the differences between different configured subset selection strategies the best. A satisfying performance in terms of acceptable FRR and security level for $N = 255$ could be a false rejection rate of 1.24%, with a corresponding secret size of $K = 63$. For $N = 511$ even better statistics are achieved, but a database that provides more reliable components for every individual should prove this. As mentioned earlier in this chapter, experiments with $N = 511$ gave very promising results. When $N_X = 5$, a secret size of $K = 93$ bits with an FRR of 0.38% was achieved. However, a lot of zero padding was necessary.

As mentioned in section 5.2, many iris images available in the CASIA database were not transformed to iris templates because of a too poor image quality. In a real iris recognition system, during enrollment and verification, a sequence of pictures is made. The picture with the highest image quality is used for template matching. This might imply that not all transformed images used for testing the performance of the protected system reach this high image quality standard. When having control over the used enrollment and verification images might lead to a better performance in terms of FRR and FAR for all configurations of $N$. For example, a number of $N_X$ captured iris images with the highest quality standard could be used to find the reliable components for an individual. One could capture multiple enrollment pictures as long a minimum of 511 reliable components are found.

The finally achieved FRR and FAR curves for the three database configurations when applying the combined reliable subset selection strategy are illustrated in figure 7.23. Clearly is visible that using more iris pictures as training data reduces the intraclass variation. Figure 7.23(a) shows that the interclass variance will be preserved when $N = 255$, while configuring $N = 511$ results in a varying interclass variance, see figure 7.23(b).

![FRR and FAR curves](image)

(a) Alternative 1.

(b) Alternative 2.

Figure 7.23: FRR and FAR curves for all three database configuration when (a) $N = 255$ and (b) $N=511$, using the subset selection strategy described in section 7.3.8 and system alternative 2. Moreover figure (b) depicts clearly the artifacts in FAR caused by zero padding, as not for every individual 511 reliable components could be found.
Section 8

Conclusions and Recommendations

8.1 Conclusions

The aim of this thesis was to develop a privacy protected authentication system, based on iris data that is robust to noise. The unprotected iris recognition system developed by J. Daugman was used as reference model. Noise influences affect the system performance significantly. The noise factor, which is present in iris templates could be caused by head tilt, occluding eyelids and eyelashes, contact lenses, luminance variations etc. Components that are robust and reliable have to be selected from the iris template to overcome this problem. Reliable components are found by comparing several genuine templates. The main challenge is to find a subset of reliable components that induces a sufficient secret size against an acceptable False Rejection Rate (FRR) and False Acceptance Rate (FAR). A secret size of about 60 bits offers privacy protection to a user. Several strategies were designed and tested to create a subset of available reliable components for each database user, that leads to a satisfying system performance. The strategy that gives the best performance, deals with statistics that are obtained from the entire population available in the database. These are statistics about:

- The probability that an information component is valid in an iris template;
- Finding discriminating reliable components for a user. Discriminating components are components that differ significantly compared with the population's average component value;
- Reliable components that have the highest overall Signal to Noise Ratio (SNR);
- Dependency between components in an iris template.

The strategy which gave the best results combines these statistics in order to create a subset of reliable components. Integrating mask vectors into the iris template protection system is not a straightforward task. Presumably the fractional HD classifier used in an unprotected iris recognition system can not be integrated into the iris template protection system, as the enrolled information feature vector is scrambled, it can not be masked with the associating mask vector. Constraints concerning rotational differences between enrollment and verification templates are subdued by guessing a secret for every rotation compensated template. The number of selected reliable components influences the size of the secret that refers to an individual. The database that was used for experiments, consists of binary iris templates (IrisCodes). These binary iris templates are derived from iris images that are available in the public CASIA database. Originally, this database consists of 108 different individuals and 7 iris images per individu. After
transforming all iris images to IrisCodes, only 63 people had at least 5 binary templates available. And 47 people had at least 6 and 25 people had at least 7 available templates. The different database sizes configure three databases: A, B and C respectively. These databases are rather small, so further research could be done if the system also works on a very large population. The optimal efficiency is achieved on database C. Using 5 templates as training data and 2 templates as test data. In case 255 reliable components are selected from the iris templates in database C, 63 secret bits can be extracted against an False Rejection Rate (FRR) of 1.24% and False Acceptance Rate (FAR) of 0%. Which is close to the performance of the unprotected system. For the same database C is holds that configuring \( N = 511 \), 93 bits can be extracted, giving FRR=0.38% and FAR=0%.

Information that is needed to verify an individual at a later point in time should be stored in a database or smartcard. These data include a hashed secret \( S \) and helper data \( W_1 \) and \( W_2 \). The stored data does not provide sufficient information to make a successful impersonation attempt.

### 8.2 Recommendations

Choosing 511 reliable components appears to be a promising choice. Unfortunately, due to the image quality of the used database, no reliable estimation on the system performance could be made. Further investigation has to be done on this subject, using a larger database with a better image quality standard. Furthermore, only binary valued templates are used in the experiments. Much information is lost in the phase demodulation step. Using real valued iris templates might give a gain to the system performance. Finally, the mask vectors associated with each iris template appeared to be one of the biggest challenges to conquer on integration of Daugman’s iris recognition system into the template privacy protection scheme. Further research may find better solutions on error correction with use of mask vectors that are associated with biometric templates.
Appendix A

Error Correction Code (BCH)

An Error-Correction code is used to correct errors when transmitting data over a noisy communication channel. To be able to receive the correct message $m$, redundancy is added to the message before transmission. This results in a codeword $C$ that is longer than the message. The received codeword $C$ can be corrupted due to noise in the channel. It is possible to decode the correct message $m$, when the number of errors in the received codeword $C$, is less than the error correction capacity $T$.

A codeword can be defined as $C \epsilon \{0, 1\}^N$. Where codeword length $N$ for some integer $M > 2$, is defined as:

$$N = 2^M - 1$$  \hspace{1cm} (A.1)

The message length $K$ is a positive integer with $K < N$. Not all positive integers for parameter $K$ less than $N$ are valid choices. The set of codewords, used in a error correction code for messages with a length $K$ bits, contains $2^K$ codewords. Table A.1 shows all valid values for $K$ with the corresponding codeword length $N$. 
<table>
<thead>
<tr>
<th>N</th>
<th>K</th>
<th>T</th>
<th>N</th>
<th>K</th>
<th>T</th>
<th>N</th>
<th>K</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>4</td>
<td>1</td>
<td>255</td>
<td>199</td>
<td>7</td>
<td>511</td>
<td>358</td>
<td>18</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>1</td>
<td>191</td>
<td>8</td>
<td>349</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>187</td>
<td>9</td>
<td>340</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>179</td>
<td>10</td>
<td>331</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>26</td>
<td>1</td>
<td>171</td>
<td>11</td>
<td>322</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>163</td>
<td>12</td>
<td>313</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>155</td>
<td>13</td>
<td>304</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>147</td>
<td>14</td>
<td>295</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>139</td>
<td>15</td>
<td>286</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>57</td>
<td>1</td>
<td>121</td>
<td>18</td>
<td>277</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>45</td>
<td>3</td>
<td>115</td>
<td>21</td>
<td>259</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>36</td>
<td>5</td>
<td>107</td>
<td>22</td>
<td>250</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>30</td>
<td>6</td>
<td>99</td>
<td>23</td>
<td>243</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>24</td>
<td>7</td>
<td>87</td>
<td>26</td>
<td>239</td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>18</td>
<td>10</td>
<td>79</td>
<td>27</td>
<td>220</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>11</td>
<td>71</td>
<td>29</td>
<td>211</td>
<td>35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>13</td>
<td>63</td>
<td>30</td>
<td>202</td>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>15</td>
<td>55</td>
<td>31</td>
<td>193</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>120</td>
<td>1</td>
<td>47</td>
<td>42</td>
<td>184</td>
<td>45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>106</td>
<td>3</td>
<td>37</td>
<td>45</td>
<td>166</td>
<td>47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>99</td>
<td>4</td>
<td>29</td>
<td>47</td>
<td>157</td>
<td>48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>92</td>
<td>5</td>
<td>21</td>
<td>55</td>
<td>148</td>
<td>53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>85</td>
<td>6</td>
<td>13</td>
<td>59</td>
<td>139</td>
<td>54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>78</td>
<td>7</td>
<td>9</td>
<td>63</td>
<td>130</td>
<td>55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>71</td>
<td>9</td>
<td>502</td>
<td>1</td>
<td>121</td>
<td>58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>10</td>
<td>493</td>
<td>2</td>
<td>112</td>
<td>59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>50</td>
<td>13</td>
<td>475</td>
<td>4</td>
<td>94</td>
<td>62</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>43</td>
<td>14</td>
<td>466</td>
<td>5</td>
<td>85</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>36</td>
<td>15</td>
<td>457</td>
<td>6</td>
<td>76</td>
<td>85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>29</td>
<td>21</td>
<td>448</td>
<td>7</td>
<td>67</td>
<td>87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>22</td>
<td>23</td>
<td>439</td>
<td>8</td>
<td>58</td>
<td>91</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>27</td>
<td>430</td>
<td>9</td>
<td>49</td>
<td>93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>31</td>
<td>421</td>
<td>10</td>
<td>40</td>
<td>95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>255</td>
<td>247</td>
<td>1</td>
<td>412</td>
<td>11</td>
<td>31</td>
<td>109</td>
<td></td>
<td></td>
</tr>
<tr>
<td>239</td>
<td>239</td>
<td>2</td>
<td>403</td>
<td>12</td>
<td>28</td>
<td>111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>231</td>
<td>231</td>
<td>3</td>
<td>394</td>
<td>13</td>
<td>19</td>
<td>119</td>
<td></td>
<td></td>
</tr>
<tr>
<td>223</td>
<td>223</td>
<td>4</td>
<td>385</td>
<td>14</td>
<td>10</td>
<td>121</td>
<td></td>
<td></td>
</tr>
<tr>
<td>215</td>
<td>215</td>
<td>5</td>
<td>376</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>207</td>
<td>207</td>
<td>6</td>
<td>367</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.1: Table BCH error correction codes
References


