Iris–Based Biometric Cryptosystems

Diplomarbeit

zur Erlangung des Diplomgrades an der
Naturwissenschaftlichen Fakultät der Paris-Lodron
Universität Salzburg

vorgelegt von
Christian Rathgeb
crathgeb@cosy.sbg.ac.at

Betreuer:
Ao.Univ.-Prof. Dr. Andreas Uhl

Department of Computer Sciences
University of Salzburg
Jakob Haringer Str. 2
5020 Salzburg, AUSTRIA

Salzburg, im November 2008

Abstract

As a result of the growing interest in biometrics a new field of research has emerged, entitled "biometric cryptosystems". Within biometric cryptosystems the advantages of biometric authentication are introduced to generic cryptographic key management systems to enhance security. Only few work, which tends to be very custom-built according to the range of application, has been published in this area. Per contra, this work provides a general overview of fundumentals, according to biometrics as well as cryptography, and literature concerning this new area of research.

Furthermore, this work gives a detailed practical insight into the construction of biometric cryptosystems based on iris biometrics. Two different ways of constructing biometric cryptosystems will be examined to build up several schemes. Finally, performance evaluations of these iris-based biometric cryptosystems will be discussed.

Acknowledgements

First of all, I want to thank Andreas Uhl, the head of the WaveLab group at the Computer Sciences department of the University of Salzburg, for his guidance and constructive discussions during the development of my diploma thesis.

I want to thank my family, girlfriend and all my friends for emotional backing, especially Dominik Gruber for the interesting talks during coffee breaks.

I want to thank all my colleagues at the Computer Sciences department of the University of Salzburg and the members of the WaveLab group, particularly Elias Pschernig for introducing me into the iris recognition software of the WaveLab group.

This work was funded by the Austrian Science Fund (FWF).

Contents

Chapter 1

Introduction

Taking into account today's ever-increasing demand on high security standards, in order to secure any kind of crucial information, the science of cryptography has become even more important. However, in generic cryptographic systems user authentication is still possession based [87]. This means the possession of a cryptographic key suffices to authenticate a user. In most cryptographic key management systems these keys are released by presenting a password (or PIN) – chosen by the user – to the system. This implies the cryptographic key is just as secure as the password which is used to release it and these passwords are often chosen weakly as is all too well known. Additionally, a physical token such as a smartcard can be lost or stolen.

The security of such cryptographic systems can be strengthened by introducing biometrics to replace password-based authentication. Biometrics is the science of measuring and analyzing human characteristics. There are certain human characteristics which are suitable to be used in online or offline authentication systems. These comprise fingerprints, eye retinas and irises, voice patterns, signatures, facial patterns and hand measurements to mention just a few. Adequate apparatuses are used to acquire each of these human characteristics. Subsequently, information in terms of features is extracted. These features are then compared with others, which have previously been stored in a so-called enrollment procedure, and a user is accepted or denied otherwise. Due to the fact that human characteristics tend to be very distinct (these cannot be lost or forgotton either) these seem to be suitable and sufficiently secure for the purpose of user authentication and identification. One could imagine to withdraw money from an ATM by presenting a fingerprint to an adequate scanner to provide secure authentication instead of inserting a four-digit PIN presenting. This would be a prime example for combining biometrics with cryptography in a key management system.

Although biometric techniques provide promising results with respect to the identification accuracy, simply replacing password-based authentication through biometric authentication does not suffice. On the contrary, new issues emerge from the use of biometric authentication in cryptographic systems. For example, the introduction of biometrics always implies sharing personal information with a second party which has to be trusted [12]. This second party could make use of this personal information from which another security leakage concludes. Furthermore, biometric information has to be stored in databases in a secure

way which is rarely the case. Using different biometric templates for different applications is required as well [59]. Considering such critical issues the marriage of biometric and cryptographic techniques becomes non-trivial.

A new field of research has been established to attend to these issues which goes by the name of "biometric cryptosystems". Until now only a fistful of scientific articles attending to biometric cryptosystems have been published. Approaches of how to generate cryptographic keys out of biometric measurements have been proposed [6, 20, 81]. Furthermore, techniques of how to hide and retrieve user specific cryptographic keys in and out of a biometric information [35, 36] and how to generate several different forms of biometric templates from a single biometric measurement have been suggested [59, 60]. Although some of these publications may be aimed at different issues, all publications try to merge cryptographic and biometric techniques in one way or another.

This work provides an exhaustive overview of literature concerning biometric cryptosystems which comprises a meaningful classification of existsing systems by abstracting the aim of these. This turns out to be challenging because over a period of time several notations have been established to describe biometric cryptosystems, additionally different human characteristics are used in different approaches. Furthermore, differences and similarities of proposed techniques are discussed with respect to all relevant parameters of a cryptographic system.

Subsequently a more practical insight into the area of biometric cryptosystems is given. Using the iris as one of the most unique human characteristics two different approaches are pursued to implement several iris-based biometric cryptosystems. Applied iris recognition algorithms are explained in detail. This includes the preprocessing, feature extraction and matching process of these algorithms. Additionally, prior generic aspects, preliminaries and preconditions concerning these approaches are discussed in detail, which is rarely done in existing literature. Finally, performance evaluations of these systems are analyzed and compared to each other.

Chapter 2

Biometrics

A biometric is defined as a unique, measurable, biological characteristic or trait for automatically recognizing or verifying the identity of a human being. Statistically analyzing these biological characteristics has become known as the science of biometrics. These days, biometric technologies are typically used to analyze human characteristics for security purposes.

Security applications (especially cryptographic systems) need certain private information to authenticate a persons privilege. In the science of biometrics this private information is replaced by personal information filtered out of human characteristics. These human characteristics are aquired with adequate aparatuses, analysed and distinct features are extracted. In an so-called enrollment process a user registers with the system by presenting biometric data to it. The system generates a so-called biometric template for the user and stores it in a database. At the time of authentication another biometric input is acquired, processed and compared with the previously stored template in the matching process. This form of authentication provides considerable advantages over simple password-based authentication. As a consequence of this, biometrics are combined with cryptography to enhance security.

In the following chapter first the merge of both cryptography and biometrics is pointed out (Section 2.1). Then several commonly used biometric characteristics are listed and their properties are examined (Section 2.2). Afterwards potential variations of these biometric characteristics are explained (Section 2.3). Then the fundamentals of a so-called biometric authentication system are examined in detail (Section 2.4). Subsequently the terms biometric key (Section 2.5), biometric template (Section 2.6) and biometric hash (Section 2.7) are declared. In the end of this chapter entire biometric cryptosystems are classified and it is shown how performance is measured in such a system (Section 2.8).

2.1 Merging Biometrics and Cryptography

Cryptography is a very important field in the science of computer security. Many cryptographic algorithms are available for securing any kind of information. For all traditional algorithms the security depends on the secrecy of the secret or private key when a person deploys a symmetric or a public key system, respectively. The person chooses a password that is used to encrypt the cryptographic key and this key is then stored in a database (these keys are long and random and thus hard to remember). In order to retrieve the

key back, the person enters the password which will then be used to decrypt the key. This means the authentication (decryption) in such a cryptographic system is possesion-based. The possesion of the decrypting key ensures that the user is legitimate. This is one security leakage in generic cryptographic systems which can be avoided by introducing biometric authentication.

In general cryptographic systems two different types of systems can be distiguished, namely symmetric systems, where all participants of the secret communication share the same secret key, and public key systems, where pairs of a private key and a publicly available key are used to encrypt and decrypt secret information. While systems of the first category are typically designed for efficient cipher systems, the second type is used mainly in digital signatures or protocols to securely exchange secret session keys. In either category it is required to protect the private keys from unauthorized access. As cryptographically strong keys are rather large, it is certainly not feasible to let users memorize their personal keys. As a consequence of this digital keys are typically stored on smart cards or in databases and retrieved through password-based authentication as mentioned previously. Since the password is not directly tied to a person, the system is unable to differentiate between a legitimate person and an attacker. Additionally, the security of the cryptographic key is weak due to practical problems of remembering various passwords or writing them down to avoid losing data and furthermore, passwords can simply be guessed by attackers (especially those which depend on social circumstances).

Thus key management systems are the first field where biometrics can be introduced to enhance security. Several approaches have been made attending to secure password-based storage of cryptographic keys. The authentication procedure can simply be replaced through biometric authentication [35, 36]. Depending on the biometric characteristic which is used to retrieve the key the level of security is increased. Another way of introducing biometrics would be to strengthen the already existing password by means of biometrics to form a kind of two-factor authentication system [49, 52] instead of replacing the password. A biometric input can even be used directly to generate a cryptographic key [6, 20, 81] or a biometric hash [77, 79, 90] out of it.

In conclusion, key management is the major point in the science of cryptography where biometrics can be applied to enhance the security of the system. Still there are several ways in which biometrics can be used to build a cryptographic key management system which implies there are many different types of biometric cryptosystems and these use different types of human traits. In the following subchapter most of these will be discussed in order to give an overview in how to use these different types of human characteristics for security purposes.

2.2 Biometric Characteristics

There are several biometric characteristics for various applications. However, each of these biometrics has its strengths and weaknesses and therefore the choice of the biometric depends on the application [87]. Furthermore, to make use of these biometrics one must figure out which human characteristics are the most suitable for the required application and how to use the features these characteristics provide. Each of these biometrics are acquired using different apperatuses [88]. Therefore the matching process of a biometric authentication has to be adapted to the biometric characteristic. Additionally, the choice of the biometric input

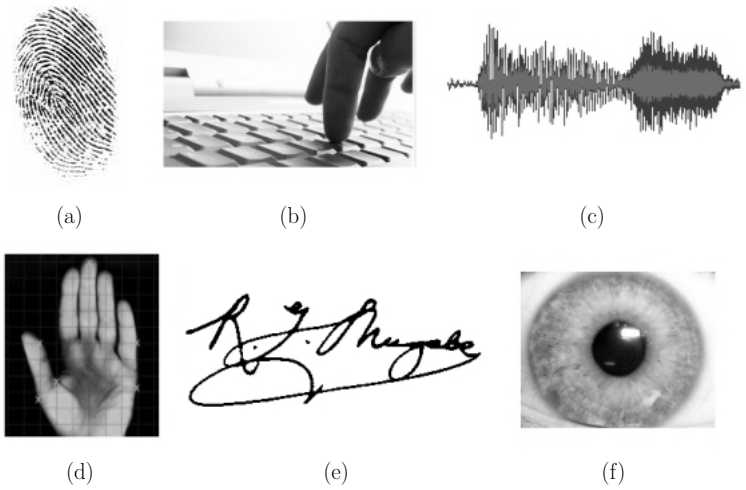

Figure 2.1: Examples of physiological (static) and behavioral (non-static) biometrics: (a) fingerprints (b) keystroke dynamics (c) voice (d) hand geometry (e) signature (f) iris.

has an influence on several magnitudes such as the performance of the whole system. Over the years several characteristics have been established, which can be classified as follows:

1. **Physiological (static) characteristics**: These are characteristics related to the nature of the human body such as fingerprints, face, hand geometry or the iris. In other words, these are characteristics a person can hardly influence. Furthermore, these characteristics do not change over time. These characteristics only change under special circumstances, for example injuries could change a person's hand geometry.

2. **Behavioral (non-static) characteristics**: These are characteristics related to the behavior of a person such as signature, voice or keystroke dynamics. These characteristics do change slightly over time, for example the signature of a person could change over the years. One can imagine that behavioral characteristics are more easily to forge, for example, an imposter could learn to imitate a person's signature.

In the following subsections the properties of some physiological as well as behavioral characteristics are outlined. Most commonly used biometrics are shown in Figure 2.3. Subsequently, the way of how these biometric characteristics are acquired, and which features of these characteristics can be extracted, is explained.

2.2.1 Fingerprint Recognition

To start with physiological characteristics, fingerprints are the oldest traits which have been used for more than a hundred years. In fingerprint authentication systems mostly friction minutiae-based features are used while systems are rarely designed to use an entire image of a fingerprint [48, 58]. Therefore the result of a common fingerprint authentication system's data acquisition (scan) would be a set of minutiae points. These so-called minutiae are skin ridge impressions of fingers which only slightly change over time. These minutiae points serve as biometric features and are compared to each other in the matching process. This

means there is a whole set of features which has to be compared to another set of features while it is not sure if during the capturing of a persons fingerprint the whole set of features is recorded or just a small subset due to bad quality of the fingerprint (if hash functions are involved, this circumstance becomes a critical issue which will be pointed out later).

The main difficulty within fingerprint biometrics is the inability to somehow normalize fingerprint data, for example, by finding specific fingerprint orientation and its center. If fingerprint data is not normalized, then all calculations resulting out of minutiae are destined to be orientation/position-dependent. The way to overcome this difficulty is to have the matching algorithm deal with transformations of fingerprint data. Much work has been done to solve the problem of aligning fingerprint images including the use of high curvature points and orientation lines.

Annother challenge is to deal with low quality images of fingerprints, which in the worst case do not include distinct features (minutiae points) necessary for the matching process. Enabling a system to authenticate a person if only a subset of features are captured during the acquisition of the fingerprint is still a topic of research.

2.2.2 Iris Recognition

Another physiological characteristic is a person's iris, the sphincter around the pupil of a person's eye. Data acquisition is performed with a special camera (iris scanner) which is able to capture the iris of a person's eye [7]. Thus one disadvantage of using iris scans for authentication is that all persons to be authenticated have to fully cooperate with the system.

Breakthrough work to create iris recognition algorithms required for image acquisition and matching were developed by J. G. Daugman [18, 19], University of Cambridge Computer Laboratory. Daugman's algorithms for which he holds key patents are the basis of all today's commercially used iris recognition systems. The algorithm of filtrating information out of such an image of a person's eye involves several steps, which can be summarized as follows: First the iris has to be extracted out of the whole image of the person's eye. Therefore the center of the iris and the inner and outer boundaries have to be detected. This detection has to be performed carefully because of the dynamic dimension of the pupil and dilation of the person's eyelid. To solve this problem Daugman proposed a method called "exploding circles". The main idea of this concept is that there are strong changes of brightness in the image at these boundaries which can be detected using cicular integrals. In the beginning an initial center of the pupil is approximated. Then circular integrals are calculated. The derivation of such integrals is very high at the boundaries where the brightness changes drastically. So applying this method for an approximated center, radii to the boundaries between the pupil and the iris and between the iris and the sclera are calculated. These radii are then used to compute a new center of the pupil and the whole method is applied again until convergence is achieved.

In the next step so-called "analysis bands" are defined for the extracted iris (in form of a ring). These bands are used to position points which are then explored using 2D Gabor filters. These 2D Gabor filters are designed to denoise the acquired signal. This process must not be confused with the smoothing of a signal. Then iris ring is unwrapped by mapping polar-coordinates to cartesian-coordinates which results in a rectangular image. In the rectangular image the radii of the previously defined analysis bands is fixed and every explored point is a center of a 2D Gabor wavelet. For this wavelet the coefficients are generated out of which

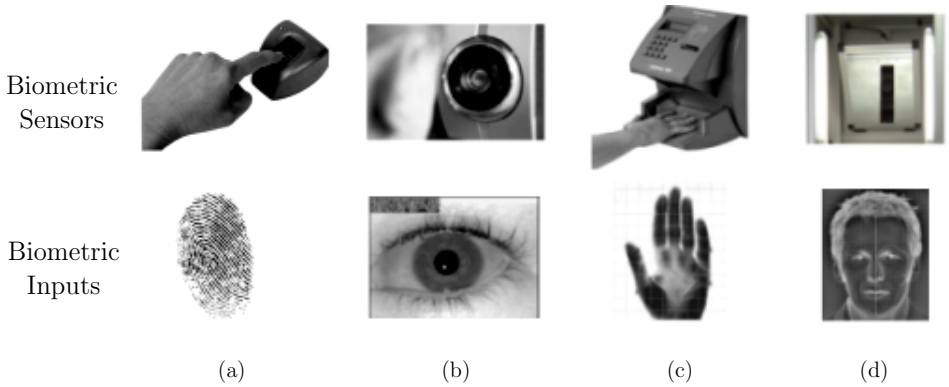

Figure 2.2: Examples of biometric sensor and the according biometric inputs for physiological (static) biometrics: (a) fingerprints (b) iris (c) hand geometry (d) face.

two bits are extraced. This method is applied again until enough bits are extracted. The rectangular image of the iris mostly includes some part of the eyelid and eyelashes. Eyelashes are seen as noise which have to be detected during the unwrapping of the iris. This is done by calculating a bit-mask where one bit represents a region of the iris and is set to 0 if any noise is detected and otherwise to 1. Other forms of noise could be, for example, camera pixel noise or specular refletions.

The result of the whole procedure is a so-called iris-code (for example 2048-bit long in J. G. Daugman's approach). This iris code serves as a biometric template which can be stored in a database together with the bit-mask.

After the extraction of the iris-code the matching process can be performed using several metrices, for example, the Hamming distance. This could be easily done by just bitwise XORing two iris-codes and comparing the number of mismatching bits to a specific threshold.

2.2.3 Face Recognition

In a face recognition system images of the whole face of a person are captured out of which unique key features are extracted to identify persons reliably [14, 85]. The acquired set of key features include relative distances between characteristics such as eyes, the nose, the mouth cheekbones and the jaw. Using all of this information a unique template is created by applying dimension reduction. In generic face recognition systems this is done by applying, for example, Eigenfaces [85]. This template may then be compared to databases of facial images to identify a person.

While a face-recognition system has high acceptance, its accuracy is low [25]. The problem arises mainly from three factors: insufficient capability of representing features in the feature space and within-class and between-class variations. Most face recognition systems are highly sensitive to variance of a person's face. Unfortunately there is plenty of variation, for example small movements of the head or changing haircuts. Thus, dimensionality reduction is performed to improve the capability to represent features and harmonizing the image taken.

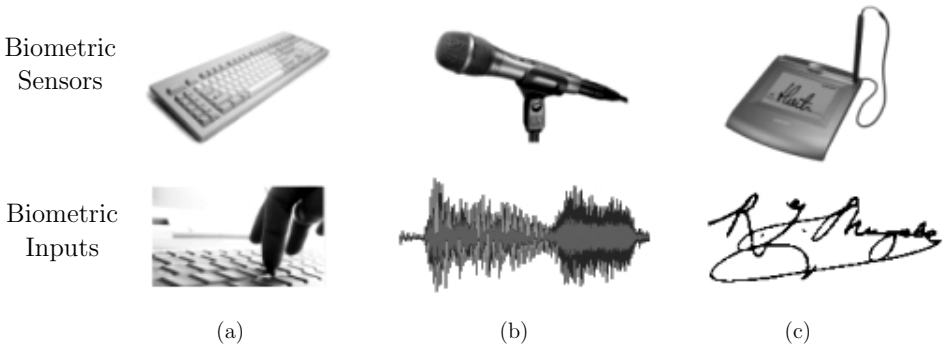

Figure 2.3: Examples of biometric sensor and the according biometric inputs for behavioral (non-static) biometrics: (a) keystroke dynamics (b) voice (c) on-line signatures.

2.2.4 Hand Geometry

Hand geometry is a biometric that identifies users by the shape of their whole hand. For data acquisition so-called hand geometry readers are used to measure a user's hand along many dimensions [62, 63]. These measurements serve as features which are used to authenticate a user comparing these against previously stored ones.

Since hand geometry is not thought to be as unique and widespread as fingerprints, fingerprinting remains the preferred technology for high-security applications due to various forgery opportunities such as changing lengths of fingers with caps. Thus it is advisable using hand geometry combined with fingerprints to form a so-called two-factor authentication system.

2.2.5 Speaker Recognition

One biometric characteristic which tends to be very difficult to handle is voice. On the one hand voice biometrics are behavioral characteristics because it depends on the way a person talks (pronounciations, volume of speech). On the other hand the voice can be seen as a physiological characteristic of a person as well.

Data acquisition is first performed during the enrollment process where a person utters a password or passphrase to a device (usually a microphone) when prompted to do so. This signal is digitalized with an analog to digital converter and subsequently analysed resulting in a so-called voice model of a person [10]. In the authentication process the repeated utterance of the same password by the same person should authenticate a legitimate user. If a correct password is necessary the whole system is called token-based [49, 50]. Uttering an incorrect password (token) the user will be rejected.

Solving this difficulty voice authentication would offer many facilities, for example a person could be identified during a phone call. However, a forgery could still attempt to record a person to gain possession of a password. Voice is one human characteristics where the temporal order of the feature is important which is typical behavioral biometric characteristics.

2.2.6 Signature Verification

Speaking of signatures as a biometric characteristics of a person one has to distinguish between so-called "off-line" and "on-line" signatures:

Off-line signatures are for example signatures on documents where nearly only "spatial information" such as features of curves can be analyzed to identify a specific person, which is very unsatisfying. This is because off-line signatures refer to the result of a complete writing process. In other words there is no information but the raw image of the signature. This image can be modified with the technique of dynamic time warping to correlate the result with other acquired off-line signatures [26]. Furthermore, shape-matching can be performed.

On-line signatures are signatures which are acquired using a palm or tablets. Therefore access to signals during the writing process, so-called temporal information, is demanded [9, 32, 54]. This means using on-line signatures the physical activity of signing is mearsured and analysed. By doing so many additional signals are offered which can be analyzed to identify a person. These signals include the position of the pen, the time, the angle of the pen and the pen pressure. Additionally the analog-digital conversion is performed. Some important features calculated out of these signals are for example the number of pen-ups/downs, the average of absolute writing acceleration in y-direction, the effective average writing velocity in x-direction and the time it took the person to sign [88]. Thus there are plenty more features to analyze with on-line signatures.

2.2.7 Keystroke Dynamics

Keystroke dynamics is another behavioral biometric characteristic which could be additionally used to identify a person [52, 56]. For data acquisition the keyboard serves as biometric sensor with which two events, the "key down" and "key release" event, are measured. Every user develops a specific timing pattern when typing a password called keystroke dynamics. Duration and latencies of a user's keystroke dynamics are measured by the authetication system to enhance security. This means the correct password is necessary but does not suffice any more if the keystroke dynamics are measured as well. Very distinctive durations can be measured out of letters often following behind each other such as the "th" in an English password for example.

Problems occur within a system which used keystroke dynamics as a biometric characteristic when keyboards are changed or if a user suffers from a hand injury. The approach of using keystroke dynamics is a simple example for combining the knowledge of something with a biometric characteristic.

2.2.8 Hybrid Biometrics

Hybrid biometrics also called multi-factor authentication scheme is an approach to enhance security by combining two or more biometric characteristics. A weighted set of biometric characteristics of a person could be used for identification, for example the image of a person's face and a spoken password could be combined [57]. By doing so forgery attempts become nearly impossible.

Merging features of two or more biometric characteristics still seems to be a tough challenge and even more an intelligent weightage of these characteristics and how a set of authentica-

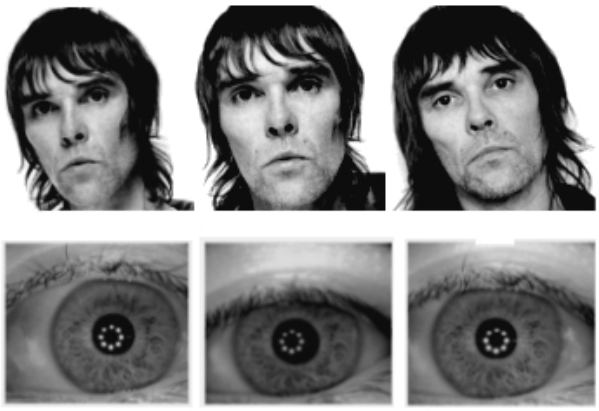

Figure 2.4: Variations of biometrics: top: variations of an individual's face image due to pose and lightning. bottom: variation of an individual's iris image due to lightning and diltation of eyelids.

tion processes are combined to form one hybrid system. Furthermore, hybrid biometrics can be used if one of the biometric characteristics is not available from a person, for example, due to an accidental injury so that there have to be other ways for authentication.

2.2.9 Other Biometric Characteristics

There are plenty more biometric characteristics such as physiological characteristics like a person's retina or DNA or behavioral characteristics like gait or lip movement [87]. However on these biometric characteristics only little research has taken place in terms of combining these characteristics with an cryptographic authentication system.

2.3 Biometric Variance

In a password-based authentication system an accurate matching is no problem because the result is perfectly calculated, therefore the matching process is not difficult to engineer. In a biometric authentication system two measurements of the same person's biometrics cannot be expected to be equal.

Taking a look at the above biometric characteristics one can imagine how all of these biometrics can vary from one measurement to another depending on the biometrics properties. For instance, a person's haircut could drastically change the output of a face recognition system [14]. Variations of several human characteristics are illustrated in Figure 2.4. Thus the system should be tolerant in a way so that similar but not perfectly equal inputs are accepted. Therefore the difficulty for an authentication system lies in having to be as tolerant as to authorize legitimate users. On the other hand, if the system gets too tolerant it will perhaps accept unauthorized users as well. Thus a trade-off between the amount of fuzziness the system is able to handle and the security it provides has to be found.

There are various reasons for signal/representation variations, such as inconsistent presentation, irreproducible presentation and imperfect signal/representational acquisition pointed out in [87]. One promising way to deal with these forms of fuzziness is to find the most

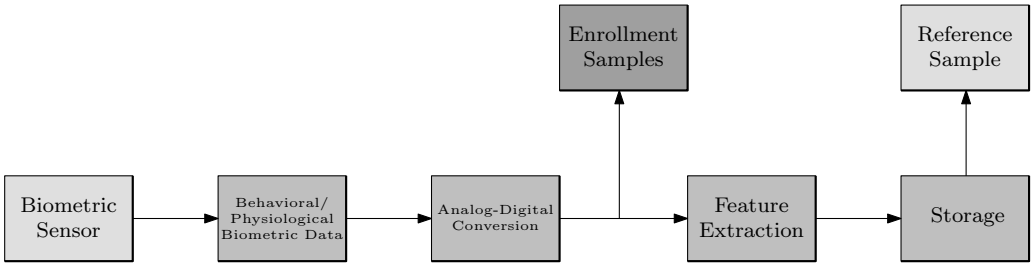

Figure 2.5: The Enrollment process: biometric data is acquired, analog-to-digital conversion is performed, features are collected and a reference sample is stored

significant biometric features of a person and use these in the authentication process.

2.4 Biometric Authentication Systems

A biometric authentication system is a system which is able to perform automated authentication of users depending on their physical and behavioral characteristics (Section 2.2). Such an authentication system consists of several basic entities:

- **Biometric Sensor**: the biometric sensor performs the data acquisition and therefore the analog to digital conversion. The output of the biometric sensor are the raw biometric data. The sensor is used at the enrollment of a user and every time a user needs to be authenticated.

- **Feature Extraction**: in the feature extraction the raw data are processed and analyzed. The result of the feature extraction should be the most distinctive features for every user. Feature extraction is performed during the enrollment process as well as during an authentication.

- **Database**: in almost all biometric authentication systems a database is required. This database is used either to store cryptographic keys which are released when an authorized user represents biometric features to the system or the database could store raw biometric features or a hash value of these.

- **Matcher**: the biometric matcher is responsible for the matching process which should – as mentioned before – in some way be tolerant but should not provide any security leakage. Matching is performed whenever a user needs to be authenticated.

The two basic processes of a biometric authentication system are the “enrollment” process and the “authentication” process [88]. In the enrollment phase of a biometric authentication system, all users are registered with the system, and references are stored in the database of the system. On the other hand, the authentication process denotes the process of identity verification or determination. In this phase the authentication system performs a comparison between the presented biometrics and the stored references of the previous enrollment phase.

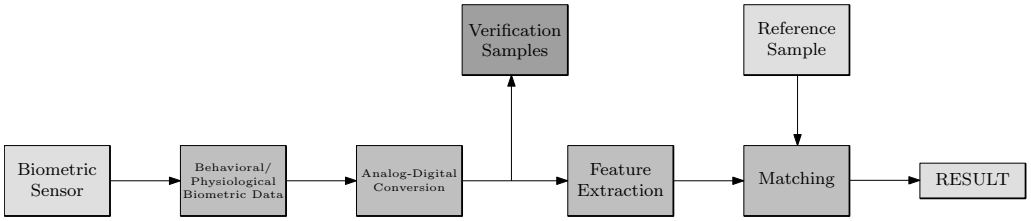

Figure 2.6: Authentication process: collected data is compared to reference data

2.4.1 Enrollment

In the enrollment phase, as shown in Figure 2.5, a user's biometric data is presented to the authentication system for the first time. This analog data, depending on the biometric characteristic, then needs to be digitalized for further use. The result of this analog-digital conversion are so-called "enrollment samples". These samples are then preprocessed, features are extracted and the extracted features are then stored in databases.

In most biometric authentication systems not just one but several data acquisitions are performed during the enrollment of a single user. With the help of several biometric sample magnitudes such as standard and mean deviations of biometric features can be calculated and thus a more representative sample can be calculated or boundaries can be generated for these features. Furthermore, the acquisition of several samples can be used to filter out the most stable features where deviation values are small. Thus common biometric authentication systems demand at least two or three biometric samples of a user during the enrollment process.

2.4.2 Authentication

After a user has been enrolled the biometric authentication system should be able to authenticate the user as illustrated in Figure 2.6. Once again biometric data is presented to the system and digitalized. The obtained data, the so-called "verification samples" which are of the same (raw) data format as the enrollment samples resulting out of the enrollment phase, are preprocessed and features are extracted.

In the matching process the derived features are then compared to the template resulting from an earlier enrollment process. If the matching succeeds, then the user is authorized, otherwise the user is rejected.

2.4.3 Verification and Identification

There are two modes in which a biometric authentication system can operate, namely "verification" and "identification" [88]. In the first mode a user claims to be someone who needs to be verified and thus only an one-to-one comparison has to be performed by the biometric authentication system while in the second mode a user needs to be identified and therefore a one-to-many comparison has to be performed. Furthermore, the process of identification can be modeled as sequences of one-to-all verification and therefore the fundamental underlying mechanism is always verification.

2.4.4 Performance Measurement

Due to the fuzziness of the matching process of biometric systems several errors occur. In generic biometric verification systems there are two main types of errors:

- Misrecognizing measurements of two different persons to be of the same person, called "false acceptance".
- Misrecognizing measurements of the same person to be of two different persons, called "false rejection".

The performance of a biometric system is commonly described by its "false acceptance rate" (FAR) and "false rejection rate" (FRR). The FAR and FRR are commonly accepted and quoted in almost all publications concerning biometric authentication systems. These two measurements can be controlled by adjusting a threshold, but it is not possible to exploit this threshold by simultaneously reducing FAR and FRR [39]. Both have to be traded-off, as reducing FAR increases FRR and vice versa. For example, if an authentication scheme tends to be tolerant with respect to accepting similar biometric data, the FAR of this system will be very high while the FRR would be satisfying.

Another important performance index of a biometric system is its "equal error rate" (EER) defined as at the point where FAR and FRR are equal. A perfect system in terms of accuracy would provide an EER of zero.

Unfortunately, however, over several years of investigation, a perfect biometric verification system has not been developed. Additionally the FRR/FAR numbers quoted by biometric vendors are often unreliable. In most cases these values were calculated under unrealistic circumstances or several preassumptions were taken which do not hold in practise.

Beyond that there are some other commonly used measures for technical evaluation of a biometric system including the above measurements and furthermore, False Match Rate, False Non Match Rate, Receiver Operating Characteristic, Failure to acquire Rate and the Failure to enroll Rate. In Table 5.1 these evaluations are summarized and explained.

2.5 Biometric Keys

In the sense of cryptography a key is a piece of information with which one is able to encrypt a plaintext into a so-called ciphertext and vice versa during the decryption process.

In biometrics it is aimed at deriving a cryptographic key from one or more biometric samples during the enrollment phase of the biometric authentication system. This key may later be regenerated using another biometric sample that is close to the original samples. The basic idea of biometric-based keys is that the biometric component performs user authentication, while a generic cryptographic system can still handle the other components of containment such as secure communication [87].

The result of the data acquisition are raw biometric data which are further processed in the feature extraction step. The result of the feature extraction is a set of features called "feature vector", denoted by Φ. This feature vector Φ, consists of k features ϕ_i such that $\Phi = \{\phi_1, ..., \phi_k\}$. Every ϕ_i represents the value of a measured feature of a user.

For generating a cryptographic key of a specific length m out of this set of collected features

Measure	Description
Failure to Acquire Rate	Ratio between numbers of biometric samples which could not been correctly acquired due to some reasons and total number of acquisitions
Failure to Enroll Rate	Ratio between numbers of users which could not been enrolled correctly due to some reason and total number of users
False Acceptance Rate (FAR)	Ratio between numbers truly non-matching samples which are matched by the system and total number of tests (including to first two rates as well)
False Rejection Rate (FRR)	Ratio of truly matching samples, which are not matched by the system and total numbers of tests (including to first two rates as well)
Equal Error Rate (EER)	The point on the error rate diagrams where FAR and FRR are equivalent
False Match Rate (FMR)	Ratio between numbers truly non-matching samples which are matched by the system and total number of tests (The FMR does not include the first two rates)
False Non Match Rate (FNMR)	Ratio of truly matching samples, which are not matched by the system and total numbers of tests (The FNMR does not include the first two rates)
Receiver Operating Characteristic (ROC)	The diagram of a verification system where the FMR and the FNMR specify the x and the y-axis

Table 2.1: Overview of the most common evaluations in biometric authentication systems

it requires that there be a way of mapping Φ to a so-called "feature descriptor" b of length m: $b = (b_1, ..., b_m)$ where $b_i \in \{0, 1\}$ and $m \leq k$. This is mostly done by applying functions which match the ϕ_is of Φ against some thresholds specified by the system [49, 52].

One very simple way of generating such a feature descriptor would be to obtain the i-th bit b_i of the feature descriptor b by comparing ϕ_i to a fixed threshold and assigning b_i to be 1 or 0 depending on whether ϕ_i was less than or greater than the threshold. Nevertheless, this simple approach rarely suffices in practice.

Once feature descriptors are derived these should separate persons in the sense that descrip-

tors produced by the same user are "sufficiently similar" so that there is a small "intra-class" variation, but ones produced by different users are "sufficiently different" so that there is a large "inter-class" variation. If this property is satisfied, and the features can be reproduced reliably, then the feature descriptor could be a candidate for use as a person's biometric key. Taking a look at a person's biometric characteristics (see Section 2.2) one can imagine that the variations of these characteristics take a huge influence on this procedure and therefore the challenge in generating such keys lies in finding those features which are highly consistent from one measurement to another but still tend to be distinctive for the particular person.

Beside the property of separating users there are other important requirements for creating such a biometric key. The most important requirements can be summarized as follows [2]:

1. **Key Randomness**: The produced key should appear to be random to any adversary even if the biometric data which was used to derive the key is available to the adversary. Furthermore, similar persons whose biometric data is close should not receive similar keys. Otherwise this would give an adversary the opportunity to filter information out of several biometric data. The adversary could correlate these to find out which features are extracted and thus the adversary could learn how the key generating mechanism works. This requirement is very similar to those of hash functions.

2. **Key privacy**: An adversary should not learn any useful information about a biometric given the biometric data used to derive the key. In addition, access to the key should not help an adversary to receive any information of how the key was generated, altough this scenario is commonly not considered.

3. **Key entropy (strength)**: Instead of developing longer cryptographic keys to resist brute force attacks, a more intelligent approach might be to aggregate features and parameters from an individual in such a way that their correlation generates a key that is much stronger than the individual size of the actual key.

4. **Key uniqueness**: The uniqueness of a biometric key will be determined by the uniqueness of the individual biometric characteristics used for deriving the key. Instead of trying to find a single unique feature, a biometric key needs to find only a collection of somewhat unique features or parameters that when assembled collectively create a unique profile of an individual. The incorporation of a simple passphrase will improve the accuracy of the biometric key by incorporating "something you are" with "something you know". Furthermore, a multibiometric key could be generated by combining several biometric characteristics. This would increase the security of the generated key although it is challenging to combine several features extracted out of several biometrics into one biometric key (features have to be weighted and adequate functions have to be defined).

5. **Key stability**: A major problem with biometric authentication is that an individual's set of features collected during the enrollment phase and the reference features can vary from session to session. This variation can occur for a number of reasons including different environments according to lighting, orientation, emotional state or physical changes like facial hair, glasses or cuts. The goal is to find a set of relatively stable features for which the amount of variation can be reduced to an acceptable number of bits. If this is the case a valid user only has to search a very limited key space to recover an encrypted transmission while making a brute force search by an attacker remain difficult, if not impossible.

Often biometric keys get confused with cryptographic keys. While cryptographic keys are often combined with biometric data, a biometric key is directly derived from the biometric data. Thus not all biometric authentication systems comprise a biometric key generator. In such systems cryptography is decoupled from biometrics.

2.6 Biometric Templates

A cryptographic template is any piece of information that is stored on the system for the purpose of re-generating the cryptographic key or performing comparisons [4].

A biometric template could include several data. It could just consist of the reference samples acquired during the enrollment process, perhaps encoded in some way. Furthermore, a biometric template could be seen as the set of features extracted in the feature extraction process or the further processed features, the feature descriptor. Calculated boundaries for the deviation of biometric data could also be stored as part of the biometric template or parameters of feature extracting functions. In general terms when speaking of a biometric template it is referred to biometric data modified in some way.

For biometric data biometric templates are usually stored unprotected in a central database [84]. During authentication of a user the comparison of templates is still performed using decrypted templates even if the stored ones are encrypted. This decryption process can be compromised as well. If the biometric database is compromised and an intruder obtains a person's biometric template, using this biometric will be impossible for the rest of the person's life. Thus biometric templates have to be handled carefully.

2.7 Biometric Hash

In cryptography a hash function is a reproducible method of turning some kind of data into a (relatively) small number that may serve as a digital "fingerprint" of the data. Hash functions are designed to be fast and to yield few hash collisions in expected input domains. In hash tables and data processing, collisions inhibit the distinguishing of data, making records more costly to find. A hash function must be deterministic, which means if two hashes generated by the same hash function are different, then the two inputs are different in some way.

Hash functions are usually not injective, thus the computed hash value may be the same for different input values. This is because it is usually a requirement that the hash value can be stored in fewer bits than the data being hashed. It is a designated goal of hash functions to minimize the likelihood of such a hash collision.

A desirable property of a hash function is the mixing property: a small change in the input should cause a large change in the output. This is called the avalanche effect.

The goal of a biometric hash function is to find a function for mapping biometric features into a value space of defined dimensionality, adopting three of the key properties of cryptographic hashes [88]:

- Mapping from a (very) large value domain to a smaller value space.
- Infeasibility to find input that maps pre-specified outputs.

- Infeasibility to find any two distinct biometric signal inputs originating from any two different users, which map to the same output.

Biometric templates (see Section 2.6) do not satisfy security requirements because of several vulnerabilities. According to storing a collection of extracted features of a user's biometrics, this set of features could be transformed by a hash function H into a so-called hash value $H(\Phi)$. This transformed version of collected features $H(\Phi)$ could be stored as a so-called "private template" during the enrollment process [11]. When a user wants to authenticate to the system, again features are collected, hashed and compared to a hashed template of an earlier enrollment process. Therefore the matching process is performed in another space and if a user's template is ever compromised, a new space for the matching process can be issued by just changing the hash function H.

2.8 Classification of Biometric Cryptosystems

Over the years a commonly accepted classification of biometric cryptosystems has been established [27, 87]. Two different types of biometric cryptosystems, namely key release schemes and key generation/binding schemes, can be specified as follows:

1. **Key Release Scheme**: In a so called key release scheme the biometric authentication is completely decoupled from the cryptographic part of the system. Thus the user's key and the biometric data are independent of each other which is the major benefit of the key release approach. In the authentication process acquired biometric data is compared against a reference templated acquired during enrollment. If this comparison succeeds with respect to some applied metric, the cryptographic key is given to the user. Therefore this key could easily be modified or updated at any time in case it is compromised, which means key release schemes provide cancellable biometrics [59], which will be described later.

   This is an easy approach which could be for example realized with fingerprint biometrics. A user could present a fingerprint to the system which matches the acquired samples against a stored fingerprint template and if the matching succeeds, gives a key to the user. Nevertheless, key release schemes are not frequently used although it would be easy to implement a biometric cryptosystem with this approach because of the following vulnerabilities:

   (a) The biometric template which is not secure has to be stored in a database. This is a very critical subject because biometric templates could be stolen.

   (b) Due to the fact that authentication process and the key release are decoupled it would be possible to manipulate the biometric matching process.

   (c) The cryptographic key has to be stored as part of the template.

   Therefore a biometric cryptosystem based on the key release scheme is not appropriate for high security applications.

2. **Key Generation/Binding Scheme**: In the key generation scheme the user's key is directly derived from the user's biometric data (=biometric key) and therefore does not have to be stored anywhere. This means a user presents biometric data to the system which generates a key out of the extracted features. This key is then given to

the user who could use it to, for example, encrypt private data. The major problem seems to be that in this approach a key could not be changed if it was compromised once. To achieve the benefits of a biometric cryptosystem with cancellable keys it is common to combine the user's biometric data with the cryptographic key. For example a reference sample of a user's biometrics combined with this key could be stored as biometric template. During the authentication process biometric data is presented to the system with which the key can be detached from the stored template. A more secure environment is provided by combining the key and the template together, which on the other hand makes it a bit more difficult to implement provided that the key is intelligently mixed with the biometric data. Therefore it is harder for an adversary to get into the possession of the user's key or the biometric template because these won't appear raw in a database. Additionally the biometric matcher performs authentication and key release in a single step.

2.9 Cancellable Biometrics

Unlike simple PINs or passwords, biometric characteristics are permanently associated with a person and cannot be changed. Thus a great disadvantage comes up with biometric cryptosystems based on the classic key generation scheme. Here the cryptographic key is directly derived from a person's biometrics, which means, if the biometric data is stolen, for example by capturing images of a person's iris or by recording a person's voice, the biometric data becomes useless and is lost forever. Furthermore, biometric data becomes useless for all applications it was used because a person can potentially be tracked from one application to the next by cross-matching databases [59, 60].

To overcome this disadvantage an intermediate step has to be established, which adds secret information. A commonly used approach is to apply transform functions to the biometric data. If a transformed sample of a person's biometrics is compromised only the transform function, which is either directly applied to the biometric data in the signal domain or later when the biometric features have already been extracted, has to be changed. The most important condition these functions have to fulfill is invertability so that neither the comparison of transformed data to the raw nor the comparison to another transformed sample reveals any useful information.

2.10 Transforms and Filters

Signals do not exist without noise and therefore noise has to be reduced to proceed with further analysis [76]. Denoising of signals must not be confused with smoothing of signals, while smoothing removes high frequencies and retains low frequencies, denoising attempts to remove whatever noise is present and retain whatever signal is present.

Biometrics are noisy, which means that two measurements of the same biometric characteristics, for example, capturing the image of a person's face, will not result in the identically same data. So denoising functions are used to denoise biometric signals with respect to, for example, observing several facial expressions in a face recognition system. These functions, for example wavelets, work as feature filters and obtain good results in the biometrics [30].

2.11 Error Correcting Codes

Another way to repress the fuzziness of biometric measurements is to introduce error correcting codes. The goal of an error correcting code is to transmit a message m through a noisy communication channel so that no information is lost [36]. To achieve that, m is mapped to a longer string c with the property of correcting single bit errors up to a specific threshold depending on the length of c. With the ability of error correction the receiver can reconstruct c out of a received c' and thus is able to calculate the intact message m.

More formalized an error correction code contains a large set of codewords $C \subseteq \{0,1\}^n$. If an l-bit message m has to be transmitted, where $l < n$ is necessary, a function $g : M \rightarrow C$ has to be defined. Here $M = \{0,1\}^l$ represents the message space while g is a one-to-one mapping from messages of M to codewords of C. Furthermore, a decoding function $f : \{0,1\}^n \rightarrow C \bigcup \{\emptyset\}$ is needed which maps a codeword of the received message to its "nearest" codeword in C or if this is not possible, puts out $\emptyset$ to indicate decoding failed.

Error correcting codes can be used to overcome the fuzziness of biometric measures as well: for example, denote $b \in \{0,1\}^n$ a feature descriptor resulting out of a biometric measurement and $k \in \{0,1\}^l$ a cryptographic key with which the feature descriptor should be combined. First the representation of k in C is calculated, $k' = g(k)$, and afterwards the bitwise XOR of k' and the feature descriptor b, $\hat{b} = b \oplus k'$, is computed. The XORing of these two bitstreams represents the binding of the cryptographic key with the user's biometrics. This is first done during the enrollment process and the combination of the user's biometrics and the key (prepared with an error correcting code) is then stored in a database as biometric template. During the authentication process a person presents biometric data to the system, a feature descriptor b' is extracted and $b' \oplus \hat{b} = b' \oplus b \oplus k' = k''$ is calculated. In the end the decryption function f is applied and if $f(k'') = k$ the authentication succeeds. Thus the error correcting code is used to overcome some bit errors which were produced due to the variance in the captured biometrics. This means, it is assumed that if two biometric measurements are from the same person, these are sufficiently similar so that only some bit errors occur, which can be corrected with common error correcting codes.

One class of most commonly used error correcting codes are so-called "Reed-Solomon" codes [33]. Reed Solomon codes are well-established codes based on polynoms, where message of length l to be encoded is represented as the evaluation of a polynom of degree $l-1$ (see Section 4.4.2). Another class of error correcting codes are so-called "Hadamard Codes" [3, 47]. These Codes use special kind of matrices to detect/correct single bit errors (see Section 4.4.1).

Chapter 3

Literature on Biometric Cryptosystems

This chapter summarizes published achievements with respect to generating cryptographic keys out of biometric data, secure storage of biometric templates and other related topics. In the first subsection (Section 3.1) the first and classic algorithm to derive a key from a person's biometrics called "Biometric Encryption" is discussed. Subsequently the so-called "Private Template" scheme (Section 3.2) is presented where the biometric template itself or a hash value of it is used as a cryptographic key. The combination of "knowing something" and the behavioral characteristics of a person for authentication purpose is described in the a technique called "Password Hardening" (Section 3.3). Then one of the most important milestones namely the "Fuzzy Commitment" scheme and an improvement of it with respect to the alignment of biometric data called "Fuzzy Vault" are described (Section 3.4) and afterwards several papers resulting from these schemes, which deal with constructs called "Fuzzy Extractors" and "Secure Sketches" (Section 3.5) are summarized. Accordingly, an overview of a very new technique called "BioHashing" (Section 3.6) is given. Furthermore, biometric cryptosystems which define intervals for the matching process (Section 3.7) and the approach of Cancellable biometrics (Section 3.8) are summarized. At the end of the chapter some other related work (Section 3.9) is discussed which does not fit into one of these schemes and a discussion (Section 3.10) to classify the described approaches is given.

3.1 Biometric Encryption

In the following the first approaches of using biometric together with cryptography, resulting in an algorithm called Biometric Encryption^{TM} are described. The goal of this algorithm is to provide a mechanism for the linking and subsequent retrieval of a digital key using a biometric such as a fingerprint.

In this algorithm a filter function is generated with the use of correlation which should consistently produce the same output pattern for a legitimate user. This output pattern is linked with a cryptographic key during the enrollment process using a so-called linking algorithm which generates a look-up table. Furthermore out of the key an identification code is generated and stored together with the filter function and the look-up table. If the

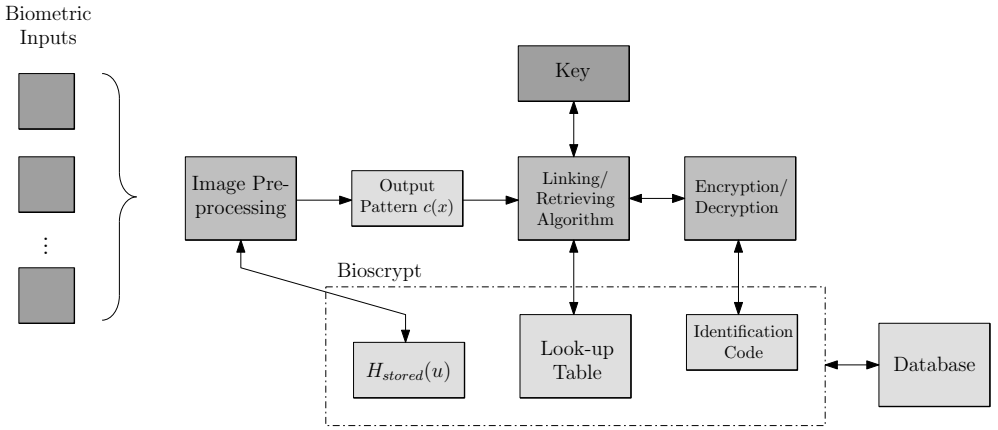

Figure 3.1: The basic operating mode of the Biometric Encryption^{TM} algorithm

user wants to retrieve this key the user's biometric is captured, another output pattern is generated and a key is retrieved via a look-up table. Out of the retrieved key another identification code is generated and matched against the stored one. The whole process of the Biometric Encryption^{TM} algorithm is displayed in Figure 3.1.

The prior idea of generating a personal cryptographic key out of a person's biometrics was presented in a German patent by Bodo [6]. In Bodo's approach a key was directly derived from a person's biometrics, thus this method would function as a pure key generation system including the disadvantage of not being able to update the key, in case a person's biometric was comprised.

The original concept of the Biometric Encryption^{TM} algorithm was published by Soutar *et al.* [81]. The patent comprises a description of how a whole public key cryptosystem, based on fingerprint biometrics, should look like. In an enrollment phase a person's fingerprint and a unique random number (which is not further described) are combined. This is done by applying a filterfunction, which is the Fourier transform of the fingerprint (a 2D grayscale image) and furthermore the unique random number is generated out of the coefficients of the transform. The goal is to retrieve this unique number each time an authentication takes places. It is suggested that out of this number a public and a private key are generated to form a public key cryptosystem. In the end of the enrollment phase information about the filter function is stored on a smartcard. For the generation of the public and the private key it is suggested that the enrolled person has to insert the card into some kind of apparatus. Furthermore, fingerprints are scanned by a fingerprint scanner. The apparatus then generates an optical Fourier transform of the scanned fingerprint. The Fourier transform signal is incident on to a spatial light modulator programmed with the filter information from the card. The inverse transform is generated out of the filtered signal and furthermore, used to regenerate the unique number. In the end of the key generation phase the person receives a public and a private key (the generation of these key out of the unique number is not further explained).

This was the first practical approach of generating updateable cryptographic keys out of a unique number and a person's biometrics. Altough in the description of the system many

things are left open, the elementary idea of applying functions to fuse biometrics with random numbers and just storing these function parameters to regenerate the numbers is presented.

In several publications Soutar *et al.* [12, 69, 70, 71] improved the above mentioned concept and described the Biometric Encryption™ algorithm in detail: During data acquisition entire fingerprints are processed instead of using significant features of a fingerprint like in feature-based biometric systems. The basis of the whole algorithm is the mechanism of correlation which tends to be an effective mechanism for determining the similarity of biometric data. In the following a two-dimensional image array is denoted by $f(x)$ and its corresponding Fourier transform is denoted by $F(u)$, which serves as filterfunction. At the time of enrollment a filter function, denoted by $H(u)$, is derived from $f_0(x)$ (0 indicates the first measure). Subsequently, a correlation function $c(x)$ between the initial measurement $f_0(x)$ and any other input $f_1(x)$ obtained during verification is defined by $c(x) = FT^{-1}\{F_1(u)F_0^*(u)\}$, the inverse Fourier transform of the product of the Fourier transform of a biometric input and $F_0^*(u)$, where $F_0^*(u)$ is typically represented by $H(u)$ (FT^{-1} is the inverse Fourier transform). The output $c(x)$ is an array of scalar values describing the degree of similarity. To provide a degree of distortion tolerance, the filter function is calculated using a set of T training images $\{f_0^1(x), f_0^2(x), ..., f_0^T(x)\}$. The output pattern of $f_0^t(x)$ is denoted by $c_0^t(x)$ with its Fourier transform $F_0^t(u)H(u)$. The filter function $H(u)$ represents a part of the biometric template which is termed *Bioscrypt*.

To enhance security only a modified version of the filter function $H(u)$ denoted $H_{stored}(u)$ is used as part of the biometric template. This is done by just storing the phase component $e^{i\phi(H(u))}$ of the function. The output pattern $c_0(x)$ is then linked with a N-bit cryptographic key k_0 using a linking algorithm. The key k_0 is generated using a RNG. Then a part (N bits) of $c_0(x)$ are binarized using some threshold (commonly zero). The linking is performed as follows: if the n-th bit of k_0 is 0 then L locations of the selected part of $c_0(x)$ which are 0 are chosen and the indices of the locations are written into the n-th column of a look-up table. This look-up table is stored as part of the Bioscrypt for the authentication process. During the linking redundancy is added by applying a repetitive code (an error correcting code could also be used here). Finally a combination of standard encryption and hashing algorithms is used to derive an identification code denoted by id_0 from the key k_0. The identification code id_0, which can be seen as hash of k_0, is stored as part of the Bioscrypt as well.

During the authentication process a set of biometric images is combined with $H_{stored}(u)$ to produce an output pattern $c_1(x)$. A so-called retrieval algorithm, which represents the inverse analogon of the linking algorithm, calculates an N-bit key k_1 with the use of the previous created look-up table. This is done by using the look-up table to extract the constituent bits of the binarized output pattern. For the n-th element of k_1 the sum of the L bits of the binarized output pattern whose indices are specified by the n-th column of the look-up table is calculated. The n-th element of k_1 is set to 1 if the sum of these bits is greater than $L/2$ otherwise it is set to 0. How to choose the value L and the L locations for each bit of the key is not explained further. Then the same standard encryption and hashing algorithms as during the enrollment process are applied to produce id_1, which is compared to id_0 to check the validity of k_1. If both keys are the same authentication succeeds.

Soutar *et al.* accomplished their Biometric Encryption algorithm in patent [72], which also includes explanations of how to use the algorithms for other biometric characteristics such as the iris. In conclusion, the Biometric Encryption™ algorithm is an algorithm for the

linking and retrieval of digital keys in which a cryptographic key is created, independent of the biometric system. It focuses on three critical issues, namely discrimination capability, distortion tolerance and security. While the first two requirements are satisfied by the use of correlation and the number of input images, the last is achieved by the intelligent storing of the biometric template. Unfortunately in all the publications performance measurements and test results are renounced.

Adler [2] provided an insight into image reconstruction from biometric templates and described a potential vulnerability in the above presented scheme which would offer a less than brute force regeneration of the secret and an estimate of the enrollment image. The weakness could be utilized during the comparison process to calculate an analogon for the match score using a so-called "hill-climbing" strategy.

Scheirer and Boult [65] come up with another security analysis of the biometric encryption approach. Three new classes of attacks are introduced, namely attacks via record multiplicity, a surreptitious key-inversion attack and new types of substitution attacks. It is concluded that biometric encryption is impacted by surreptitious key-inversion attacks via improved hill-climbing and compromised by attacks via record multiplicity and substitution attacks.

3.2 Private Template Scheme

The objective of a so-called "private template" scheme is to provide user authentication by generating a hash out of user's biometric data which is then matched against a stored hash of this user.

With the use of several biometric inputs and a majority decoder a representative feature vector is generated during enrollment. This vector is then concatenated with an error correction code to overcome the variance in biometric measurements. Afterwards out of the resulting vector and personal information of the user a hash is generated and stored. In the authentication process again several biometric inputs are captured and majority decoded. Subsequently the error correction code is used to detect errors and finally a hash is computed and matched against the stored one. The whole process is described in Figure 3.2.

Davida *et al.* [20, 21] proposed the so-called "private template" scheme in which the biometric template itself, or a hashed value of it, are used as a cryptographic key which implies that if a person's biometric data is compromised, it becomes useless and must not be used for authentication purpose. In their work they discuss an off-line biometric system based on iris biometrics.

In the enrollment process M iris scans are performed and M 2048-bit iris codes are acquired. These M iris codes are then put through a majority decoder. The majority decoder works as follows:

Let $Vec(v_i) = (v_{i,1}, v_{i,2}, ..., v_{i,n})$ be an n-bit code vector. Given odd M vectors $Vec(v_i)$, the majority decoder computes the vector $Vec(V) = (V_1, V_2, ..., V_n)$, where $V_j =$

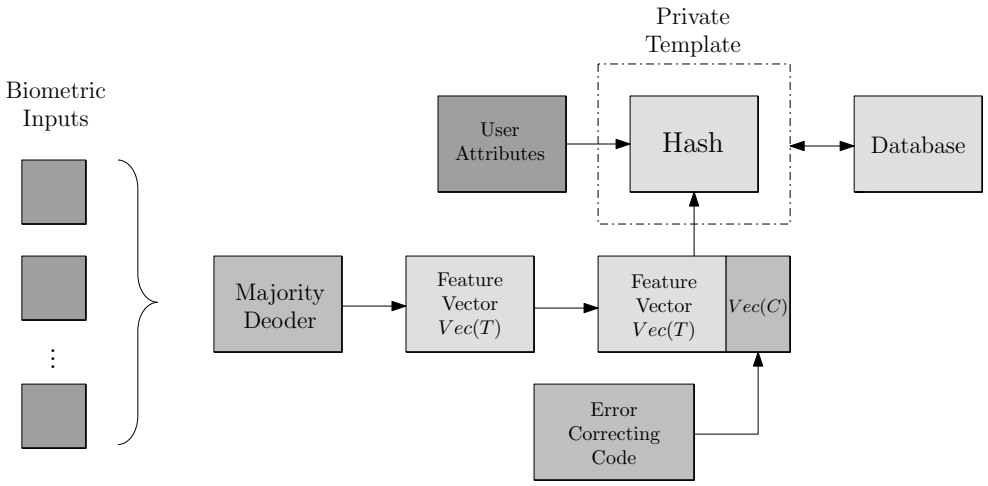

Figure 3.2: The basic operating mode of a Private Temple scheme

($majority(v_{1,j}, v_{2,j}, ...v_{M,j})$. The most common metric for V_j will be the majority of 0's and 1's of bit j from each of the M vectors. This process reduces the Hamming distance between two majority decoded vectors $Vec(V_1)$ and $Vec(V_2)$ and thus given a sufficient number of samples the majority decoder reduces the number of expected bit errors in the 2048-bit iris codes.

After constructing the majority decoded version of a given iris code T, $Vec(T)$, it is concatenated with check digits $Vec(C)$ to form $Vec(T)||Vec(C)$. The check digits are part of a $[n, k, d]$ code (a code of n-bit codewords, where k denotes the number of information digits and d is the minimal distance of the code, which is capable of correcting at least $t = (d - 1/2)$ errors) defined at system setup. At the end of the enrollment process a hash value Hash(Name, Attr, $Vec(T)||Vec(C)$) is generated, where Name is the user's name, Attr are public attributes such as the user's access control list and Hash($\cdot$) is a partial information hiding hash function as proposed by Cannetti [11]. Then the authorization officer signs this hash resulting in Sig(Hash(Name, Attr, $Vec(T)||Vec(C)$)) where Sig(x) is the authorization officer's signature of x (the signature is not further explained).

At the time of authentication again M 2048-bit iris codes are captured and majority decoding is performed resulting in $Vec(T')$. With the use of error correction provided by $Vec(C)$ the corrected template termed $Vec(T'')$ is constructed. In the end Hash(Name, Attr, $Vec(T'')||Vec(C)$) is calculated and Sig(Hash(Name, Attr, $Vec(T'')||Vec(C)$)) is checked. A successful signature implies that the user passed the authentication step.

Wu *et al.* [92] proposed a novel private template scheme based on iris biometrics where a 256-dimensional feature vector is extracted out of a preprocessed iris image using a set of 2-D Gabor filters. A hash function is applied to this vector to generate a cipher key. At the same time an error correction code is generated using a Reed Solomon code. It is suggested to encrypt a secret message with the cipher key. At the time of authentication another feature vector is extracted from a biometric input. This feature vector is error correction decoded and the same hash function like in the encryption phase is used to generate a key

which is used to decoded the encrypted message. Thus the whole scheme aims at reliably generation cryptographic keys out of preprocessed iris images which are subsequently used in a symmetrical cryptographic system.

After the acquisition of an iris image the iris texture is localized and normalized like in generic iris recognition systems. The resulting normalized iris texture is filtered by a set of 2-D Gabor filters with $\Theta = 0^o$, 45^o, 90^o and 135^o, where Θ denotes the orientation of the Gabor filter. Then the filtered iris texture is divided into a total number of 16×4 blocks where for each block the mean gray scale value is calculated. This means, $16 \times 4 \times 4 = 256$ values are calculated which are furthermore normalized to integers in the range $[0, 15]$ to remove most of the noise. The normalized vector, called iris feature vector, is defined by $V = (M_1, M_2, ..., M_{256})$.

If the difference between two iris vectors is below a defined threshold T it should be possible to generate the correct chipher key using the error correction code. Thus, it is suggested to calculate a $RS(N + 2T, N)$ Reed-Solomon code (where $N = 256$) because this code is capable of correcting up to $(N+2T-N)/2 = T$ errors. Then the feature vector is translated into the cipher key using a Hash function, here MD5 is suggested. For the encryption and decryption AES is suggested.

At the time of authentication one iris image is preprocessed and again a normalized iris vector is calculated and error correction decoded. Subsequently the same Hash function, like in the enrollment procedure, is applied to generate the cipher key which is used to decrypt a received message. To eliminate variation in the rotation of the iris image the normalized iris texture is circular shifted in the range of $[-6, 6]$ pixels.

Like in the classic private template scheme the cryptographic key is a hash of the regenerated feature vector. By setting the threshold to $T = 111$ for a total number of over 100 persons a FAR of 0.0% and a FRR of approximately 5.55% was achieved.

3.3 Biometrically Hardened Passwords

In contrast to the above schemes where biometrics are used to create keys or hashes to authenticate legitimate users in a password hardening scheme an existing password is “salted” with biometric data.

In the original concept of the password hardening scheme out of typed password, durations of keystrokes and latencies between keystrokes, are measured. During enrolment out of these measurements the most distinguishable features are extracted and used to generate an instruction table which provides information to reconstruct a hardend password. This instruction table is encrypted with the typed password. At the time of authentication the instruction table is decoded and used to generate the hardened password. Furthermore a history file is stored which is encrypted with the hardened password. This history file includes information about a fixed number of the last successful logins and thus the whole system is capable of adjusting to slight changes of a user's biometrics. In Figure 3.3 the basic operating mode of such a system is shown.

Monrose *et al.* [52] proposed a technique for improving the security of password-based applications by incorporating biometric information into the password. Typed passwords are a weak mechanism to authenticate persons and thus it is suggested to harden such a password.

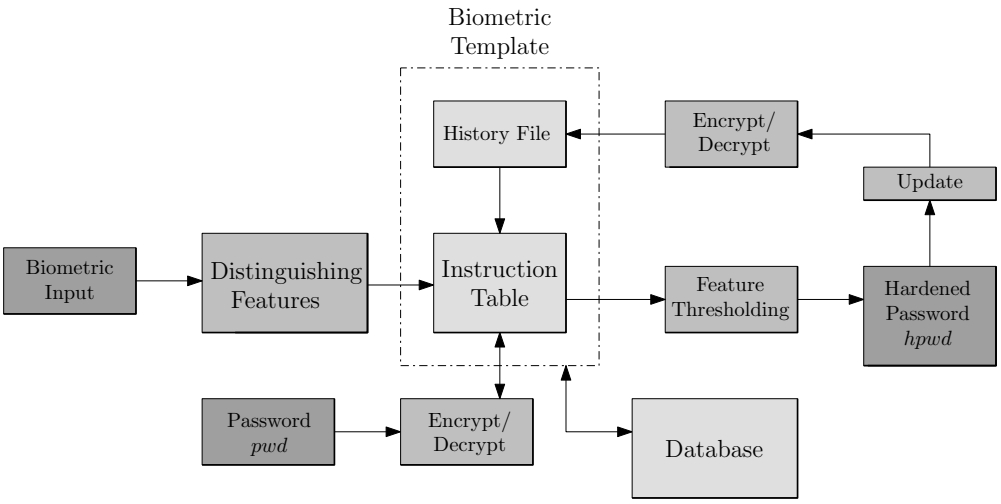

Figure 3.3: The basic operating mode of a Password Hardening scheme

In their approach the legitmate user's typing patterns such as durations of keystrokes and latencies between keystrokes are combined with the user's password, denoted by *pwd* resulting in a hardened password *hpwd*. This hardened password can be tested for login purposes or used as cryptographic key for file encryption.

Assume a computer system with a set A of user accounts where pwd_a is the correct typed password for the user account a and $hpwd_a$ is the corresponding correct hardened password. Furthermore a single measured biometric feature ϕ is defined by a function $\phi : A \times \mathbb{N} \to \mathbb{R}^+$ where $\phi(a, l)$ is the measurement of this feature during the l-th (successful or unsuccessful) login attempt to the user account a.

At the initialization of the account the value of *hpwd* is chosen at random from $\mathbb{Z}_q$ where q is a fixed, sufficiently large prime number (q is suggested to be at least 160-bit long). Then $2m$ shares of *hpwd* denoted by $\{S_t^0, S_t^1\}$, $1 \leq t \leq m$ are created using Shamir's secret sharing scheme [66]. This sharing scheme is a method to share a secret (for example a key) among several participants. Therefore this secret is divided into several parts. Then a polynomial of a chosen degree is defined which can be reconstructed if enough shares are available.

For each $b \in \{0,1\}^{\frac{a}{m}}$ the shares $\{S_t^{b(t)}\}$, $1 \leq t \leq m$, where $b(i)$ is the i-th bit of b, can be used to reconstruct $hpwd_a$. The shares are arranged in an instruction table of dimension $2 \times m$ where each element is encrypted with pwd_a.

During the l-th login pwd' is used to decrypt the elements of the instruction table resulting in the previously stored values if pwd' is correct (the correctness of pwd' is necessary but not sufficient). For each feature ϕ_i the value of $\phi_i(a, l)$ indicates which of the two values should be used to reconstruct *hpwd*. This is done by comparing each $\phi_i(a, l)$ to a threshold $t_i \in \mathbb{R}$, which is a fixed parameter of the system. If $\phi_i(a, l) \leq t_i$ then the value of the left column is used, otherwise the value of the right column.

Central to this scheme is the notion of a *distinguishable feature*. Let μ_{ai} be the mean deviation and σ_{ai} be the standard deviation of the measurement $\phi_i(a, j_1)...\phi_i(a, j_h)$ were $j_1...j_h$ are the last h successful logins to the account a (again h is a fixed parameter of the system). Then ϕ_i is a distinguishable feature if $|\mu_{ai} - t_i| > k\sigma_{ai}$ where $k \in \mathbb{R}^+$ is a parameter of the

system. If ϕ_i is a distinguishing feature for the account a, then either $t_i > \mu_{ai} + k\sigma_{ai}$, that is, the user consistently measures below t_i on this feature, or $t(i) < \mu_{ai} - k\sigma_{ai}$, that is, the user consistently measures above $t(i)$ on this feature. Furthermore, a feature descriptor b which is a partial function $b : \{1, ..., m\} \rightarrow \{0, 1\}$, and the feature descriptor b_a for account a are defined as $b_a(i) = 0$ if $t_i > \mu_{ai} + k\sigma_{ai}$ and 1 if $t_i < \mu_{ai} - k\sigma_{ai}$. That is, $b_a(i) = 1$ for every distinguishing feature ϕ_i on which the user is "slow" and $b_a(i) = 0$ for every distinguishing feature ϕ_i on which the user is "fast". For other features b_a is undefined ($\perp$).

As distinguishing features ϕ_i for this account develop over time, the login program perturbs the value in the second column of row i if $\mu_{ai} < t_i$, and perturbs the value in the first column of row i otherwise. So, the reconstruction to find $hpwd_a$ in the future will succeed only when future measurements of features are consistent with the user's previous distinguishing features. Therefore potential attacks are difficult even if the typed password is chosen poorly which is one main advantage of the proposed scheme. Additionally, if a person's typing patterns change slightly, the system will adapt.

For each account a the according instruction table, which is encrypted with pwd is stored in the system. The instruction table and a constant-size history file form the biometric template. The history file is encrypted with $hpwd_a$ and contains the measurements for all features ϕ_i over the last h successful logins to the account a. Thus the stored information is necessarily based on pwd_a and $hpwd_a$, but will not include either of these values themselves. This is similar to the password storing mechanism Unix^{TM} systems.

Kanak *et al.* [37] provide a detailed insight into biometric key generation from keystroke dynamics and summaries the whole process of password hardening in four steps. First the parameters of a user's keystrokes (durations and latencies) are collected. Then these parameters are processed by the validation algorithm (decryption of the instruction table) and new parameters are generated. In an decision-making step the new values are transferred to a decision function (calculation of the feature descriptor's bits). In this step the user is either accepted or rejected. Finally, the biometric template is updated, which means the history file is decrypted and a new entry is made and the oldest is erased.

Monrose *et al.* [49, 50, 51] apply their principle of hardening passwords to voice biometrics, which was the first approach of generating a cryptographic key out of speech. Typed passwords are unambiguous while speech recognition is still a topic of research. Furthermore, with speech recognition there are plenty more features to focus on than just time. In their approach the representation of a user's utterance is utilized to identify suitable features. This is done by dividing the utterance into a sequence of frames. Each frame is represented by a twelve dimensional vector of cepstral coefficients characterizing a 30ms window of the utterance. After capturing the voice, some endpoint detection, silence removal and cepstrum mean subtraction are applied. Subsequently, a speaker and text-independent acoustic model is used to segment the sequence of frames into m portions.

Each segment i is associated with a closest centroid c_i in the acoustic model. The i-th feature ϕ_i is the position of the segment mean denoted by μ_i relative to a fixed plane translated to a coordinate system with c_i at the origin. That is x is a twelve dimensional vector of coefficients specifying the plane $\alpha \cdot x = 0$, then the i-th feature is the value of $\alpha \cdot (\mu_i - c_i)$. If this value is less than the specified threshold t_i, the i-th feature is assigned 0, otherwise 1. The result of this whole process is a feature descriptor which is used like in the above scheme applied to keystroke dynamics.

Facing the complex problems of using voice biometrics within a cryptography system a false negative rate of approximately 6% and a false positive rate below 20% was achieved.

In another work Monrose *et al.* [4] analyze and formalize two major requirements of biometric key generators. These two requirements are correctness and security. Correctness is described as the ability of being able to reliably extract biometric features and combine those features with a template resulting in the correct output of the key with overwhelming propability. As this requirement seems to be unambiguous it is not discused any further.

Security is composed of *key randomness* and *biometric privacy* while the latter is subdivided into weak and strong biometric privacy. These three requirements are mathematically formalized and the impact of public information on these requirements is discussed.

In conclusion, the biometrically hardened password scheme is a simple scheme for improving the security of password-based authentication systems based on Shamir's secret sharing scheme. It tends to be secure due to the fact that neither the typed passwords nor the hardened passwords are stored as part of the biometric template. The generated hardened password can be used for login purposes as well as a cryptographic key.

3.4 Fuzzy Commitment/Fuzzy Vault Scheme

The objective of a so-called "fuzzy commitment" scheme is to bind biometric features of a user with a key, prepared with an error correction code to overcome the fuzziness of biometric measurements.

At the time of enrollment the extracted features are combined with a codeword of an error correction code. The resulting bitstream is stored in a database together with a hash of the codeword. During the enrollment process, again, biometric features are extracted, combined with the previously stored bitstream and error correction is performed, resulting in another codeword. If the hash of this codeword matched the stored one authentication succeeds. In Figure 3.4 this process is illustrated.

Juels and Wattenberg [36] combined well-known techniques from the ares of error correcting codes and cryptography to achieve a new type of cryptographic primitive that they refer to as "fuzzy commitment" scheme. Fuzzy commitment is the analogon to "fuzzy logic" in artificial intelligence.

In their definition a fuzzy commitment scheme consists of a function F, which is used to commit a codeword $c \in C$ and a witness $x \in \{0,1\}^n$. The set C is a set of error correcting codewords c of length n and x represents a bitstream of length n termed witness (in a biometric cryptosystem x would represent the biometric data). To enhance security only the difference vector of the codeword and the biometric measurement, $\delta \in \{0,1\}^n$ where $x = c + \delta$, and a hash value $h(c)$ are stored as the commitment. The commitment, which is nothing else then these two values is termed $F(c, x)$ (in a biometric cryptosystem $F(c, x)$ would represent the biometric template).

To deal with the fuzziness of x it is proposed that every x', which is sufficiently "close" to x according to an appropriate metric, for example the Hamming distance, should be able to

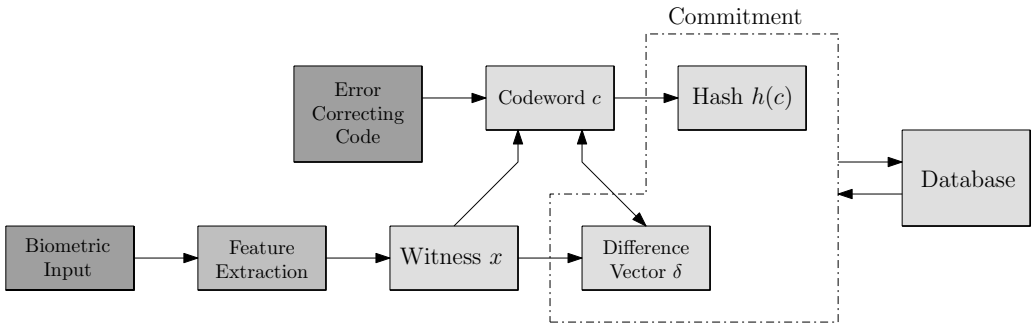

Figure 3.4: The basic operating mode of a Fuzzy Commitment scheme

reconstruct c. If the system is presented with a witness x' that is near x, the difference vector δ is used to translate x' in direction of x. If the correct codeword c is reconstructed with the use of error correction the hash of c', $h(c')$ will match the stored hash value resulting in a successfull authentication.

Obviously the amount of information contained in the codeword c and thus the amount of information about the witness x hidden in $h(c)$ depends on k (the number of codewords in C), where $|C| = 2^k$. The greater the number of codewords the greater is the amount of information about the witness x hidden in $h(x)$. In contrast, the amount of information in δ determines the level of resilience in F. Therefore a tradeoff between resilience and security has to be chosen by varying k and thus the relative distribution of information between δ and $h(c)$.

The enrollment and authentication process of the a whole system would operate as follows: during enrollment a user U presents a witness x to the authentication system S. The system selects a codeword $c \in C$, calculates the fuzzy commitment $F(c, x)$ (the difference vector δ and the hash value of the codeword c, $h(c)$) and stores it in a database. At the time of authentication a user purporting to be U presents a witness x' to S. The system looks up the commitment of user U and checks whether x' yields a successful decommitment, which would lead to a successful authentication.

Altough the proposed fuzzy commitment scheme is an important milestone, Juels and Wattenberg only provide a theoretical approach and do not focus on the practical use of this scheme in biometrics.

Hoa *et al.* [31] applied the fuzzy commitment scheme to iris codes and thus provide a practical insight into the use of the fuzzy commitment with biometric data. Their cryptosystem operates as follows:

During the enrollment the iris of a user is scanned and a 2048-bit iris code prepared with 2D-Gabor wavelets according to Daugman standard algorithms. This 2048-bit iris code is termed θ_{ref} (θ_{ref} corresponds to x is the above scheme). Furthermore, another 2048-bit pseudo iris code is generated denoted by θ_{ps} (θ_{ps} represents a codeword $c \in C$ in the above scheme). This pseudo iris code θ_{ps} is a 140-bit cryptographic key k prepared with Hadamard and Reed-Solomon error correcting codes resulting in a 2048-bit bitstream. The length of 140 Bits should suffice, for example, for the use in the AES algorithm (AES operates on 128-bit). Now the commitment of θ_{ref} and θ_{ps} has to be calculated. This is done by bitwise

XORing θ_{ref} and θ_{ps} resulting in $\theta_{lock} = \theta_{ref} \oplus \theta_{ps}$ (θ_{lock} represents the difference vector δ in the above scheme). Then a hash value of the key k, $H(k)$ is generated using a standard hash function. The hash value $H(k)$ together with θ_{lock} form the biometric template (the commitment $F(c, x)$ in the above scheme). Furthermore, it is suggested to store θ_{lock} on a smartcard or any other physical token.

In the authentication process an iris code is generated in the same way as in the enrollment process. This iris code is denoted by θ_{sam} (θ_{sam} represents the witness x' in the above scheme). Additionally, the person presents the smartcard on which θ_{lock} is stored. Out of θ_{sam} and θ_{lock} another pseudo iris code can be calculated denoted by $\hat{\theta}_{ps}$. This is done by XORing θ_{sam} with θ_{lock}, $\hat{\theta}_{ps} = \theta_{sam} \oplus \theta_{lock}$. With the according Hadamard and Reed-Solomon decoding a 140-bit key $\hat{k}$ is calculated. This key is then hashed and if $h(\hat{k}) = h(x)$ the authentication is successful otherwise the key will be deemed false and rejected. The whole process of generating $\hat{k}$ out of θ_{sam} and θ_{lock} and the matching of the hash values form the decommitment process.

The system was tested with 700 iris images of 70 propands reaching a success rate of 99.5%. Additionally, a FRR of 0.47% and a zero FAR was announced. These are very impressive results which were not achieved until then, especially with iris scan because of the complicated engineering process of generating usable iris codes.

Zheng *et al.* [96] proposed a lattice mapping based fuzzy commitment method for cryptographic key generation. They employed a set of error tolerant lattice functions to map biometric data from the feature space into lattice spaces. Thus only storing lattice functions suffice. Lattice mapping works as follows:

Let $x = (x_1, x_2, ..., x_p)$ be a p-dimensional biometric feature vector and $x_i \in \mathbb{R}, i = 1, ..., p$. A codeword c is defined as a p-dimensional vector with each element being a random binary string. Let $c = (s_1, s_2, ..., s_p)$, $s_i \in_R \{0, 1\}^q$, $i = 1, ..., p$, where $\in_R$ means uniform random selection from a set. Furthermore, the codeword c is treated as the coordinate of x in a lattice space, L, mapped from its real feature space with δ being the half of the lattice grid size. This lattice space L can then be defined by its origin $O = (o_1, o_2, ..., o_p)$ and the grid size δ where $o_i = x_i - \delta - 2\delta s_i$, $i = 1, ..., p$.

With this arrangement, the decoding function $f(\cdot)$ becomes a simple mapping from x to c using the lattice system $L(O, \delta)$, where, $[\cdot]$ is an operator taking the integer part of the input. It can be seen that any biometric data x that lies within the hypersphere centered at x with radius δ will be mapped into the same grid (codeword c). In other words, δ serves as a parameter of distortion tolerance. In addition, it is impossible to calculate original biometric data x from the lattice system $L(O, \delta)$ and x is not required to decommit c from x. Therefore, the system only needs to store $L(O, \delta)$ and the codeword c and x are secure even when $L(O, \delta)$ is open to an attacker. Due to the fact that this method is generic it is claimed that it is applicable to all types of biometric data.

Teoh and Kim [80] try to improve the security provided by a generic fuzzy commitment scheme. A method known as randomized dynamic quantization transformation is proposed to binarize biometric data which should fulfill the requirement of a uniformly random biometric template. The proposed method is applied to fingerprints.

The randomized dynamic quantization transformation consists of three steps: non-invertible random projection, dynamic quantization and condensation. In the first step feature ex-

traction is performed using a multichannel Gabor filter and the resulting feature vector is projected onto a random subspace which is constituted by a random matrix R. This matrix of dimension $m \times n$ can be derived by using a user-specific information such as a password which is stored on a secure device (smartcards are suggested). The random projection is defined as $v = \sqrt{1/m} = Rx$ where v is a set of underdetermined systems of linear equations and if R and v are given it is possible to find the exact values of all elements in x. In the second step quantization is applied and the projected feature element i of user j is mapped into the quantized domain denoted by Q_{ij}. To fulfill security concerns of not revealing the actual quantized values Q_{ij} a method called dynamic thresholding is performed. In the final step a row-wise modulo-two operation which is not addressed further is applied to generate a bit stream.

For evaluations of the proposed technique binary feature vectors of 375 bits were extracted from fingerprint images and Reed-Solomon error correction codes were used. As a result of the randomized dynamic quantization transformation the entropy distribution comprises a mean of 92% (=345 bits) and a standard deviation of 13% (=49 bits) which means the empirical bound is in between $[296, 375]$. These are remarkably good results in terms of entropy of the binary feature vector. Furthermore a FRR of 0.9% and a FAR of 0.0% are reported. However, there is one critical issue: the matrix R, which constitutes the random subspace onto which a users feature vector is projected, is generated using a user-specific token. It is requested that this token, whether it is a password or a user specific seed, is stored on a secure device. These are unrealistic precondition because physical tokens can be stolen, copied, lost of compromised. It is questionable if equal results were achieved using the same seed for all persons (this would be realistic precondition, for instance, if the whole database was compromised). Probably this would result in an decreasing entropy of the binary feature vector as well as in an increasing FAR and FRR.

Maiorana and Ercole [46] proposed a method based on user adaptive error correction codes to secure biometric templates and achieve cancellability. The proposed technique is applied either to signature or iris biometrics.

At the time of enrollment I biometric measurements are acquired for a total number of S users and a intra-class vector μ^s and a inter-class vector μ are estimated where μ^s is the average distance vector of the feature vectors of one user and μ is the average distance vector of all μ^ss. For signature biometrics statistical features such as the number of strokes or the average x- and y velocity are mearsured. Using iris biometrics, normalized iris textures are decomposed into 60 feature vectors, one for each pixel row of the image. Then the extracted feature vectors of a user s is binarized by using the inter-class mean μ where feature vector elements are set to 0 if these are less or equal μ and 1 otherwise. Subsequently a majority feature vector denoted by b^s is calculated out of the I binarized feature vectors. Using a BCH error correction code (a representative of block level error correction codes) a codeword c^s is generated from a randomly selected number N^s. Then the XOR operation between c^s and b^s is performed resulting in so-called helper data $HD1^s$, $HD1^s = c^s \oplus b^s$,which is stored together with μ and a hash value of N^s denoted by $h(N^s)$. Thus cancellability of the template is provided by simply changing N^s.

Furthermore, a second technique is suggested in which the error correction code is adaptively selected based on the intra-variability of the considered biometrics. Therefore, intra-class analysis is performed in the enrollment procedure and the length of the BCH error correction code is chosen with regard to the mean intra-class distance of each user. The selected BCH code is then stored in $HD2^s$. So none of the stored data, μ, $HD1$, $HD2$ and $h(N)$ supply

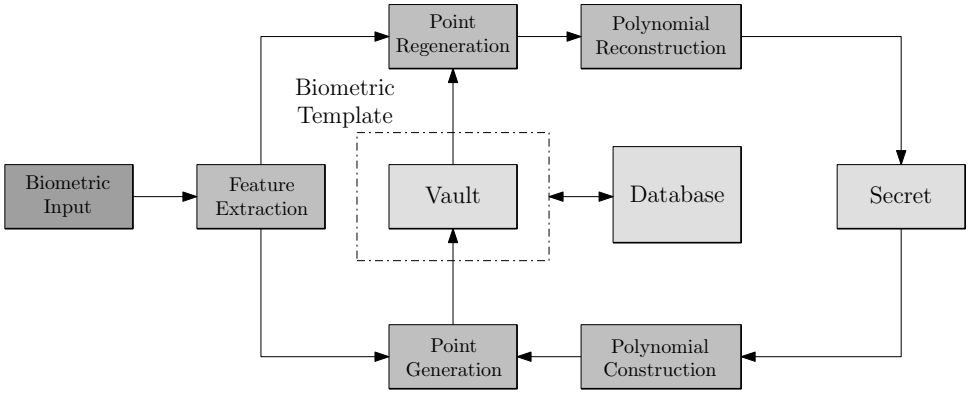

Figure 3.5: The basic operating mode of a Fuzzy Vault scheme

any information about the user's biometrics.

The idea of introducing adaptive error correction code seems to be promising. However, the proposed results are not satisfying providing a FRR of 13.07% and a FAR of 4.0% for signature biometrics and a FRR of 50.0% and a FAR of 7.0% for iris biometrics.

The so-called "fuzzy vault" scheme can be seen as extended version of the "fuzzy commitment" scheme. The great advantage of fuzzy vault scheme is the feature of order invariance.

Within a fuzzy vault scheme a polynom, which serves as secret, is used to commit extracted features by mapping them onto the polynom. If during the authentication process enough points lying on this polynom can be reconstructed the secret, namely the parameters of the polynom, is retrieved. The basic operation mode of such a fuzzy vault scheme is shown in Figure 3.5.

Juels and Sudan [35] improved the work of Juels and Wattenberg by introducing a simple and novel cryptographic construction to which they refer to as "fuzzy vault". This fuzzy vault scheme possesses the useful feature of order invariance. While a simple fuzzy commitment scheme is just able to handle some bit errors up to a specific threshold it is not capable of handling any sort of unordering which is often is the case when measuring biometric data. For example, the ordering of fingerprint minutia could strongly vary form one measure to another if the fingerprint is slightly rotated.

A fuzzy vault may be thought of as a form of error-tolerant encryption operation where keys consist of sets. In a fuzzy commitment scheme a codeword $c \in C$ is used together with a witness x to construct a commitment of a cryptographic key k. In the fuzzy vault scheme a set A and a witness x are used to construct a vault of a cryptographic key k purportedly k is locked using a set A, yielding a vault denoted by V_A. If another set B is presented to the system and B overlaps largely with A, k can be reconstructed purportedly the vault V_A is unlocked.

During the enrollment phase a polynom p is selected which encodes the key k in some way (for example taking the information symbols in k to be the coefficients of the polynom). This conversion is denoted by $p \leftarrow k$. Now the elements of A are mapped onto the polynom

p. The elements of A can be seen as x-coordinates and the mapping of the elements is simply the projection of all points in A onto p. Finally some so-called chaff points are added which are just randomly placed points to represent random noise to enhance security. The set of all points, both those lie on p and the chaff points, called R form the commitment.

To achieve a successful authentication another set B, which is presented to the system, needs to overlap with A sufficiently to generate the correct polynom. First all points of R are projected onto the elements of B which, like the elements of A, represent x-coordinates. For each element $b_i \in B$ a y_i-coordinate has to be found so that $(b_i, y_i) \in R$. If such a pair is found, (x_i, y_i) is denoted (b_i, y_i), if no such pair is found a so-called null-element is assigned to the pair (x_i, y_i). If enough points are lying on p, p can be reconstructed and decoded to receive k.

The security of the whole scheme lies in the number of chaff points and in the infeasibility of the polynomial reconstruction, which means if not enough points lying on p are known, an attacker has hardly any chance to reconstruct p.

Juels and Sudan only provided the theoretical basis for this improved version of the fuzzy commitment scheme. The main achievement of their work is the feature of order invariance. This is provided due to the fact that they use sets as witnesses instead of simple bitstreams. Furthermore, security is enhanced by applying polynoms.

Clancy *et al.* [16] proposed the first practical use of the fuzzy vault scheme by applying it to fingerprint minutiae in a "fingerprint vault" system. In their system multiple fingerprints of a user are acquired during enrollment and the minutiae positions are extracted. A correspondence between the feature points is established by applying a bounded-nearest-neighbor algorithm. This can be figured as overlaying all captured fingerprints on top of each other and clustering spatial close points. The distance denoted by d is predefined by the system. Those points for which a correspondence is found are denoted by G. Thus G represents the embedded elements of A onto the polynom p. Together with some randomly added chaff points denoted by N, which are at least d away from these points form the commitment, denoted by R. Then the ordinates are determined by the polynomial embedding of a key k to be shared. The calculation of ordinates is the mapping of the minutiae points and the chaffpoint lying on p onto the y-axis.

During authentication the minutiae points of several presented fingerprints are extracted. By applying a bounded-nearest-neighbor algorithm new clusters are established of which the ordinates are calculated. Finally Reed-Solomon codes are applied to reconstruct the polynom p out of which the key is recreated.

In this work the most apparent way of how to use the fuzzy fault scheme with biometrics is presented. Especially for fingerprint biometrics the fuzzy vault appears to fit well. The great vulnerability of their work was the assumption that all the fingerprints are prealigned, which is rarely the case in practice.

Uludag *et al.* [86, 87] analyzed and experimented with the above fingerprint vault system. To overcome the vulnerability of assumed prealignment they propose a "fuzzy fingerprint vault", which uses minutiae lines to lock a key. In their system this means a line-based minutiae representation is used in which the whole fingerprint consists of lines where the line K_{ij} between minutiae i and j is defined as: $K_{ij} := (x_i, y_i, \phi_i, x_j, y_j, \phi_j, d_{ij}, \Phi_{ij}, \omega_i, \omega_j)$ where the first six parameters are the coordinates and angles of the minutiae point d_{ij}

represent the distance between the points, Φ_{ij} the line direction and ω_i and ω_j the angles between the line direction and the minutiae points. Such a representation eases the issue of aligning fingerprint minutia.

Nandakumar *et al.* [55] extended the idea of the fuzzy fingerprint vault scheme and presented an fully automatic implementation of the fuzzy vault scheme based on fingerprint minutiae. In their work they face the main implementation challenge of a fingerprint authentication system based fuzzy vault scheme, namely the alignment of a query fingerprint measurement to the original template. In their system they store high curvature points derived from the orientation field of the template fingerprint as helper data to assist the process of alignment. The helper data accurately aligns the template and query minutiae but does not reveal any information about the minutia points that form the template.

Instead of only using the location of minutia points like in the original fuzzy fingerprint vault scheme they use minutia location and orientation attributes which increases the number of chaff points that can be added. Points are used in a 3-tuple (u, v, θ) where u and v indicate the row and column indices in the image and θ the orientation of the minutiae with respect to the horizontal axis. More points of which the location is close to a true minutia but with a different direction can be added and thus the security is improved.

Points of high curvature are extracted from the fingerprint orientation field. Since high curvature points are global features in the fingerprint pattern high curvature points do not reveal any information about the minutia attributes which are local characteristics in the fingerprint.

The generation of the helper data and using it for alignment is the main achievement of this work and can be summarized in the following steps: First the orientation field of the fingerprint captured during enrollment is estimated. Then flow curves are extracted from the orientation field. The points with the maximal curvature are determined and theses points are clustered.

In the authentication phase a so-called interactive closest point algorithm is used together with the calculated high curvature points to align the presented fingerprint to the stored template.

Tulyakov *et al.* [83, 84] presented a new approach for hashing fingerprints. To overcome the problem of ordering fingerprint minutiae symmetric complex hash functions are used. This means, instead of ordering the fingerprint minutiae, they simply apply order invariant hash functions.

The minutiae points are presented as complex numbers $\{c_i\}$. The position of such a point can vary from one measurement to another. This transformation is described by the complex function $f(z) = rz + t$. Thus a minutiae point of a new measurement can be written as $c_i' = f(c_i) = rc_i + t + \epsilon$, where ϵ represents errors occurred during the capturing of the fingerprint. For an acquired set of minutiae points $C = \{c_1, c_2, ..., c_n\}$, m different hash functions are calculated where $h_1(C) = c_1 + c_2 + ... + c_n$, $h_2(C) = c_1^2 + c_2^2 + ... + c_n^2$, ... , $h_m(C) = c_1^m + c_2^m + ... + c_n^m$, where n is the number of neighboring minutiae points. The matching between the generated hash values of the fingerprint minutiae during enrollment and any other set of hash values consists of finding r and t that minimizes the error. In their work they mention that a hash value could become greater than the biometric data using a very large number for m. Nevertheless experiments are presented with $m \leq 2$.

In this approach the authentication is performed in another space while only hash values are transferred between the user and the authentication point and only hashed values are stored, of course. This approach to solve the problem of aligning order invariant data is a very simple one but therefore the scheme is restricted to use just these symmetric complex hash functions.

Yang and Verbauwhede [93] introduced another method for improving the alignment of minutiae points. To use minutiae coordinates as rotation invariant features all points are mapped from cartesian to polar coordinates. If the origin of the polar coordinate system is selected correctly, each feature will be independent of rotation of the input images. The method of finding this origin is not further described and the case this origin cannot be found is not discussed. A minutiae feature is defined as a tuple $M = (d, \theta, \varphi)$ where d is the distance to the origin, θ the position angle and φ the direction difference to the origin.

This is a method of solving the problem of aligning fingerprint minutiae points in feature space (in contrast to the above mentioned method where order invariance is achieved in another space). The rest of their system works like the fuzzy fingerprint vault scheme.

Nagar and Chaudhury [53] presented a modified fuzzy vault scheme, which is used in an asymmetric cryptographic system focusing on fingerprints. Users can send and receive secure information using just the fingerprints. The process of creating the fuzzy vault can be described in the following steps:

First a message is encoded using Reed-Solomon codes resulting in an encoded message C of length n. Then each element of C is placed on a grid of size $n \times 3$ such that i-th row of the grid contains the i-th element of C placed randomly in one of the three columns. This grid is denoted by $gridC$. The biometric template (in this case fingerprint template) of length n is placed on a similar grid denoted by $gridB$ so that its position and order coincides with that of C in $gridC$. The rest of the elements of $gridC$ are filled with random numbers in the appropriate range while the rest of the elements of $gridB$ are filled in such a way that each row becomes an arithmetic progression of distance equal to the tolerance value, FV_{tol}. This means that the two random numbers, which are added to each row of $gridB$, have the same difference (FV_{tol}) to the correct value.

To unlock the vault the knowledge of either the correct positions of the legitimate elements in $gridC$ or $gridB$ suffices. The sequence of numbers of the legitimate points of $gridC$ is nothing but the Reed-Solomon code for the encrypted message. This code can be easily decoded using any of the standard algorithms to get back the desired message.

If the receiver has the actual biometric feature, the legitimate-point sieving algorithm is just to select one point out of three from each row of $gridB$ which is nearest to the corresponding biometric value.

For a FV_{tol} between 4 and 8 a zero error rate which means that FRR and FAR are both zero is reported. This would be an optimal system, therefore these results are doubtable. Nevertheless, in this work a simple and fast method of creating a fuzzy vault for fingerprints is presented.

Reddy *et al.* [61] enhance the security of a fuzzy vault scheme based on iris biometrics by embedding an additional layer of security, namely a password. With this password the

generated vault as well as the secret key is hardend. The hardening scheme consists of the following steps: first a random transformation function derived from the user's password is applied to the biometric template. Then the transformed template is secured using the fuzzy vault framework. Finally the vault is encrypted using a key derived from the password.

In the preprocessing step an iris image is acquired and the iris texture is localized. Afterwards morphological operations are applied on the highlighted iris texture to extract a pseudo structures from which minutiae (nodes and end points of textures) are calculated. The resulting set of minutiae coordinates is then divided in four quadrants. The minutiae of each quadrant are then translated and rotated using a 8-letter password which is divided into 4×16 bit password blocks. Each password block is used to transform the points of one quadrant where the first component T_r of 7 bits is used for radial translation and the second component T_Θ of 9 bits is used for angular translation. The translation values are added to original values modulo some appropriate range so that the translated minutiae points lie within the respective quadrant. Furthermore the vault of the resulting points is encrypted using the password.

At the time of authentication the vault is decrypted and the password-based transformation is applied again. Then an acquired iris image is used to extract another set of minutiae which are then used to "unlock" the vault.

In experiments a fuzzy vault scheme which exhibits a FRR of 8% and a FAR of 0.03% is hardened, where the ratio of chaff points and original points is taken as $10 : 1$ and a total number of 100 templates are used. As result of the hardening scheme the FRR increases to 9.8% due to misclassification of a few minutiae at the boundaries of the quadrants. At the same time the FAR decreases to 0.0%. It is claimed that this is due to the fact that minutiae are distributed more randomly. However, if passwords are compromised the system's security decreases to that of an ordinary fuzzy vault scheme which indicates that the FAR of 0.0% was calculated under unrealistic preconditions.

Scheirer and Boult [65] proposed a theoretical security analysis of the fuzzy vault scheme focusing on fingerprint biometrics. Three classes of attacks are introduced which could be exploited to crack fuzzy vault schemes, namely attacks via record multiplicity, a surreptitious key-inversion attack and new types of substitution attacks. It is concluded that one fuzzy vault tends to fulfill security requirements because chaff-points are added and it is not feasible guessing the subset of legitimate minutiae points. However, given two such fuzzy vault instances generated from the same print, but with different keys and different random chaff points, minutiae points will be recoverable by matching these two vaults. Probably some random chaff points will match the second template but still a better part of the minutiae points will be detected.

Finally requirements for fuzzy vault schemes are constituted such like that no combination of data from multiple enrollments by the same user should be able to be combined to recover the biometric template.

3.5 Secure Sketch/Fuzzy Extractor Scheme

The aim of a "secure sketch" and a "fuzzy extractor" is to extract a random bitstream out of a given biometric data in an error tolerant way and provide some information about this bitstream so that a user with a biometric data "close" to this is able to reconstruct this

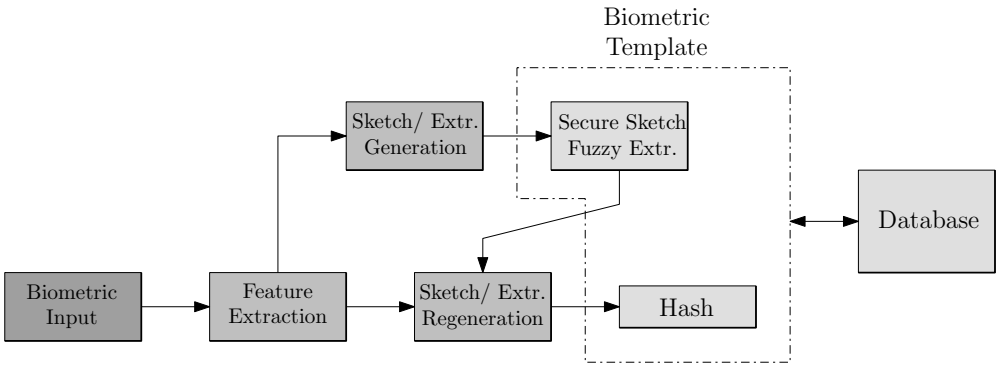

Figure 3.6: The basic operating mode of a Secure Sketch/ Fuzzy Extractor

bitstream. Figure 3.6 shows a basic scheme using a secure sketch or a fuzzy extractor.

Dodis *et al.* [23] introduced two new primitives, namely a fuzzy extractor and a secure sketch for which they provided formal definitions. Both of the primitives aim at turning biometric data into keys for cryptographic applications and reliably securely authenticate biometric data. The first primitive, the secure sketch, addresses the problem of error tolerance. It is a probabilistic function outputting a public value v about its biometric input w. While revealing little about w, v allows the reconstruction of w from any other input w' that is sufficiently close to w.

The second primitive, the fuzzy extractor addresses error tolerance and nonuniformity. It reliably extracts a uniformly random string R from its biometric input w in an error-tolerant way. To assist the recovery of this random string R from an input w', sufficiently close to w, a public string P is outputted (like v in the secure sketch). However R remains uniformly random.

In the fuzzy extractor scheme the authentication server stores $\langle P, f(R)\rangle$ termed fuzzy key storage, where f is a one way function which is not further described. Given a biometric input w' the public information P is used to reconstruct R and $f(R)$ is checked. In the secure sketch scheme v and $f(w)$ are stored and w is reconstructed with the help of the public information v.

In their work they defined formal constructions for three metrices, the Hamming distance, set difference and edit difference. In terms of Hamming distance a secure sketch can be seen as a fuzzy commitment scheme. The metric of set difference is applied whenever the biometric input only provides a subset of biometric features. For example, the size of the symmetric difference of two input sets w and w' could be calculated as set difference while the edit difference, for example, is the number of insertions and deletions needed to convert one string into another.

They do not further address details of how to realize the presented primitives, they only provide formal definitions. Nevertheless, this is important work, especially due to the introduction of the set difference metric, which offers new possibilities in the field of hybrid biometrics which will be described later.

Boyen [8] discovered several security vulnerabilities in the concept of fuzzy extractors

and secure sketches. With multiple invocations of fuzzy extractors and secure sketches the exposion of the complete secret is obtained, which implies that the above introduced primitives can only be applied securely for single uses. To overcome this vulnerability new formal definitions of the fuzzy extractor and the secure sketch are proposed.

Burnett *et al.* [9] created an identity-based signature scheme, which uses biometric data to construct a public and a private key. This is done by combining fuzzy extractors with the biometric data resulting in a random key string. On this random key string an elliptic curve point embedding algorithm is applied to generate a public and a private key. The basic units for this elliptic curve arithmetic are points (x, y) on the elliptic curve, E, over a finite filed, F_p, denoted by $E(F_p)$ of the form $y^2 = x^3 + ax + b$ with x, y, a, $b \in F_p$.

First a fuzzy extractor with respect to the Hamming distance metric is defined, which outputs a random string $U \in \{0,1\}^k$, given a user's signature. Then the random string U is embedded onto a point P on the elliptic curve E. This is done by hashing U with a hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ where $\mathbb{G}_1^*$ is a subset of points lying on E. The hashing of U is divided into two steps. First U is hashed with a standard hash function $H : A \rightarrow \{0,1\}^*$ (SHA-1 is suggested as standard hash function). Then a deterministic encoding function $g : A \rightarrow \mathbb{G}_1^*$ is applied resulting in $g(H(U)) = H_1(U)$. Now that U is embedded onto a point of the elliptic curve, out of this point P_b, $P_b = g(H(U))$, a private and a public key are generated. This is done by calculating $P_s = xP_b$ where x is a randomly generated key in F_p^* (the generation of x is not further described). The private key is presented by x while the public key is represented by P_s.

Voderhobli *et al.* [91] introduced the idea of using hybrid biometrics together with a secure sketch to generate cryptographic keys. It is proposed to define a set B of all biometric characteristics which the system is able to acquire so that $B = \{R_{iris}, R_{fing}, R_{voice}, R_{face}, R_{pwd}\}$. Additionally, these characteristics are weighted. During authentication an adequate subset of B, for example $S = \{R_{iris}, R_{fing}, R_{pwd}\}$ should suffice to authenticate a person, which entails the challenge of correlating S and B.

At the time of enrollment all biometric measurements are performed, combined using a so-called consolidate function, hashed and stored in a major template key denoted by t_{key}. The secure sketch $SS(w)$ with respect to the set difference metric is defined where w represents the measurement of one biometric characteristic defined in B. Furthermore, a recover function Rec is defined which is capable of reconstructing w out of a biometric input w' if $w - w' \leq vl$ where vl is the variation allowed. The variation allowed vl is a set of scalar values for each biometric characteristic which is measured while $w - w'$ is the distance measured for each characteristic based on the Hamming distance. Thus the recover function Rec and the major template key t_{key} form the biometric template.

During the authentication process a set $S = \{w'_1, w'_2, ..., w'_n\}$ is measured where $n \leq |B|$. Then Rec is allied to this subset . If S is only a subset of B, padding bits are appended. The the hash function which is not further described, is calculated resulting in an comparator key c_{key} which is matched against the template key t_{key}.

Altough no details of implementation are described, the idea of applying a secure sketch to hybrid biometrics if only a subset of the required characteristics is available during authentication appears to fit. Furthermore, the idea of performing the template matching for each subsystem as part of the overall process is an innovative approach for combining

weighted biometric measurements.

Li *et al.* [43, 73] were the first to bring up the problem of entropy loss generating a secure sketch by the use of error correcting codes. The main difficulty is that many biometric templates are represented as points in continuous domains with unknown distributions, whereas known results either work only in discrete domains, or lack rigorous analysis on the entropy loss. In their work, instead of trying to solve these problems directly, it is proposed to examine the relative entropy loss of any given scheme, which bounds the number of additional bits that could be extracted if optimal parameters were used. They analyze general secure sketch schemes focusing on face biometrics and estimate that the relative entropy loss of this schemes is at most $n \log 3$ where n is the number of points, and the bound is tight.

In the second work they show how to use secure sketches to protect biometric templates. They examine an authentication scheme using face biometrics and show how to apply a known secure sketch scheme on top of it to protect the biometric templates. The min-entropy of the feature vectors is estimated to be about 108 bits which means the probability of getting any particular string from the secure sketch is at most 2^{108}. The experiments show that, the average size of the sketch is about 73 bits, which gives a guarantee of 35 bits in the left-over entropy. Nevertheless, this is just a lower bound for the left-over entropy, and the exact security would require further investigation.

Draper *et al.* [24] introduced a new approach of combining measured biometric fingerprint data with any sort of error correcting code by applying Slepian-Wolf codes. Thereby they increase entropy and thus security. In the enrollment phase feature extraction and quantization function $f_{feat}(\cdot)$ (this function is not further described) maps a raw biometric input into an enrollment biometric x. This enrollment biometric x is shared between the legitimate user and the access control system. A security function f_{sec} maps x into a secure biometric s, $s = f_{sec(x)}$. The access control point stores s and a hash value of x, $f_{hash}(x)$.

In the authentication phase out of an acquired raw biometric input y another x', where $x' = f_{feat}(y)$, is generated and $f_{hash}(x')$ is matched against $f_{hash}(x)$ (stored during enrollment). If $f_{hash}(x') = f_{hash}(x)$ the authentication succeeds, otherwise the user is rejected.

The above mentioned security function, applied during enrollment is performed with the use of a Slepian-Wolf code R_{SW} [67]. This Slepian-Wolf R_{SW} is a random "binding" function, which according to the Slepian-Wolf theorem, is capable of encoding two correlated sources at a rate equal to the joint entropy of theses sources. Thus the use of Slepian-Wolf codes is another approach to overcome the problem of entropy loss when using secure sketches.

Martinian *et al.* [25] described a scheme how to store biometrics in a fuzzy extractor scheme via syndromes and Slepian-Wolf codes and present a prototype biometric cryptosystem for iris recognition. Like the above approaches this work addresses the issue of keeping the entropy of the modified biometric data as high as possible.

Tong *et al.* [82] proposed a fuzzy extractor scheme based on fingerprint biometrics to associate and further retrieve a committed cryptographic key to be used in security applications. The proposed method uses a stable and order invariant representation of biometric data called Fingercode [34]. This technique is texture based which means the variation of the ordering of minutiae points does not affect the output of the algorithm.

Using this special fingerprint data representation a cryptographic key is rebuilt using a public available data which is referred to as FingerKey. The secret key cannot be recovered either from the FingerCode nor form the FingerKey.

As the result of the applied FingerCode method a 640-component vector of integers that range from 0 to 7 are extracted. These components are ordered and stable in size. As error correction code a Reed-Solomon code is applied. The secret key is represented as a word of $d+1$ letters which corresponds to the $d+1$ integer coefficients of a polynomial $p \in \mathbb{Z}[X]$ of degree d. In the enrollment procedure the public available FingerKey is extracted from the the pair (F, p), where F is the FingerCode. Furthermore, n randomly chosen points of p where $n > d$ are bound with n stable subparts of the FingerCode. To retrieve the secret polynom p Reed-Solomon decoding is applied. If at least $d+1$ points are decommited p can be reconstructed.

Using the standard FingerCode algorithm a FRR of 78% and a FAR of 0.1% are reported. These results are not very satisfying and far from being suitable for practical use.

In another work Li *et al.* [42] study how to build secure sketches for asymmetric representations based on fingerprint biometrics, in the sense that the biometric template created during the enrollment procedure contains more data than that which is created during verification.

In the proposed scheme pre-aligned minutiae points serve as biometric features. These minutiae points are transformed using a transformation T. Applying this transformation n components are obtained. This is done by drawing n randomly chosen straight lines in 2-D space and for each line the difference between the number of minutiae points on the "left" and on the "right" side of the line is calculated. The resulting n components are associated with weights. In the enrollment procedure m biometric samples are acquired and for the i-th component the standard deviation s_i and the mean m_i are estimated. The weight w_i which is associated with the i-th component is defined as $w_i = m_i/s_i$. According to these weights groups are calculated so that the n components are divided into q groups. For the j-th group, denoted by G_j, g components, denoted by $d_{j,1}, d_{j,2}, ..., d_{j,g}$, are combined by computing $x_j = \sum_{k=1}^{g} c_k d_k$, where $c_j \in \{-1, 1\}$. The sign of the c_js are chosen such that the value of x_j is maximized. Finally a bit b_i is assigned to each group such that $b_i = 1$ if $x_i > 0$ and $b_i = 0$ otherwise. As secure sketch the whole grouping information $G = (G_1, G_2, ..., G_q)$ is stored where G_j consists of the indices of the components of the j-th group and information about how these are combined.

During the authentication procedure a single biometric sample is acquired and the grouping information is used to obtain a bitstream Y of length q and furthermore X is reconstructed using a error correction code C which is not further described. Depending on the desired FAR and FRR a threshold t on the number of bit errors to tolerate is defined.

As experimental results a FAR of 1% and a FRR of 20% is reported. Furthermore the security of the system is analyzed and it is concluded that asymmetric representations can be employed when the main concern is the key strength but it is not suitable for high security applications.

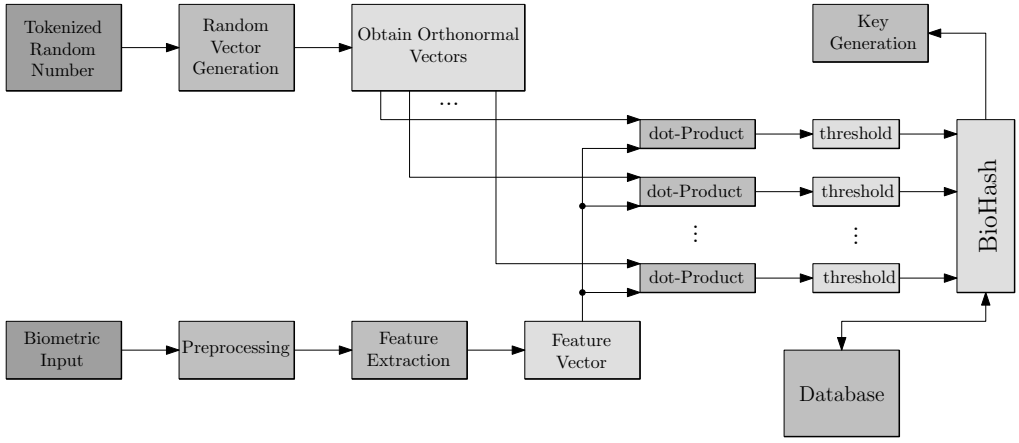

Figure 3.7: The basic operating mode of the BioHashing algorithm

3.6 BioHashing

In the BioHahsing approach a user's biometrics together with a tokenized random number are used to generate a hash which is used to authenticate the user. Out of this user-specific random number several random vectors are generated. These random vectors are mixed with a feature vector, resulting out of a preprocessing step where denoising and feature extraction is performed, via an inner-product. The generated values of this whole procedure are then thresholded resulting so-called BioHash which is stored in a database. The whole process of BioHashing is shown in Figure 3.7.

Ngo *et al.* [28, 29, 78, 79] introduced a new technique applied to face biometrics which they refer to as "BioHashing". The whole process of BioHashing can be summarized into two stages while the first stage is subdivided in two substages:

In the first part of the first stage the raw image $I \in \Re^N$, where N specifies the pixelisation domain, is transformed to an image representation in log-polar frequency domain $\Gamma \in \Re^M$, $M < N$, where M specifies the log-polar spatial frequency dimension. This is done by first applying a wavelet transform in order to reduce noise and produce a representation in a lower frequency domain, which makes the output immune to changing facial expressions and small occlusion. Then a Fourier–Mellin transform is applied to produce translation, rotation and scale invariance.

In the second part of the first stage the generated face feature $\Gamma \in \Re^M$ is reduced to a set of single bits $b \in \{0,1\}^{l_b}$ of length l_b. This is done via a uniform distributed secret random numbers $r_i \in \{-1,1\}$ which are uniquely associated with a token. These tokenized random numbers, which are created out of a user's seed, take on a central role in the BioHashing algorithm. The process of creating b out of tokenized random numbers is subdivided into four stages:

First the user's seed, which is distinct for each user, is used for generating a set of random vectors $\{r_i \in \Re^M | i = 1, ..., l_b\}$. Then the Gram-Schmidt process is applied to the set of random vectors $\{r_i \in \Re^M | i = 1, ..., l_b\}$ resulting in a set of orthonormal vectors $\{r_{\perp_i} \in \Re^M | i = 1, ..., l_b\}$. The third step is to calculate the dot product of the feature

vector and all orthonormal vectors $\{\langle\Gamma|r_{\perp_i}\rangle \in \Re^M | i = 1, ..., l_b\}$. Finally a l_b-bit FaceHash $b \in \{0,1\}^{l_b}$ where b_i, the i-th bit of b is 0 if $\langle\Gamma|r_{\perp_i}\rangle \leq \tau$ and 1 otherwise where τ is a predefined threshold, which can be set to zero. Thus the result of the whole first stage is the so-called FaceHash.

The second stage of the BioHashing algorithm is the key computation. A cryptographic key k_c, which is an element of the field $\mathbb{Z}_q$ for a large prime q is generated out of the FaceHash b. This is done by applying Shamir's secret sharing scheme [66]. The calculated shares are stored in a device.

In conclusion the algorithm of BioHashing can be decomposed into two components: firstly an invariant and discriminative integral transform to extract features from the biometric data (here face biometrics), with a moderate degree of offset tolerance. This would involve the use of integrated wavelet and Fourier-Mellin transform. And secondly a discretisation of the data via an inner-product of tokenized random number and user's biometric data, for integral transform functions with enhanced offset tolerance. Additionally, the cryptographic key is generated with the use of Shamir's secret sharing scheme [66].

A detailed insight is given into how they perform dimensionality reduction on the captured face images to improve the capability to represent features in the feature extraction process. To improve the robustness with respect to the variation in the biometric data error correcting codes are applied. They provide cancellable templates by applying noninvertible functions during the generation of the FaceHash. As results a FRR=0.93% and a zero FAR are reported which are much better than for conventional face recognition systems.

In another work Ngo *et al.* [68, 77] applied the above algorithm to fingerprint biometrics. The idea remains the same, thus all the above mentioned steps of the algorithm are performed. Fingerprint images are captured out of which a FingerHash, b of length l_b (the analogon to the above FaceHash) is generated. They extend this idea and present a whole key release scheme.

This key release is based on the fuzzy commitment scheme. First a cryptographic key is prepared with an Reed-Solomon error correction code resulting in the bitstream k of length l_b. To bind a specific bitstream k with the biometric data b both bitstreams are XORed. The result of this operation is a so-called BioCode β, $\beta = b \oplus k$. At each authentication a new FingerHash termed t is calculated. This FingerHash is XORed with the BioCode resulting k', $\beta \oplus t = k'$. In the end a hash value of k', $H(k')$, where $H(\cdot)$ is a discrete hash function, is compared to a hash value calculated during enrollment and if $H(k') = H(k)$, the user is accepted.

Ngo *et al.* [17] presented another approach for applying the BioHashing algorithm to palmprints. In this approach they combined palmprints with tokenized random number resulting in a PalmHash which they suggested to be stored on a smartcard. Furthermore, like in the above schemes it is required to use different PalmHashes for different type of services.

Kong *et al.* [39] gave an overview of the BioHashing algorithm and presented an implementation of FaceHashing. Furthermore, an explanation for the zero EER reported in the first

works of BioHashing is given. This result was achieved due to the tokenized random numbers, which are assumed to be unique across users. The authors assume that these tokenized random numbers will not be stolen, lost, shared or duplicated. This means that the introduction of biometrics becomes meaningless since the system could rely on these tokenized random numbers without risk. Therefore EER was achieved under a hidden and unpractical assumption.

As a variant of BioHashing, FaceHashing was implemented. In conclusion, following an example of FaceHashing, this work is exposing the true performance of BioHashing.

Lumini and Nanni [44] established several vulnerabilities of the original concept of BioHashing and proposed several improvements to make BioHashing more secure. The main vulnerability they detected is that the performance of BioHashing depends on m, the number of linear independent random vectors. Furthermore, m is bounded by n, the number of features that are extracted out of a user's biometric data.

The second vulnerability they detected is that the performance of BioHashing also depends on τ, the threshold against which $\langle \Gamma | r_{\perp_i} \rangle$, the dot product of the feature vector and all orthonormal vectors, is matched against. This was not supposed in the original concept.

To overcome these vulnerabilities they suggested three improvements of the original BioHashing concept: The first is to normalize the biometric vectors by their module before applying the BioHashing procedure, so that the scalar product is within the range $[-1, 1]$. In the second improvement instead of using a fixed value for τ, several values for τ are used and combined. The last improvement suggests more projection spaces to create more BioHashes for each user. This is suggested because m, the dimension of the projection space cannot be increased.

Ngo *et al.* [40, 94] proposed a novel method to generate cancellable keys out of dynamic hand signatures based on the random mixing step of BioPhasor and user-specific 2^N discretisation for a better recognition rate.

The feature extraction process of their scheme is function-based. It consists of a discrete wavelet transform for location-sensitive compression and a discrete Fourier transform for frequency analysis to obtain a compact representation of the biometric feature.

The extracted feature is then randomly mixed with a token T using a BioPhasor mixing method. The BioPhasor mixing is an inner product based mixing which encrypts the biometric feature in a one-way manner. Like in the original BioHashing algorithm a user specific token is used to calculate several random vectors which are orthonormalized and combined with the feature vector by calculating the inner products of all these random vectors and the feature vector. Additionally an one-to-one arctan transformation is computed which maps the biometric vector onto a bounded range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. This transformation is proven to be non-invertible and thus provides higher security than the original BioHashing algorithm.

Then the real to binary conversion is done with the help of 2^N discretisation (and user specific statistics). In the end Gray coding is applied to decrease the Hamming distance of "close" signatures. The result of the whole process is a SignatureHash just like a FaceHash or FingerHash in the above approaches.

By mixing the extracted features with the BioPhasor mixing method they provide cancellable biometrics. If a SignatureHash is stolen only the basis of the BioPhasor mixing process has to be changed.

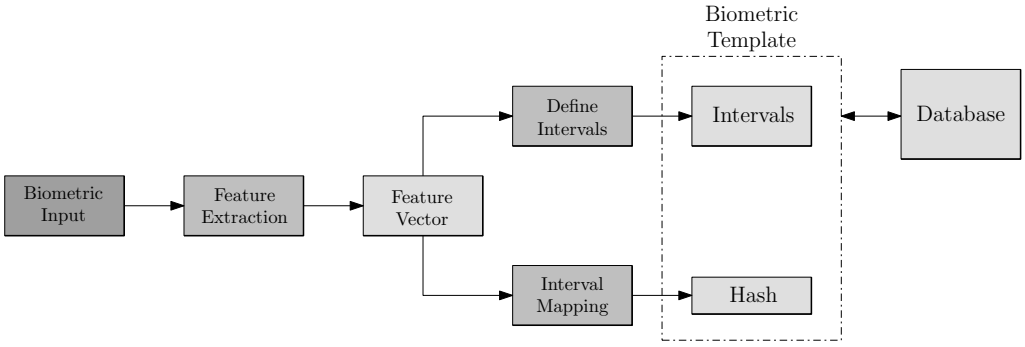

Figure 3.8: The basic operating mode of a scheme using intervals

3.7 Schemes using Intervals

Another group of schemes extracts biometrics features out of several enrollment samples and defines intervals for each of these features. At the time of authentication again a user's biometrics are measured and fit into the previously defined intervals out of which a hash is generated. Finally the calculated hash is compared against a stored hash in the matching process. Figure 3.8 illustrates this process.

Feng and Wah [26] use on-line handwritten signatures to authenticate a user for whom after successful authentication a public and private key are created. Their whole system can be summarized in three stages, namely shape matching, feature coding and key generation.

In the first stage both x and y-coordinates of the signature are warped through dynamic time warping. Then the shapes of x, y waveformes of a test sample is aligned with the reference sample acquired during a previous enrollment. Correlation coefficients can be obtained between position-warped x, y data and the reference ones. Low correlation coefficients will result in rejection of the sample. In this stage an average of $\sim$ 47% of false samples are rejected.

In the feature-coding stage three boundaries are defined for all features (43 different features are used in their approach), the whole boundary, the database boundary and the user boundary, while the last is specific for each user. The whole bound defines a boundary for all possible values for a distinct feature (for this boundaries infinity value can be assigned). The database boundary consists of all values measured by the system. While these two boundaries are equal for all users the user boundary is different for all. It is defined as $(\overline{T} - b \cdot std_T, \overline{T} + b \cdot std_T)$ where $\overline{T}$ is the mean of ten feature values, std_T is the standard deviation of these values and b is a parameter to be adjusted.

The whole boundary is divided into several segments defined with numbers (starting with 0). The boundaries for all these segments are stored as a template. If a biometric sample passed the shape-matching stage, all the features are fitted into these segments and a feature code is returned (for example the segment number).

In the third stage a public and a private key are generated out of the feature codes using the standard DSA algorithm.

As the results of this approach a FRR of 28% and a FAR of 1.2% for the generation of 40-bit long keys are reported. The small FAR seems to result from of the breed selection in the shape matching step. Unfortunately they only list a small set of the features they use in their system.

Vielhauer *et al.* [88, 89, 90] proposed a method for generating hash values out of online signatures. Out of three input signals $x(t)$, $y(t)$ and $p_{0|1}(t)$, where $p_{0|1}(t)$ is a binary pen-up/pen-down signal, 24 feature parameters are calculated. These parameters include for example duration of the complete writing process and the average writing velocity in x and y direction.

During enrollment an interval matrix is generated for each user out of less than five samples. This interval matrix is of dimension $N_{Parameters} \times 2$ where each line consists of $\triangle I_i$ and Ω_i. The first value of each column, $\triangle I_i$ is computed, based on two intervals. The first interval, the initial interval, is defined as $[I_{InitLow}..I_{InitHigh}] = [MIN(n_{ij})..MAX(n_{ij})]$ where n_{ij} is the value of the i-th feature of the j-th sample, $j \leq 5$. The second interval, the extended interval, is defined as $[I_{Low}, ..., I_{High}] = [I_{InitLow} \cdot (1 - t_i)..I_{InitHigh} \cdot (1 + t_i)]$ where t_i is the so-called tolerance factor (deviations greater 20% are discarded). This tolerance factor is determined by statistical testing of samples against the above intervals. The second value of each column, Ω_i defines the interval offset for the i-th feature.

The hash values are generated by interval mapping of every single feature against the interval matrix. As performance measurement a FAR of zero and a FRR of 7.05% is reported. Furthermore, they describe the issue of choosing significant features and introduce three measures for feature evaluation: intrapersonal feature deviation, interpersonal entropy of hash value components and the correlation between both.

The proposed scheme is very similar to the above scheme of Feng and Wah. In both schemes feature parameters of online signatures are defined and fitted into intervals, defined with the help of enrollment samples, out of which hashes (in the above scheme the hash is called feature code) are generated, which are then matched to authorized a user.

3.8 Cancellable Biometrics

The basic idea of "cancellable biometrics" is to apply transforms to captured biometric data and furthermore perform the matching process in the transformed space. If biometric data is stolen, lost or comprised only the applied transform has to be changed. Furthermore several different transforms can be used for several applications to prevent imposters from tracking users by cross-matching databases. Figure 3.9 illustrates the idea of cancellable biometrics. Ratha *et al.* [59, 60] introduced the concept of "cancellable biometrics". Biometric data can be compromised and therefore can become useless because it can not be modified ex post. The idea of cancellable biometrics consists of intentional, repeatable distortion of a biometric signal based on a chosen transform. These distortion transforms are selected to be non-invertible, that is the inverse transform is one-to-many. Therefore the recovering of the original biometric data is not possible if an attacker is in possession of the transform function and the transformed biometric data. Additionally, the correlation of several transformed biometric measurements does not reveal any information about the original biometrics.

The distortion of the biometric signal can be either performed in signal domain or in feature domain. Performing distortion in the signal domain means manipulating the raw biometric

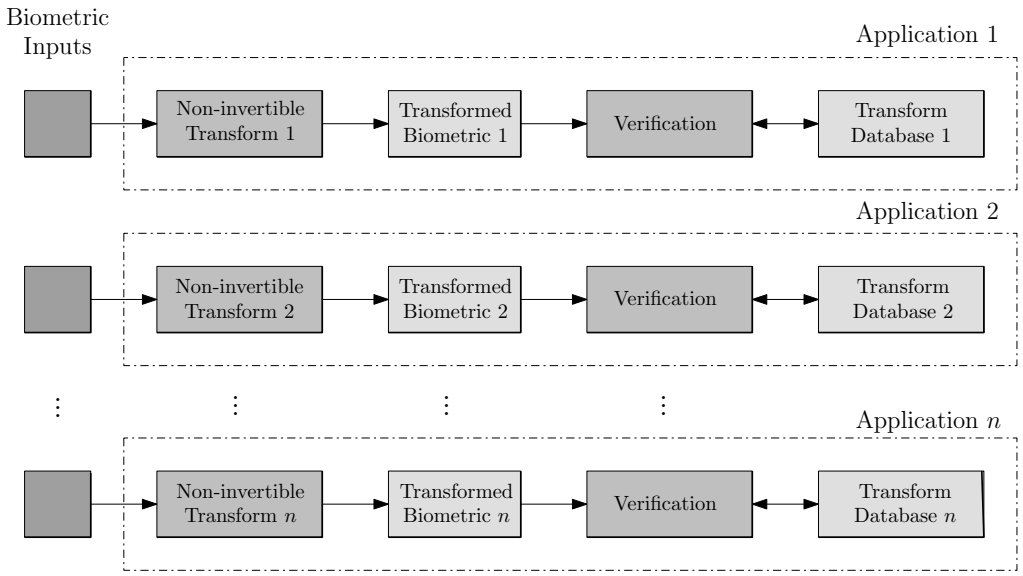

Figure 3.9: The basic idea of cancellable Biometrics

input, which can be achieved by grid morphing where the measured signals are aligned to a specified grid or block permutation where the ordering of the measured data is modified. Performing distortion in the feature domain means that after specific features have already been extracted randomly, repeatable permutations of these features are generated.

They introduce several types of transforms for constructing multiple cancellable biometrics from pre-aligned fingerprints in the feature domain. The first transform is a so-called cartesian transform, which divides a captured image of a fingerprint into cells. These cells are then permutated using a method they refer to as cell-mapping. This cell-mapping is non-injective and thus provides the required non-invertibility. The second proposed transform is called polar transform. This transform operates like a cartesian transform but divides the image of a captured fingerprint into sectors with respect to polar coordinates. Within this transform the mapping is done using translations. The advantage of using polar coordinates is that the permutations do not alter the natural distribution of minutiae points. Nevertheless these two transforms hold the disadvantage of yielding unfeasible cancellable biometrics if in subsequent measurements minutiae points cross the boundaries of cells or sections.

The third type of transforms which are introduced are called functional transforms. These transforms have parametric forms that are governed by random keys. Functional transforms are locally smooth but globally not smooth functions of the x and y-coordinates of every minutiae points.As an example for such a functional transform surface folding transformations are presented. In a surface folding transformation, both the position and the orientation of the minutiae are changed by some parametric transfer function. Conceptually, the minutiae are embedded in a sheet which is then crumpled. Such a function is locally smooth but globally not smooth. Furthermore it is noninvertible because of its foldings.

If the transformed biometric data is compromised just the transform function has to be changed, that is, the biometric template is updated. It is suggested to use different transform functions for different services. The transform function could be stored on a smartcard which is presented during the authentication process. The biometric input is

transformed with this function and compared to the stored transformed template. This is the basic idea of cancellable biometrics which is realized in several forms, for example, by combining biometric data with tokenized random numbers like it is done in the BioHashing approach.

Savvides *et al.* [64] showed a way of how to generate cancellable templates from face biometrics using so-called minimum average correlation energy filters.

In their architecture a user presents a personal PIN to the system, which serves as seed for a random number generator. This random number generator uses the PIN to generate a random convolution kernel (it is suggested that the generation of this kernel should depend on the desired operation of the system). During the enrollment process several images of a person's face are captured. These images are then convolved with the previous generated random convolution kernel. These convolved images are then used to produce a biometric filter.

To generate this filter the before mentioned minimum average correlation energy filters are used. These type of filters provide the invertibility, necessary for cancellable biometrics, because the inverse transform of these filters does not reveal any useful information about the biometrics involved. Furthermore these filters, which are synthesized from several images (unlike common correlation filters), minimize the average correlation energy of the correlation outputs due to the training images. This means that the resulting correlation planes yield values close to zero everywhere except at the location of a trained object. Therefore these filters produce sharp correlation peaks.

This approach is very similar to the BioHashing algorithm which also uses a user specific seed to generate a random basis for the whole algorithm. Here so-called minimum average correlation energy filters are used, which are highly discriminative, and thus it becomes more complicated for imposters to trick the system.

Lee *et al.* [41] presented a method for generating alignment free cancellable fingerprint templates. Very similar to Tulyakov *et al.* [83, 84] and Yang and Verbauwhede [93] orientation information is used for each minutiae point. Such information could include neighboring minutiae points as well as specific angles. The goal is to find such an translation and rotation invariant value m_i for each minutiae point. In their scheme the translation and rotation invariant value m_i is extracted by using the orientation information within a neighboring region around each minutiae point and a user-specific random vector.

If such values are found for each minutiae point these values are fed into two so-called "changing functions" denoted by L_{PIN} (the distance changing function) and Θ_{PIN} (the orientation changing function). These changing functions are user specific functions depending on a user's PIN. Thus cancellability is provided by the user's PIN and the user-specific random vector used to extract translation and rotation invariant values of minutiae points.

3.9 Other Related Work

Sutcu *et al.* [74] proposed another biometric authentication system in which they create hash values out of face biometrics. In their work they focused on how to robustly generate these hashes (feature extraction is not addressed).

During the enrollment process several samples of a user's face are captured and features are extracted resulting in n-dimensional feature vectors $V_i = [v_{i1}, v_{i2}, ...v_{in}]$ where i stands for the i-th user. With the use of several sample for each value of this feature vector a range is calculated so that $\overline{v}_{ij} - \delta ij \leq v_{ij} \leq \overline{v}_{ij} + \delta_{ij}$ where $i = 1, ..., N$ and $j = 1, ..., n$. Thus $2\delta_{ij}$ defines the range of the j-th component of the feature vector of the i-th user.

In the next step a single Gaussian function is fitted to every corresponding range. Additionally, a number of fake Gaussian functions are added. It is suggested that the parameters of the resulting function are stored on a smartcard. The system only stores a hash value of the features generated with a standard hash function (SHA-1 is suggested).

During the authentication process the smartcard is presented to the system and a feature vector is extracted. Then the function of the smartcard is applied to these features and the result is hash. If this hash matches the stored hash, the user is accepted.

The whole scheme is somehow similar to the above schemes applied to online signatures because a range is defined for each feature. But in this work this range is defined by a Gaussian function. Furthermore, noise is added in the form of fake Gaussian functions.

Chang *et al.* [13] proposed a framework for biometric key generation and give a general approach for distinguishable feature generation and a stable key generation mechanism applied to face biometrics. The goal of the distinguishable feature generation is to apply user-dependent transformations such that each transformed feature is distinguishable to separate the authentic user from others and thus potential imposters. Furthermore, by applying these functions the key space is enlarged. Each transformed feature contributes one or more bits to the cryptographic key. A projection vector is calculated which should maximize the ratio of the determinant of the between-class scatter matrix of the transformed features to the determinant of the within-class scatter matrix of the transformed features. For a two-class classification the optimal projection vector is determined as $w = S_w^{-1}(m_a - m_t)/||S_w^{-1}(m_a - m_t)$ where $S_w = 0.5 \cdot (S_a + S_t)$ and S_a and S_t are covariance matrices of features of the authentic user and the mean of all other. The mean of the features for these matrices are m_a and m_t. Thus the mean between the authentic features and the imposter's features is maximized.

The optimal projection is repeated n times resulting in an n-dimensional feature space, where in each dimension the mean between the authentic samples and the mean of the imposters samples is maximized. This process provides the greatest distinguishability between the biometric sample of a single user and all other samples.

Furthermore, a minimal and maximal boundary is defined for each feature (like in the above techniques). In other words, a so-called authentic region is defined for every feature. These regions are specified with unique indices onto which the features are mapped resulting in a biometric key.

This approach is similar to the above mentioned approaches applied to signatures. However, this technique makes use of the information of all accounts and potential imposters (if this information is available) to improve the security of one user. To apply this technique with the use of just a few users could work well, but to apply it to a large number of accounts seems to be a challenging issue, which is not addressed further.

Freire *et al.* [27] presented a general biometric hash generation scheme based on the concatenation of binary strings extracted from a set of feature vector subsets. In this scheme they make use of so-called genetic algorithms to select this subset out of a feature set. The

feature subset concatenation is performed as follows:

Given a feature vector $x = [x_1, ..., x_N]$ with $x_i \in \mathbb{R}$, a biometric hash $h = [h_1, ..., h_L]$ with $h_i \in \{0, 1\}$ of dimension L is extracted. Let x_j with $j = 1, ..., D$ be formed by a subset of features of x of dimension M ($M < N$), with possibly overlapping features for different j. Let C^j be a codebook obtained by vector quantization of x_j using $x_k^j = 1, ..., K$ is a subset of features called development set. The biometric hash h for an input feature vector x_T is defined as: $h(x_T) = concat(f(x_T^j, C^j))$ where $j = 1, ..., D$ and f is a function that assigns the nearest-neighbor codewords. The concatenation of binary strings is denoted by $concat(\cdot)$.

The codebooks C^j are computed with vector quantization as follows. Let $x^j_{k=1,...,K}$ be feature vector subsets forming a development set. The k -means algorithm is used to compute the centroids of the underlying clusters, for a given number of clusters Q. Then, centroids are ranked based on their distance to the mean of all centroids. Finally, binary codewords of size $q = log_2 Q$ are defined as the position of each centroid in the ranking, using Gray coding.

Then genetic algorithms, which are non-deterministic methods inspired by natural evolution, which apply the rules of selection, crossover and mutation to a population of possible solutions in order to optimize a given fitness function, are applied. In the proposed scheme the fitness function of the Genetic Algorithm is defined as $f = \text{EER}^{-1}$, where EER is computed for skilled forgeries from a set, different to the training set. In this work, a Genetic Algorithm with integer coding is implemented in order to obtain the best subsets of M features.

They apply this scheme to signatures and report an EER of 18.83% for skilled and 8.02% for random forgeries for a hash length of 75 with 25 subsets chosen out of which the hash is chosen. In conclusion, this approach demonstrates how an integer-coding genetic algorithm enables exploiting all information found in large feature sets.

Zhang and Chen [95] presented a generalized optimal thresholding method, which they apply to face images. The goal is to find the optimal threshold for each feature to minimize FAR and FRR and maximize the guessing entropy. The guessing entropy is the expected number of guesses an imposter has to make to receive the authentic key. Since it is not easy to maximize the guessing entropy directly, they maximize its lower bound in the optimization, the Shannon entropy.

Given a face image, a set of biometric features are extracted (feature extraction is not addressed further). The feature extraction is performed several times for the authentic user and the imposters. As result of all the measurements a threshold T_i for each feature x_i in the feature vector x is calculated. This threshold separates an authentic decision region R_{Ai} from an imposter decision region R_{Ii} (like the threshold the decision regions are defined for each feature).

This approach is very simple but the generation of the thresholds depends on the quality of the imposter data with which the system is trained but would not function without.

Chen and Chandran [15] proposed another cryptographic key generation system for face biometrics where they applied two different transforms and Reed-Solomon codes.

The first transform is a so-called Radon transform, which is not further explained. This Radon transform is used to convert a 2D image into a set of 1D projections. The result, a 1D vector, is fed into a second transform. The second transform is an interactive chaotic bispectral one way transform that accepts a one-dimensional vector input and outputs a

magnitude and an angle matrix. During enrollment several sample inputs are captured and the most desireable bits are selected. This is done by selecting those bits with the lowest intra-class entropy and the highest extra-class entropy. The inter class entropy is calculated out of an intra-class set which is a set of matrices generated from images of one user. The extra-class entropy is calculated out of an extra-class set, which is a set of matrices generated from images of all users. The detected bits form the feature vector.

The rest of the system operates like a password hardening scheme [52] except that Reed-Solomon codes are used instead of shares created according to Shamir's secret sharing scheme. For the generation of 128-bit keys a FAR of 1.22% and a FRR of 28% are reported.

In this work another way of improving the distinguishability of a feature vector by using samples of potential imposters and maximizing the "distance" to those is presented. The presented results are not satisfying, they are able to produce very long keys up to 240-bit.

Delivasilis and Katsikas [22] proposed a novel method of applying side channel attacks to a biometric key generator for voice. Side channel attacks fall into the category of passive attacks where physical characteristics of an algorithm's operation are measured. The result of such a side channel attack could be (in the best case) additional information about the cryptographic operations of the algorithm. This additional information is called information leakage. This model has derived from research studies that have shown that the power consumption of an algorithm's operation with various inputs generates power consumption traces with small but existent variations.

These variations are correlated with the Hamming distance of the input data. At the best the attacker is able to make several hypotheses about the nature of the plaintext (in this case the biometric).

As a result a tolerance function was identified. Every time a small fluctuation of the recorded signal was generated the algorithm used that specific function to test whether the input is within the acceptable range. This is an interesting pilot work which demonstrates a way of how an attacker is able to extract accurately parts of a biometric authentication system' algorithm and thus about a user's biometric.

3.10 Discussion

Most of the above schemes fit into the class of key binding systems. In these schemes biometric data is combined with some sort of cryptographic keys during enrollment. At the time of authentication this key is regenerated. Thus the objective of such schemes is to use biometrics to realize a cryptographic key out of a template into which this key was binded during registration.

In the Biometric Encryption^{TM} algorithm first several images of a person's biometrics are combined using correlation. The use of correlation is introduced to deal with the variance of the biometric inputs. Afterwards a look-up table is generated which binds the person's biometrics with a cryptographic key. Within this scheme the key could either be an existing key or a randomly generated one. It is mentionable that this algorithm processes entire images of biometrics and does not perform feature extraction.

One very simple approach of binding a cryptographic key with biometric data is presented in

the fuzzy commitment scheme. In this scheme the binding is performed by simply XORing biometric data with the key which is prepared with an error correction code. The use of error correction codes should handle the fuzziness of biometric measurements. Any sort of preprocessing and feature extraction depends on the measured biometric characteristics.

The fuzzy vault scheme, which in other words is a proper fuzzy commitment scheme, uses feature sets instead of feature vectors to overcome the problem of order invariance (especially with the use of fingerprint biometrics). Here the binding of a person's biometric with a cryptographic key is performed by embedding extracted features onto a polynom which is defined by the key. If enough points are found on this polynom it can be reconstructed and thus the key. The security of this scheme lies in the infeasibility of the polynomial reconstruction.

The secure sketch as well as the fuzzy extractor aim at extracting a random bitstream out of a biometric data. Additionally information is provided which can be bound with the biometric data to reconstruct the secret bitstream.

In the BioHashing algorithm biometric data is denoised and features are extracted which are then binded with orthonormalized random vectors by computing dot products. These random vectors are generated out of a user specific random token. Finally a key is generated by thresholding the results of the dot products. Thus in this algorithm the user specific token provides the additional information to receive the correct key.

In conclusion all of these key binding schemes store some sort of information generated during registration which is then used to either generate a cryptographic key or release one. Furthermore in these schemes cryptographic keys are updateable due to the fact that these are not generated directly out of a person's biometric.

The private template scheme represents a classic key generation scheme. Here the user's biometrics or a hashed value generated directly out of it serves as biometric key. The key is suggested to be used to encrypt any form of private information. Thus the biometric data (raw or hashed) has to be stored as biometric template which leads to the name of the scheme. This makes the scheme very insecure in case the private template is stolen. In other words, if this private template is compromised the whole system gets useless. Therefore this approach has not been proceeded any further.

Schemes which use defined boundaries to generate hashes out of biometric data also fit into the class of key generation schemes. The hashes, which are obtained by mapping biometric features into intervals and afterwards stored in a database, do not seem to be updateable if these are comprised.

Increasing the security of a password-based authentication system has been presented as the so-called password-hardening scheme. In the first approach a user's keystrokes dynamics, out of which features were extracted, have been used to make a password more secure. The system is designed to detect distinguishable features of a person's keystroke dynamics which are used in combination with an instruction table to generate a hardened password. Thus a password hardening scheme does not aim at generating a cryptographic key but enhances the security of an allready existing key. Furthermore within this approach the system is able to adapt to slightly changes in the person's biometrics. This is achieved by, instead of using enrollment samples like in the above schemes, using the last few biometric samples of successful logins.

Another way to classify biometric cryptosystems is to focus on how these systems deal with the fuzziness of biometrics. There are several ways to overcome variance. While in some schemes transforms or filters are applied to denoise biometric measurements in other schemes deviations may be calculated for each extracted feature.

In the Biometric Encryption approach a filter function is generated for each user with the use of correlation to denoise the biometric input. Applying this filter function suffices to extract identical biometric keys for legitimate users while imposters are rejected. The tolerance of the whole system is adjusted by binding the adequate parameters of these filter functions.

User-specific random numbers are combined with biometric features in the BioHashing scheme. Subsequently thresholds are applied to generate a biometric hash out of these numbers. This means in the BioHashing approach these thresholds define some kind of filter which should provide a variation tolerant hash generation.

While in the Biometric Encryption scheme as well as in the BioHashing scheme filters are applied to denoise the entire biometric input in the Biometrically Hardened Password approach only a subset of the extracted features is used, namely these which are the most distinct. This means if features tend to vary from one measurement to another these features are not used in the authentication process. Thus here occurring variations are detected and not employed.

Another way to deal with biometric variance is to introduce error correction codes. In the Fuzzy Commitment scheme as well as in the Fuzzy Vault scheme a secret key is prepared with error correction codes and bound with the biometric data. Secure Sketches and Fuzzy Extractors provide information to reconstruct a random bitstream which is extracted from a biometric input. In the Private Template scheme error correcting information is appended to the extracted features. At the time of authentication this additional information is used to correct a distinct amount of variance. In these schemes the amount of error correcting information which is used defines the tolerance of the whole system.

In so-called Interval schemes deviations of features form the biometric template. This implies distinct features create tight boundaries while unsteady features stretch boundaries. This approach is somehow similar to the Biometrically Hardened Password approach because only distinct features will separate a legitimate user from imposters. Thus to deal with the variance of the extracted features in interval schemes wider boundaries are defined for features which tend to underlie variations.

Many publications aim at providing so-called cancellable biometrics. Within this approach the objective is to transform the measured biometric data either in signal or in feature domain. Repeatable distortions of the biometric data are suggested to create these transformed versions of the biometric data. The applied transformations need to be noninvertible and random, so that neither a transformed biometric sample and the transformation nor several samples of transformed biometrics reveal any information about the original measurement. Furthermore it is suggested to use different transformations for different applications to prevent imposters from crossmatching databases to track users. In case a transformed sample of a user's biometric data is stolen the parameters of the transformation used to create this sample are simply changed, which means the stored sample is updated. Most of the schemes providing cancellable biometrics make use of pseudo random number generators. This pseudo random number generators serve as basis of the whole system.

All the above mentioned key binding schemes provide a form of cancellable biometrics though

not in the classic sense of cancellable biometrics. Within key binding schemes the cryptographic key is updateable. For example in a fuzzy vault scheme there are no transformations which could be changed but applying different polynoms would suffice to update the scheme if the biometric template was comprised. Nevertheless the classic key generation systems like the private template scheme do not provide cancellable biometrics.

Chapter 4

Iris-based Fuzzy Commitment Schemes

In this chapter two implementations of iris-based fuzzy commitment schemes are proposed. These implementations use different iris recognition algorithms and error correction codes are adapted to each algorithm. First of all, the basic working flow of a fuzzy commitment scheme is described (Section 4.1) and consequential preconditions are defined (Section 4.2). Then the general way of how to make use of a fuzzy commitment scheme in a biometric cryptosystem is described (Section 4.3). Subsequently, two forms of error correction codes are presented (Section 4.4). Then a short summary of the applied iris recognition algorithms is given (Section 4.5, 4.7). At the end implementations of fuzzy commitment schemes using these algorithms are discussed and experimental results are presented (Section 4.6, 4.8).

4.1 A Fuzzy Commitment Scheme

A fuzzy commitment scheme is a cryptographic primitive which was introduced by Juels and Wattenberg [36]. Used in a biometric cryptosystem the objective of a fuzzy commitment scheme is used to bind a biometric sample (for example, an iris code) of a user with a cryptographic key. The result of this binding is stored as template (commitment). The cryptographic key is previously prepared with an error correction code so that another biometric sample, which is very similar to the first, extracts the cryptographic key from the template.

To understand the fuzzy commitment approach the operating mode of a simple bit commitment scheme has to be explained first. Formally, a bit commitment scheme consists of a function

$$F : \{0,1\} \times X \to Y \tag{4.1}$$

To commit a bit $b \in \{0,1\}$ the sender chooses a so-called "witness" $x \in X$ (this witness is chosen randomly) and calculates the so-called "blob" y so that

$$y = F(b,x) \tag{4.2}$$

Thus y represents b in a "safe". To open (decommit) this safe the sender sends b and the receiver verifies if y is an encryption of b. In other words, the sender passes the safe to the receiver, which is not able to manipulate the value of b. The receiver can only check if y is an encrypted version of b. For the receiver it must be infeasible to learn anything about b out of a given y.

Based on the idea of the bit commitment scheme a fuzzy commitment scheme exhibits several differences. First of all a fuzzy commitment scheme can be defined by

$$F : C \times X \to Y. \tag{4.3}$$

Here C is a set of error correcting codewords of length n, $X = \{0,1\}^n$ and $F(c,x)$ is the commitment of c where $c \in C$ and $x \in X$. Secondly, in a fuzzy commitment scheme it is possible to extract c out of $F(c,x)$ using a valid x which is not the case in a bit commitment scheme. Thus any codeword $c \in C$ represents a secret which for instance could be a cryptographic key. Thirdly, the most important property which is required is the fuzziness. This means that the blob y can be decommited by x' where x' is a witness which is sufficiently close to x according to some metric, for example, the Hamming distance. These three properties summarize the fuzzy commitment scheme.

A fuzzy commitment scheme can be applied to a biometric system in combination with error correction codes. This is done to overcome the fuzziness in the biometric matching process.

With respect to biometric systems, a fuzzy commitment scheme consists of a set of codewords $C \subseteq \{0,1\}^n$ where each codeword $c \in C$ represents a secret bitstream prepared with error correcting information. This error correction code C has to be chosen with respect to the biometrics, which means, the error correction code has to be adapted to the way errors occur in the general matching process. If mostly single bit errors occur, bit level error correction codes would suffice. If mostly burst errors occur block level error correction codes would suffice. If both type of errors occur a combination of both would do.

One of these codewords $c \in C$ is then committed using x, where $x \in \{0,1\}^n$ is a biometric sample, for example an iris code. Here both, the codeword and the biometric sample must be of the same length. Then the commitment is generated by calculating a so-called difference vector δ so that

$$x = c + \delta \tag{4.4}$$

This means that now δ provides partial information about c. To prove whether the correct c was extracted additionally a hash of c, $h(c)$ where h is a secure hash function, is calculated as part of the commitment so that

$$y = (h(c), \delta) \tag{4.5}$$

The hash value of c, $h(c)$, must not reveal any information about c.

Now to overcome the fuzziness of biometrics it should be possible to transform a biometric sample x' which is sufficiently close to x into the direction of x using δ, This is done by first calculating c' so that

$$c' = x' - \delta \tag{4.6}$$

If c' is calculated, the decoding function f of the error correction code C can be applied so that if the decoding function is able to correct all errors

$$f(c') = f(x' - \delta) = c \tag{4.7}$$

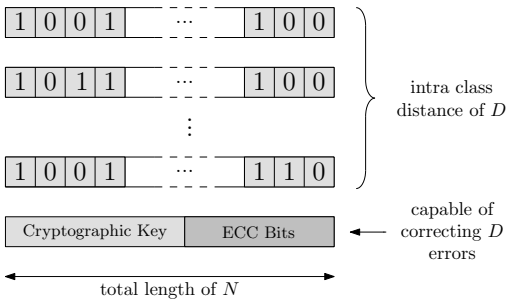

Figure 4.1: A simplified precondition for a fuzzy commitment scheme: the number of additional error correction bits should suffice to correct the a number of bits which is at least the intra class distance which can be specified as a fixed threshold.

The resulting codeword is then tested by applying h to it. If the result is equal to $h(c)$ then no errors occurred. If the result is not equal to $h(c)$, one or more errors must have occurred.

4.2 Preconditions for Fuzzy Commitment Schemes

Several preconditions evolve from the above description of a fuzzy commitment scheme. For the construction of an iris based fuzzy commitment scheme the following preconditions must be fulfilled:

First of all, the iris recognition algorithm which is applied to the biometric input needs to generate a rotation invariant output. Contrary to a fuzzy vault scheme a fuzzy commitment scheme, as it is described above, cannot handle order variant outputs.

Another precondition which has to be fullfiled is that the applied iris recognition algorithm generates a sufficiently large output (=bitstream). A cryptographic key which is prepared with an error correction code has to be bound with this output. This key has to be prepared with a sufficiently large number of error correction bits. These additional bits should provide the information to repair a distinct number of bit errors. This number should approximate the intra-class distance of the applied algorithm as it is shown in Figure 4.1.

In summary, to create a fuzzy commitment scheme for the use in a biometric cryptosystem the applied iris recognition algorithm should produce an order invariant output. Furthermore, this output should be as long as the cryptographic key concatenated with error correction bits where the error correction bits provide the information to correct the estimated number of errors between iris data of the same person.

4.3 Preface

If the above preconditions are fullfilled, which means that enough bits are extracted form the biometric input and the right error correction codes are chosen, the generation of the fuzzy commitment scheme is straight forward.

First of all, the key $k \in \{0,1\}^l$, which can be chosen at random has to be prepared with error correction information. The way the key is encoded depends on the error correction code which is used. Nevertheless, the error correction code adds redundant information to

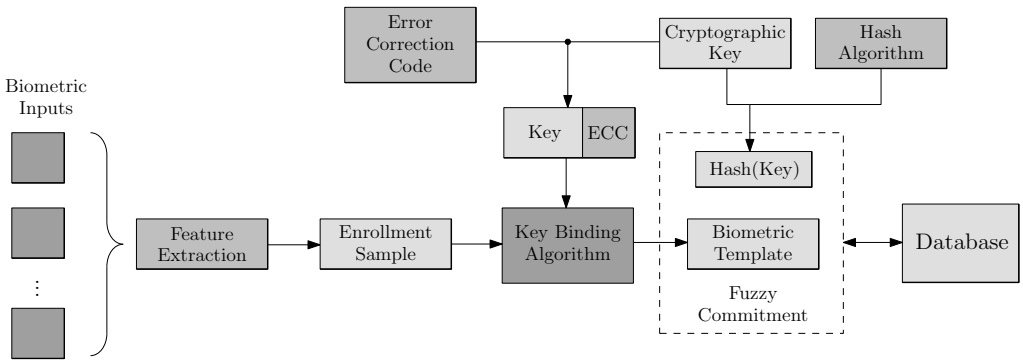

Figure 4.2: The basic enrollment procedure in a fuzzy commitment scheme used in a biometric cryptosystem.

the key and thus maps the key to another bitstream. The bit stream is of same length as the biometric sample $x \in \{0,1\}^n$ so that it is possible to XOR these two bitstreams. The result is stored as the commitment together with a hash of the key.

To decommit the commitment another biometric sample $x' \in \{0,1\}^n$ is XORed with the commitment and the result is decoded with the decoding function of the error correction code. The result of this decoding procedure is a key $k' \in \{0,1\}^l$ of length l. Now this key could be returned by the system, which means rejecting a user would result in returning a faulty key. Furthermore, a hashed value of the resulting key could be tested against $h(k)$ to whether return the valid key or reject the presented biometric sample.

In biometric systems a so-called enrollment procedure defines the registration of a person. Analogical the authentication procedure defines the verification of a person. In the following subsections these two procedures are defined for a fuzzy commitment scheme:

4.3.1 Enrollment in a Fuzzy Commitment Scheme

At the start of the enrollment procedure in a fuzzy commitment scheme one or more biometric inputs are acquired. The number of biometric input used depends on the applied algorithm, if the algorithm itself uses several biometric inputs to create the biometric template at least the same number of biometric inputs should be used. Once enough biometric inputs are acquired, feature extraction is performed and all samples are combined to form an enrollment sample $x \in \{0,1\}^n$ so that

$$x = f(x_1, x_2, ..., x_n) \tag{4.8}$$

where f is a function which combines a total number of n biometric samples $x_i \in \{0,1\}$.

Concurrently a randomly chosen key $k \in \{0,1\}^l$ is provided with error correcting information. This is done by applying an error correction code C which is none other than a set of codewords $c \in C$, so that with the encoding function of C, k is mapped to a bitstream of length n so that

$$C_{enc} : \{0,1\}^l \rightarrow \{0,1\}^n \tag{4.9}$$

where C_{enc} is the encoding function of C. The result denoted by $k_{ec} \in \{0,1\}^n$ is then combined with x (both bitstreams are of length n) using a so-called key-binding algorithm.

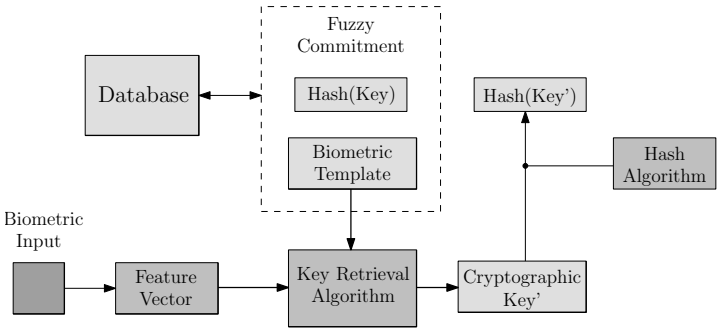

Figure 4.3: The basic authentication procedure in a fuzzy commitment scheme used in a biometric cryptosystem.

One simple and concealing key binding algorithm is XORing both bitstreams to form the first part of the commitment so that

$$y = (k_{ec} \oplus x, h(k)) \tag{4.10}$$

where y is the whole commitment and $h(c)$ is a hash of the original key, created using a secure hash function. In summary this means the result of the enrollment procedure is a biometric template resulting out of several biometric inputs combined with a cryptographic key and a hash value of this key. The whole enrollment procedure is illustrated in Figure 4.2

4.3.2 Authentication in a Fuzzy Commitment Scheme

During the authentication process one biometric input is acquired, feature extraction is performed resulting in an biometric sample $x' \in \{0,1\}^n$ of length n. This biometric sample is then used within a so-called key retrieval algorithm to extract k'_{ec} so that

$$k'_{ec} = x' \oplus (k_{ec} \oplus x) \tag{4.11}$$

where $(k_{ec} \oplus x)$ is part of the commitment. Once k'_{ec} is calculated and k'_{ec} does not contain too many errors the decoding function of C denoted by C_{dec} is used to calculate k' so that

$$C_{dec} : \{0,1\}^n \rightarrow \{0,1\}^l \tag{4.12}$$

where $k' \in \{0,1\}^l$. Now k' can be returned which implies that a non-valid user would receive a faulty key, $k' \neq k$. To check whether k' is the correct key $h(k')$ is calculated and tested against the hash value of the correct key $h(k)$, which is stored as part of the commitment. If $h(k') = h(k)$, the sample is accepted and the cryptographic key k is released, otherwise the user is rejected. Subsequently the released key is used to encrypt/decrypt any kind of secret information. The whole authentication procedure is illustrated in Figure 4.3.

4.4 Error Correction Codes

With respect to the construction of a fuzzy commitment scheme two classes of error correction codes are relevant:

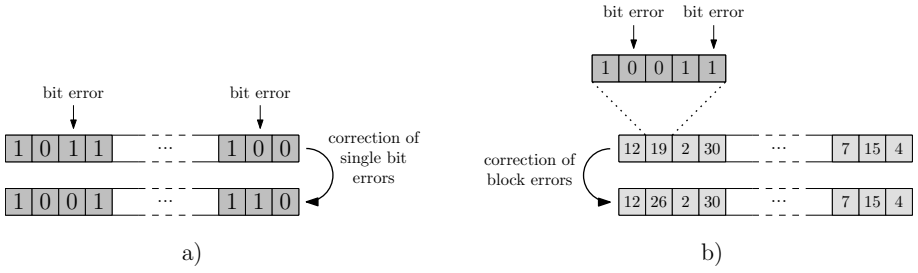

Figure 4.4: Two classes of error correction codes: a) an error correction code operating on bit level b) an error correction code operating on block level.

1. **Bit Level Error Correction Codes**: These codes are capable of correcting single bit errors while the error correction information lies within each codeword.

2. **Block Level Error Correction Codes**: These codes are capable of correcting blocks of bits in which errors occur. Here the error correction information is provided by other bit blocks. Thus the information is distributed over a set of bit blocks.

In Figure 4.4 the basic operations of these two classes of error correction codes are illustrated. In the following subchapters two representative error correction codes of these two classes are described in detail.

4.4.1 Hadamard Codes

Named after Jacques Hadamard, Hadamard codes are error correction codes which operate on the bit level [3][31][47]. Hadamard codes are of the type $\left[2^n, n+1, 2^{n+1}\right]$, which means bitstreams of length $n+1$ are mapped to codewords of length 2^n and the whole code consists of a total number of 2^{n+1} codewords. Hadamard martices are used to generate a Hadamard code. A Hadamard matrix of order n is a matrix H_n with elements 1 and -1 such that $H_n H_n^t = nI_n$ where H_n^t is the transposed matrix of H_n and I_n is the identity matrix. This is the first important property of a Hadamard matrix. For example,

$$H_1 = \left[1\right], H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{4.13}$$

Hadamard matrices of size $n \times n$, where $n > 2$ only exist if 4 divides n. Furthermore, since $H_n H_n^t = nI_n$ two different rows of H_n are orthogonal. Any matrix obtained from the permutation of rows or columns of H_n is a Hadamard matrix as well as $-H_n$.

If H_n is a Hadamard matrix, n can be written as $n = 4m$. Let $v = (v_1, v_2, ..., v_n)$ be a vector of length n so that if v is added to an arbitrary row of H_n and $u = v + h_k$ has at most $m-1$ components different from h_k where h_k denotes the kth row of H_n. Then the kth component of $s = uH_n^t$ is at least $4m - 2(m-1) = 2m + 2$ and the absolute value of other components is at most $2(m-1)$. This is the second important property of an Hadamard matrix.

To create a Hadamard matrix the Kronecker product of two Hadamard matrices can be

applied where the Kronecker product of two matrices A and B is defined as,

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix} \tag{4.14}$$

A class of Hadamard matrices are the so-called Sylvester Matrices, which are the Kronecker product of H_2 and an arbitrary Hadamard matrix H_n such that,

$$H_{2n} = H_2 \otimes H_n, \text{ and } H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4.15}$$

Therefore these Hadamard matrices can be defined recursively so that,

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix} \tag{4.16}$$

Once a Hadamard matrix H_{4m} is created (for example with the Sylvester method), a code can be generated out of it. For the code H_{4m} and $-H_{4m}$ are used where first -1 values are changed to 0. The resulting matrices $\hat{H}_{4m}$ and $\hat{H'}_{4m}$, where

$$\hat{H}_{4m} = \left[-H_{4m} + 2J_{4m}\right] \mod 3 \text{ and } \hat{H'}_{4m} = \left[H_{4m} + 2J_{4m}\right] \mod 3 \tag{4.17}$$

where $2J_{4m}$ is the $4m \times 4m$ matrix in which all values are 1. Then all rows of the matrix C of size $8m \times 4m$ where,

$$C = \begin{pmatrix} \hat{H}_{4m} \\ \hat{H'}_{4m} \end{pmatrix} \tag{4.18}$$

defines the codewords of the resulting Hadamard code. Thus a Hadamard matrix H_{4m} of size $4m \times 4m$ generates a Hadamard code consisting of $8m$ codewords, each of length $4m$. The minimum distance of these codewords is $2m$ which corrects up to $m - 1$ errors. As defined above, $n + 1$ input bits are then mapped to a codeword of length 2^n. For the construction of a fuzzy commitment scheme this means that for a key K of length k, at most $2^{k-1} - k$ bits are appended to the key bitstream.

The decoding procedure, which just makes use of the above two properties of Hadamard matrices, for a received Hadamard code can be summarized in five steps:

1. All the bits of a received codeword c, of length $4m$, which are 0 are first changed into -1 resulting in c' where $c' = 2c - h_1$ (h_1 is the first row of a Hadamard matrix H_4m created with the Sylvester method, thus each value of h_1 is 1).

2. Multiply the received codeword with H_{4m} so that $s = c'H_4m$

3. If c' is a codeword of the Hadamard code without any errors s will be $4m \cdot e_k$ or $-4m \cdot e_k$ where e_k is the kth row of the the identity matrix I_{4m} (remember that $H_n H_n^t = nI_n$).

4. If errors occurred, s will not be $\pm 4m \cdot e_k$. But if the number of errors is at most $m-1$, then the largest absolute component of s indicates which codeword (row of H_{4m}) is the received codeword because the largest absolute value may not decrease below $2m+4$ and the absolute value of all other components may decrease down to $2m-2$.

5. If more than $m-1$ errors occurred, the codeword cannot be decoded. This means that the largest absolute value is not unambiguous.

In summary, Hadamard codes are simple error correction codes capable of correcting a large number of errors, in the best case a quarter of the whole bitstream. Hadamard codes using Hadamard matrices of size $n < 7$ have been proved to be optimal.

4.4.2 Reed Solomon Codes

Reed Solomon Codes are error correction codes named after I. Reed and G. Solomon [33] which are based on the arithmetic of finite fields. These codes are suitable for the correction of burst errors and are thus often used as a second error correcting layer.

Within Reed Solomon codes the code is defined as the mapping of a vector space of dimension m over a finite field k denoted by $V_m(k)$ in a vector space of higher dimension n, $n > m$, over the same field denoted by $V_n(k)$. Let $k = \mathbb{Z}_2(\alpha)$ where α is the root of a primitive irreducible polynomial over $\mathbb{Z}_2$. Then the elements of k are represented as $0, \beta, \beta^2, ..., \beta^{2^{n-1}}$ where β is the generator of the multiplicative cyclic group and $\beta^{2^{n-1}} = 1$. If a code is sent, the code is represented as $(a_0, a_1, ..., a_{m-1})$ where $a_i \in \mathbb{Z}_2$. Subsequently $P(x)$ is defined as

$$P(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{m-1} x^{m-1} \tag{4.19}$$

which means the Reed Solomon code maps the code so that

$$(a_0, a_1, ...a_{m-1}) \rightarrow (P(0), P(\beta), P(\beta^2), ..., P(1)). \tag{4.20}$$

This result is then sent and afterwards decoded. The decoding of a received Reed Solomon code can be described as follows: If no error occurred, then 2^n equations are received such that

$$\begin{aligned} P(0) &= a_0 \\ P(\beta) &= a_0 + a_1\beta + a_2\beta^2 + ... + a_{m-1}\beta^{m-1} \\ P(\beta^2) &= a_0 + a_1\beta^2 + a_2\beta^4 + ... + a_{m-1}\beta^{2m-2} \\ &\vdots \\ P(1) &= a_0 + a_1 + a_2 + ... + a_{m-1} \end{aligned} \tag{4.21}$$

where all equations are linearly independent, which means an unique solution exists. In case errors occurred, one fact is utilized, namely that for s errors, we can get at most $(s+m-1) - m_c$ determinations for a single wrong m-tuple where m_c denotes a chosen m. This is because, since the equations are independent, any m of them have exactly one solution vector a. To obtain more than one vote, a must be the solution of more than m equations. An incorrect a can be the solution of at most $s+m-1$ equations, consisting of s incorrect equations and $m-1$ correct equations. Therefore, an incorrect a can be the solution to at

most $(s+m-1)-m_c$ sets of m equations. Thus, the correct solution $a=(a_0,a_1,...,a_{m-1})$ gets at least 2^n-s-m_c votes.

An incorrect solution $b_1,b_2,...,b_i$ where $t \leq s$ each solution gets at most $s+m-1-m_c$ votes. Thus if

$$2^n-s-m_c > s+m-1-m_c \tag{4.22}$$

which means that

$$s < \frac{2^n-m+1}{2} \tag{4.23}$$

then by taking the majority vote, the correct m-tuple can be determined resulting in the original message. Therefore a total number of $\frac{s^n-m+1}{2}$ errors can be corrected.

At the time Reed Solomon codes were invented, computers were not capable of decoding such a code. Berlekamp [5] was the first to invent an efficient algorithm to decode Reed Solomon codes.

4.5 An Iris Recognition Algorithm using Cumulative-Sum based Change Analysis

The first applied iris recognition algorithm performs simple texture analysis. The algorithm is proposed by J.-G. Ko, Y.-H. Gil, J.-H. Yoo and K.-I. Chung [38]. It employs a cumulative-sum-based grey change analysis for the feature extraction.

As in common iris recognition algorithm first the iris texture is extracted. The enhanced image is then divided into cells out of which mean gray scale values are calculated. Subsequently these mean values are used to form cumulative sums which are encoded to form the iris code. In the authentication process a distance, which is called the lower Hamming distance (see 4.5.3), is calculated between a sample iris code and a reference iris code. This distance is compared against a fixed threshold deciding whether a person is accepted or rejected. In the following subchapters the preprocessing of the algorithm is summarized:

4.5.1 Preprocessing

First of all one iris image is acquired for each person. Then the inner and outer boundaries of the iris need to be localized on the acquired image. Like in John G. Daugman's algorithm this is done by using an effective integro-differential operator:

$$\max(x_0, y_0, r) \left| G_\sigma(r) \cdot \frac{\partial}{\partial r} \oint_{x_0, y_0, r} \frac{I(x,y)}{2\pi r} ds \right| \tag{4.24}$$

where $I(x, y)$ is the original image. The complete operator behaves as a circular edge detector. It searches iteratively over the candidate domain with respect to increasing radius denoted by r, $G_\sigma(r)$ is a smoothing function, for example, a Gaussian function of scale σ.

In the next step the detected texture is normalized. This means, the iris ring is mapped from polar coordinates to cartesian coordinates to form a rectangular image of the iris texture. The size of the normalized image is 64×300. Due to the fact that the top and bottom of the iris are often hidden by eyelashes or eyelids, the iris image from the right side [45^o to 315^o] and the left side [135^o to 225^o] are transformed into a polar coordinate system. Furthermore, histogram stretching is applied to improve the contrast of the iris texture. The whole preprocessing procedure is illustrated in Figure 4.5.

4.5.2 Feature Extraction

Once an iris texture of fixed size is extracted and prepared, feature extraction is performed. This is done by applying cumulative-sum-based change analysis. The whole feature extraction can be divided into four steps:

1. The iris texture is divided into so-called basic cell regions (these cell regions are of size 10×3 pixels). For each basic cell region an average grey scale value is calculated.
2. The basic cell regions are grouped horizontally and vertically. It is recommended that one group should consist of five basic cell regions.
3. The cumulative sums over each group are calculated.

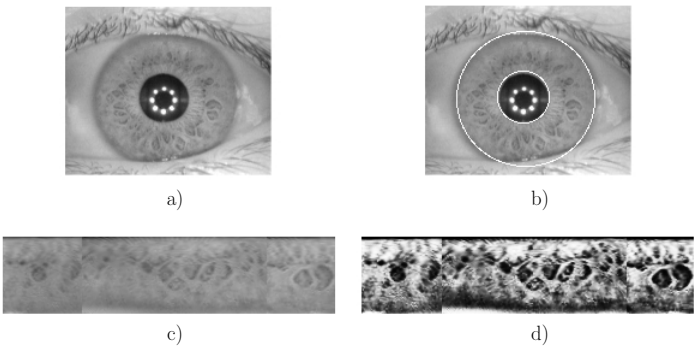

Figure 4.5: The preprocessing procedure: a) aquisation of the iris image b) detection of the inner and outer iris boundaries c) normalization of the iris texture d) enhancement of the iris texture

4. The iris code is generated out of the cumulative sums.

The cumulative sums of five average gray scale values of the basic cell regions of one group are calculated as follows:

Let $X_1, X_2, ...X_5$ be five average gray scale values. First calculate $\overline{X}$, the mean of these values so that

$$\overline{X} = \frac{X_1 + X_2 + ... + X_5}{5} \tag{4.25}$$

Then the initial cumulative sum S_0 is defined by $S_0 := 0$. The other cumulative sums are then defined recursively so that for $i = 1, 2, ..., 5$

$$S_i = S_{i-1} + (X_i - \overline{X}) \tag{4.26}$$

This means, cumulative sums are calculated by adding the difference between current value and the average to the previous sum (note that the calculation of cumulative sums only requires addition and substraction). The cumulative sums are not the cumulative sums of the values. Instead they are the cumulative sums of differences between the values and the average. These differences sum up to zero so the cumulative sum always ends at zero.

The iris code for the cumulative sums is generated as follows:

1. Calculate the minimum and the maximum for the group of cumulative sums so that $MAX := \max(S_1, S_2, ..., S_5)$ and $MIN := \min(S_1, S_2, ..., S_5)$.
2. If S_i is located between the index of MIN and MAX, check whether S_i is on a upward slope or on a downward slope:
   (a) If S_i is on upward slope, set the cell's iris code to 1.
   (b) If S_i is on downward slope set the cell's iris code to 2.
3. If S_i is located outside the index of MIN and MAX, set the cell's iris code to 0. An example of the iris code generation for two sample groups is shown in Figure 4.6.
4. Repeat 1.-3. for each group of cumulative sums.

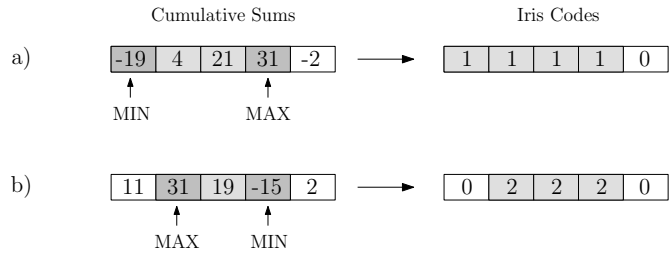

Figure 4.6: Iris code generation: a) an example of an iris code including an upward slope b) an example of an iris code including a downward slope

Analyzing cumulative sums means analyzing the variations in the grey values of iris patterns. If the cumulative sums of a group are on an upward slope the iris pattern may change from darkness to brightness. If the cumulative sums of a group are on an downward slope, the iris pattern may change from brightness to darkness.

In summary, the preprocessing and the feature extraction together form the enrollment procedure. As a result of the enrollment procedure, an iris code, which is generated out of the horizontal and vertical groups, is stored as biometric template.

4.5.3 Verification

At the time of authentication the similarity of two iris codes is calculated using a metric which is referred to as lower Hamming distance. A low distance indicates a high similarity and vice versa. This lower Hamming distance HD is calculated as

$$HD = \frac{1}{2N}\left[\left(\sum_{i=1}^{N} A_h(i) \oplus B_h(i)\right) + \left(\sum_{i=1}^{N} A_v(i) \oplus B_v(i)\right)\right] \quad (4.27)$$
$$\text{only when } A_h(i) \neq 0 \wedge B_h(i) \neq 0,\ A_v(i) \neq 0 \wedge B_v(i) \neq 0,$$

where $A_h(i)$ and $A_v(i)$ denote the enrolled iris code over the vertical and horizontal direction and $B_h(i)$ and $B_v(i)$ denote the iris code over the vertical and horizontal direction of an acquired biometric input, N is the total number of basic cell regions.

4.5.4 Experimental Results

The proposed algorithm was evaluated using the a CASIA iris database [1] with data from 108 people (no information is provided about the total number of iris images and the version of the database). In Table 4.2 the best results give by the authors with respect to the applied threshold are summarized.

To confirm these results an own implementation of the proposed algorithm was implemented and tested using a subset of the CASIA v3 database [1], where for each person at least 8 iris images are available, which makes a total number of about 100 persons. Unfortunately the results of Ko *et al.* could not be approved with respect to the accuracy of the algorithm, altough the implementation uses exactly the same algorithm as described above. Still the proposed algorithm based on the differences in upward and downward slopes of cumulative sums shows good results in distinguishing authentic from non-authentic users. In Figure

Lower Hamming Distance	False Acceptance Rate (%)	False Rejection Rate (%)
0.24	0	1.14
0.25	0.42	1.14
0.26	1.05	0.76
0.27	2.63	0.50
0.28	5.26	0.25

Table 4.1: The best proposed results with respect to the applied threshold. All results are percentaged.

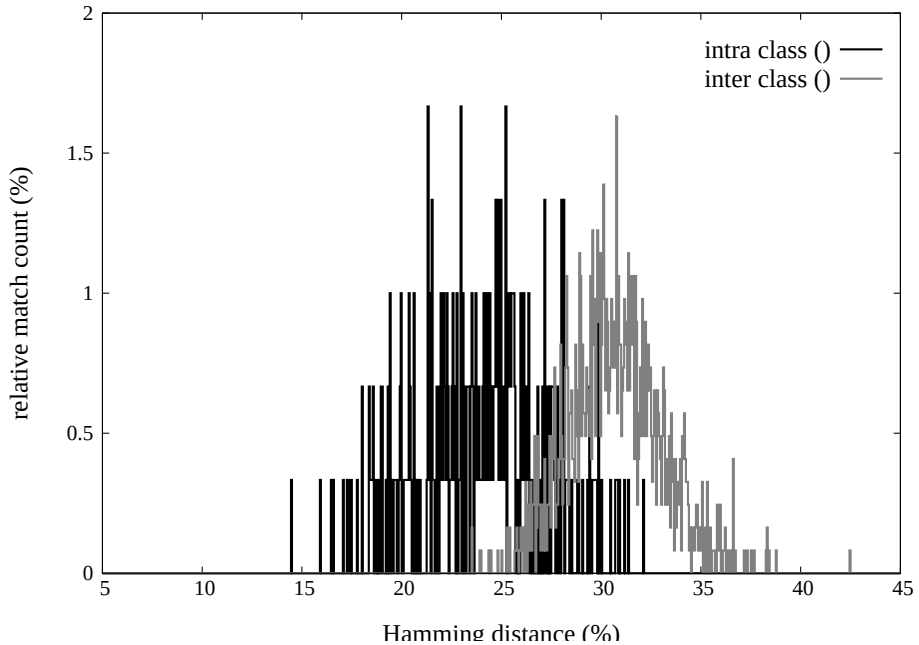

Figure 4.7: The Hamming distance between iris codes generated based on the above algorithm. By calculating the Hamming distance inter class distances and intra class distances strongly overlap.

4.8 the inter class distances and the intra class distances of the iris codes is shown. These distances were achieved by applying the suggested metric (lower Hamming distance) of the above algorithm. In contrast, Figure 4.7 illustrates the Hamming distances (XOR of tertiary code) of the same set of iris images where intra class distances strongly overlap with the inter class distances. This implies, by using cumulative sums for feature extraction, the lower hamming distance as defined above shows much better results in distinguishing genuine user from imposters than the Hamming distance. In Figure 4.9 the FNMR and the FMR of the proposed algorithm is plotted, Figure 4.10 shows the ROC of the algorithm.

In summary the algorithm which is very simple does not approve the proposed results. A success rate of 90.5% was achieved, which means for a FMR of 0.0% a FNMR of 9.5% was achieved, and a EER of 4.7%. It is conspicuous that the average lower Hamming distance of genuine users and imposters is much less than those presented by Ko *et al.*. None the less this algorithm can be used to build up a fuzzy commitment scheme, but the suggested metric of counting differences in upward and downward slopes of cumulative sums cannot be applied. This is because in a fuzzy commitment scheme the whole iris code is XORed with

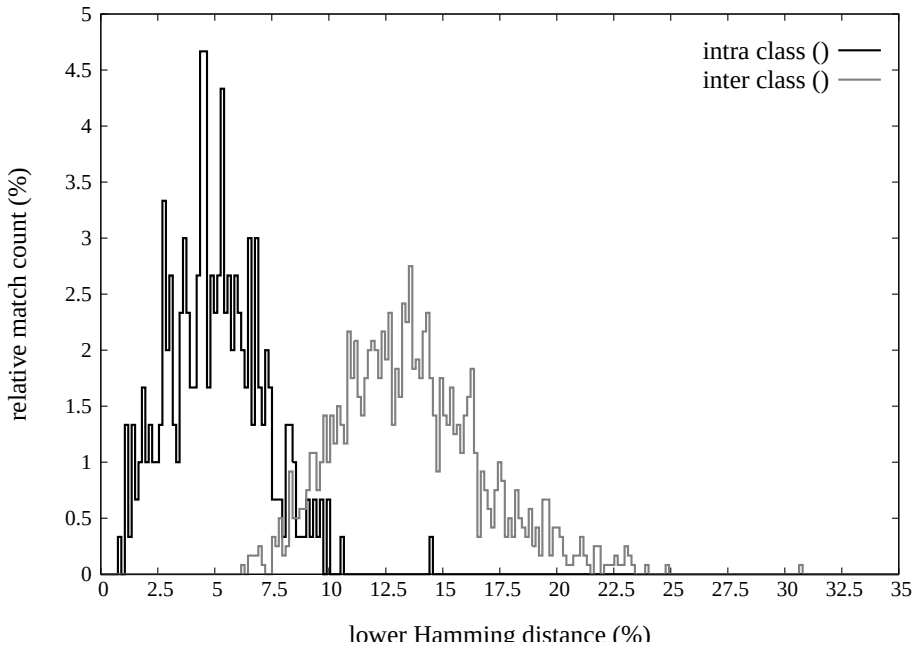

Figure 4.8: The distance of iris codes according to the suggested metric of the above algorithm. The difference in upward and downward slopes of cumulative sums shown good results for distinguishing authentic from non-authentic users.

a cryptographic key combined with error correcting information while in the above metric only distinct parts of the iris codes are used to calculate differences between these.

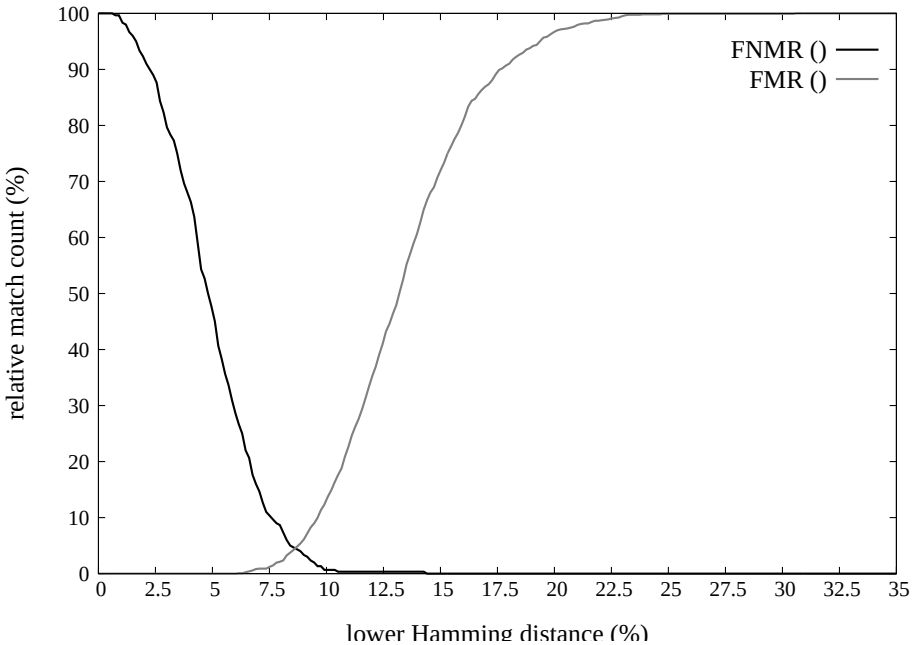

Figure 4.9: The FMR and the FNMR of the implementation using the above algorithm. For evaluation about 100 persons of the CASIA v3 iris database were used.

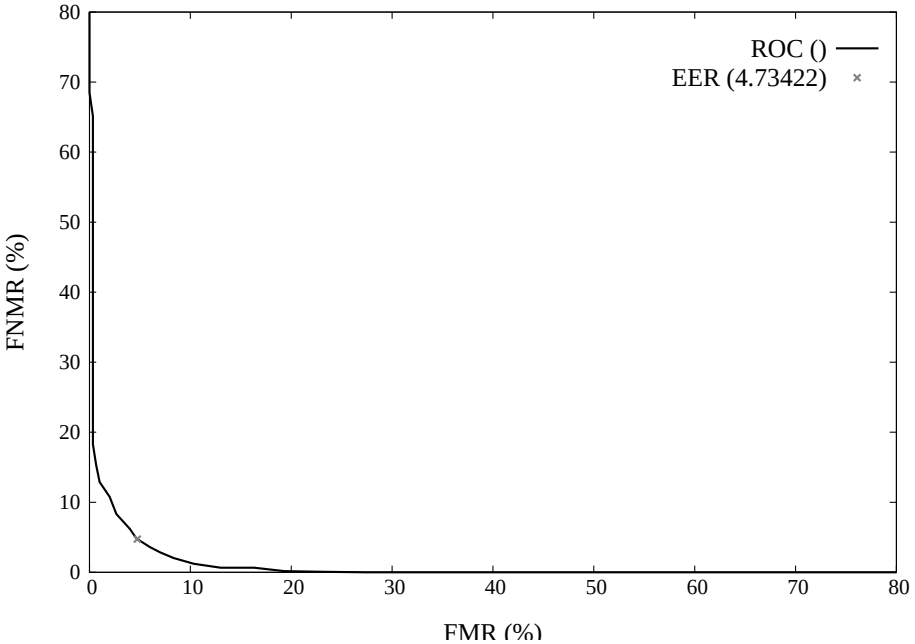

Figure 4.10: The ROC and the EER of the implementation using the above algorithm. For evaluation about 100 persons of the CASIA v3 iris database were used.

4.6 A Fuzzy Commitment Scheme using Block Level Error Correction Codes

The algorithm of J.-G. Ko, Y.-H. Gil, J.-H. Yoo and K.-I. Chung [38] turns out to be well suited for a fuzzy commitment scheme using block level error correction codes which is capable of binding and retrieving cryptographic keys with a total length of 128 bits.

4.6.1 Iris Code Generation

First of all, to build up the fuzzy commitment scheme the above preconditions have to be fulfilled. This means the applied algorithm has to generate a sufficiently large number of output bits. The general iris code of the algorithm of Ko *et al.* can be defined as follows,

$$C = \{0, 1, 2\}^n \tag{4.28}$$

where n is the length of the iris code. To keep up the semantics of the iris code the mapping of the code to a binary level implies doubling the length of the iris code so that

$$C' = \{0, 1\}^{2n} \quad 0 \rightarrow 00,\ 1 \rightarrow 01,\ 2 \rightarrow 10 \tag{4.29}$$

where C' is the resulting binary iriscode, which now consists only of 1 and 0 generated according to the above mappings. Using the above mapping implies a loss of entropy but handling the original iris code as tertiary code leads to several problems in the matching process since simple XORing is not possible then. In Figure 4.11 the mapping of a sample iris code is shown. This mapping decreases the entropy of the iris code by 25% because a pair of bits cannot consist of two 1s.

In the preprocessing step iris images of dimension 512×64 are slitted from the right side [45^o to 315^o] and the left side [135^o to 225^o] to get rid of most distortions resulting in an iris image of dimension 256×64. As in the original algorithm, the iris code consists of two parts, one for cumulative sums of the horizontal and one for the vertical groups of basic cell regions. As suggested, basic cell regions of dimension 10×3 and groups of five basic cell regions were used to generate the iris code. Thus the center 250×60 pixels of the slitted iris image are used as information. These 250×60 pixels lead to 25×20 basic cell regions resulting in a total number of 200 groups out of which 2000 output bits are extracted. In Figure 4.13 the whole process which leads to 2000 output bits is summarized. Since 2000 bits are enough bits to bind a 128 bit cryptographic key within a fuzzy commitment scheme the precondition of extracting a sufficiently large number of output bits is fulfilled.

One property of the resulting bitcode is utilized: namely that within a group a cumulative sums can either contain an upward slope or a downward slope but it cannot contain both since there is only one minimum and one maximum. This means that the iris code can be split into two parts where the first part contains all the upward slopes and the second part contains all the downward slopes. In Figure 4.14 an example of this mapping is shown.

By dividing the iris code into two parts, one for each type of slopes, the errors which occur between two iris codes do not decrease but accumulate, which means single bit errors which are close to each other are pulled together. This means, with this algorithm the number of burst errors increases while the number of bit level errors decreases. Figure 4.12 shows an example of the rearranging of the iris code, which is now even more qualified for using it in combination with a block level error correction code.

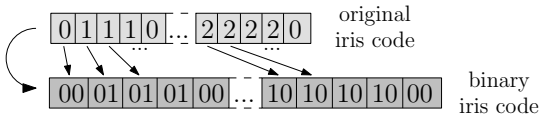

Figure 4.11: A sample iris code which is mapped to the analogue binary iris code. The mapping implies a loss of entropy but holds the semantics of the iris code.

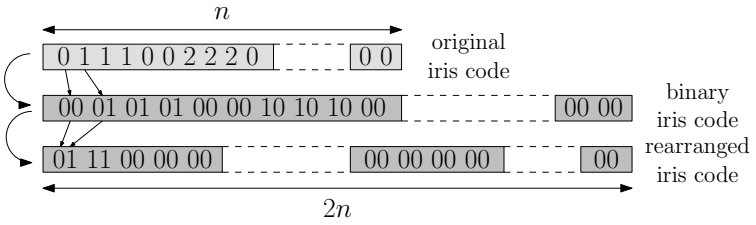

Figure 4.12: A sample iris code which is rearranged so that the first half of the rearranged iris code contains all upward slopes of cumulative sums and the second part of the rearranged iris code contains all downward slopes of cumulative sums.

Intruders may take advantage of the fact that the entropy is reduced by the mapping of the original iris code to a binary iris code. For instance, in the binarized iris code as it is shown in Figure 4.11 a pair of bits can at most contain one 1. By analogy, if in the rearranged iris code a 1 is at position m implies that there is a 0 at position $m+n \mod n$ where the entire iris code consits of n bits. Thus given one half of the iris code intruders would be able to construct several parts of the remaining half of the iris code. However, this security leakage is annihilated in the key binding process.

4.6.2 The Enrollment Process

To enroll one person a total number of three input images are acquired and three iris codes are generated according to the above algorithm. Out of these three iris codes one enrollment iris code is generated by first filling gaps between bit groups. Subsequently, for each bit the majority bit is assigned to the enrollment iris code. In the end, sequences of 1s which are of length $l \leq 2$ are replaced by 0s. The procedure of filling gaps of size $s \leq 2$ is illustrated in Figure 4.15 while the elimination of 1s is shown in Figure 4.16.

Furthermore, a randomly generated 128 bit key k is encoded with a Reed-Solomon code. The code used a symbolsize $m = 8$ which means that it is capable of generating encoded blocks of 256×8 bits. Since the key consists of 16×8 bits and the enrollment iris code consists of a total number of 250×8 bits, the key is encoded using a $RS(250, 16)$ Reed Solomon code. Thus 16×8 information bits are encoded using $(250 - 16) \times 8$ bits. This implies that the decoding function is able to correct $(250 - 16)/2 = 117$ block level errors.

Additionally, a hash value of the key denoted by $h(k)$, where h is a secure hash function, can be stored as part of the biometric template.

4.6.3 The Authentication Process

At the time of authentication one iris image is acquired and an iris code is generated according to the above method. This iris code which consists of a total num-

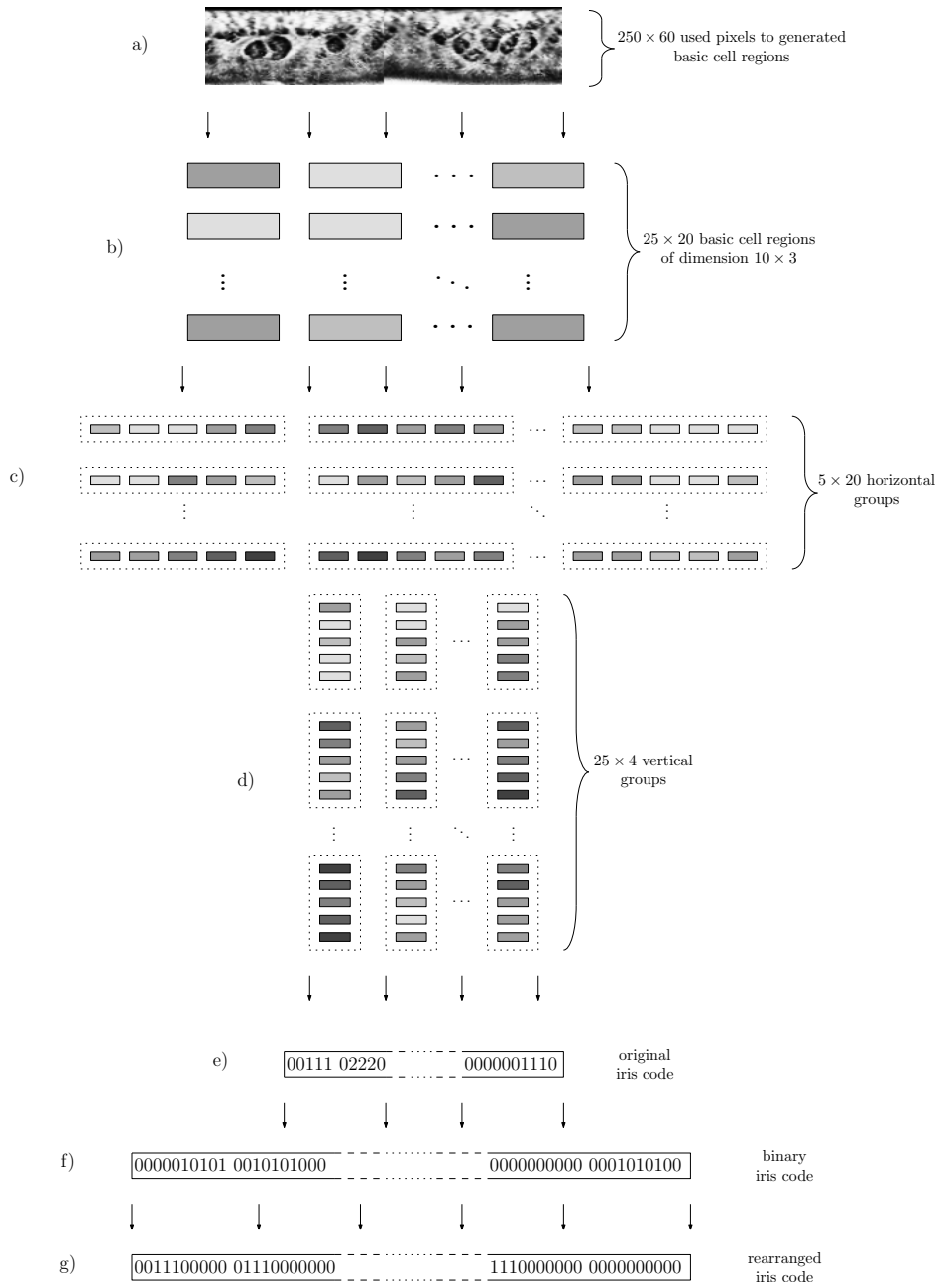

Figure 4.13: The whole process to generate the iris code: a) use the center 250 × 60 pixels of the slitted image texture as input b) calculation of the 25 × 20 basic cell regions c) calculation of the 5 × 20 horizontal groups d) calculation of the 25 × 4 vertical groups e) generation of the original iris code f) binarize the original iris code g) rearrange the binarized iris code

ber of 2000 bits is XORed with the template. The resulting bitstream is the decoded with the Berlekamp-Massey algorithm [5] for the decoding of Reed-Solomon codes (code is derived from software contributed to GMD-FOKUS by C. Schuler,

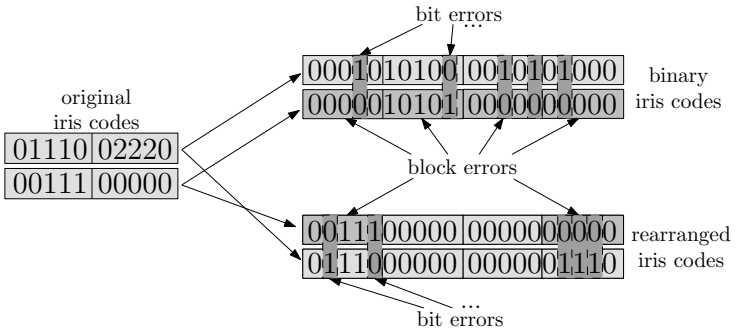

Figure 4.14: Two sample iris codes where in the binary iris codes due to five single bit errors four block level errors occur while in the rearranged version of the binary iris codes only two block level errors result out of five bit level errors.

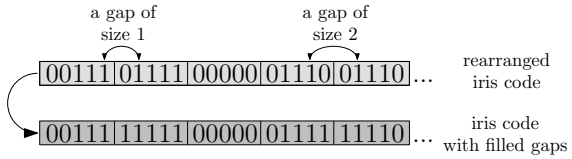

Figure 4.15: Gaps between two upward/downward slopes of cumulative sums of size $s \leq 2$ are filled to decrease the number of single bit errors which often occur at the borders of two groups.

http://home.arcor.de/christianschuler/fec_sw.html). As a result of the decoding algorithm 128 bit cryptographic key k' is returned.

In the end this key can be tested against the stored hash value of the correct key by applying h to k' and check whether $h(k') = h(k)$. If so, the sample is accepted, otherwise it is rejected.

4.6.4 Experimental Results

The fuzzy commitment scheme, as it is described above, was tested using a subset of the CASIA v3 database [1], where for each person at least 8 iris images are available, which makes a total number of about 100 persons. In Figure 4.17 the distribution of the intra-class and inter-class distances with respect to the number of block level errors is shown. Without the rearrangement described in Figure 4.12 the intra class distances regarding to block level errors increase and a significant part of genuine users cannot be authenticated as shown in Figure 4.18. As described above a total number of 117 block level errors can be corrected. For most of the authentic users correcting 117 block level errors suffices. There is only a small number of samples which contain slightly more errors, but some which contain considerable more block level errors.

This phenomenon can be explained by the fact that 8-bit blocks are used for the Reed-Solomon code, while the applied algorithm operates on 5-bit blocks. Thus one 5-bit block level error may either cause one or two block level errors on the 8-bit block level. Therefore, some block level errors result from other block level errors. Due to this reason the accumulation of the inter class distances is not very high.

In Figure 4.20 the FMR and the FNMR of the whole system is plotted, Figure 4.21 shown the ROC resulting in an EER of 5.7%. This result is not very satisfying but still a FNMR of

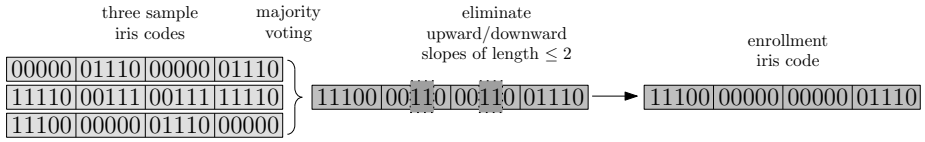

Figure 4.16: Eliminating sequences of of 1s of length $l \leq 2$ is important because these sequences cause errors if these are compared to iris codes where only sequences of 1s of length $l > 2$ are accepted.

6.57% and a FMR of 0.083% where reached setting up a border for being capable of correcting a total number of 117 block level errors. This means in the proposed fuzzy commitment scheme a total number of 117 faulty bit blocks of the overall 250 bit blocks can be corrected. Thus, if less than 46.8% of the bit blocks of the iris code contain errors the correct key is returned. If more than 46.8% of the bit blocks contain errors is is not possible to decode the bitstream which means a faulty key is returned.

According to the presented results it is shown that the usage of block level error correction codes is suitable for the above algorithm. In contrast, applying only bit level error correction codes is not adequate. With regard to the intra class Hamming distance of extracted iris codes as shown in Figure 4.7, bit level error correction codes, such as Hadamard codes, would be expected to at least authenticate a part of genuine users. However, if for instance Hadamard codes are applied, 8-bit blocks would be mapped to 128-bit blocks to generate a bitstream of approximately 2000 bits (actually $16 \cdot 128$). Then the added error correction information would be capable of correcting a total number of $128/4 - 1 = 31$ bits in each 128-bit block. Unfortunately errors between iris codes do not occur uniformly random. If more than 31 errors occur within at least one of the 128-bit blocks a user would be rejected. In Figure 4.19 the maximal number of intra class and inter class errors within 128-bit blocks is plotted. It is shown that the iris codes of all genuine users possess 128-bit blocks which contain more than 31 errors. Thus the use of bit level error correction codes in combination with the above algorithm is not adequate.

4.6.5 Conclusion

The proposed fuzzy commitment scheme is based on a very simple algorithm, which provides fast feature extraction. The generated iris codes of the applied algorithm are then adapted to minimize the number of burst errors. Altough the results of the applied algorithm were not satisfying the results of the proposed fuzzy commitment scheme, which does not use the same metric as the original algorithm, shows good results.

Furthermore by slightly modifying the iris code of the original algorithm the fuzzy commitment scheme does require only one type of error correction, namely block level error correction.

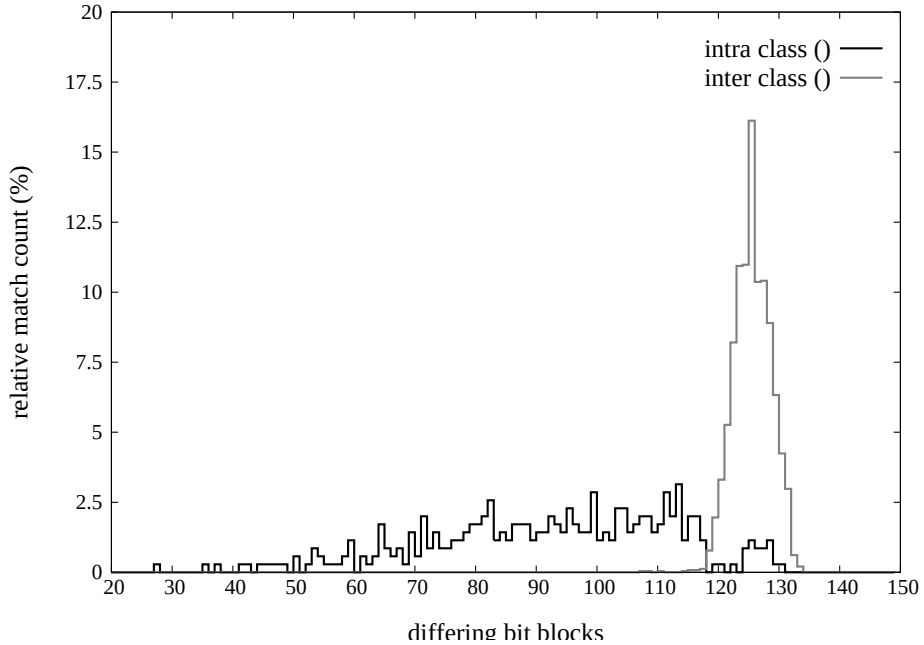

Figure 4.17: The inter-class and intra-class distances of the proposed fuzzy commitment scheme with regard to the total number of block level errors. For evaluation about 100 persons of the CASIA v3 iris database were used.

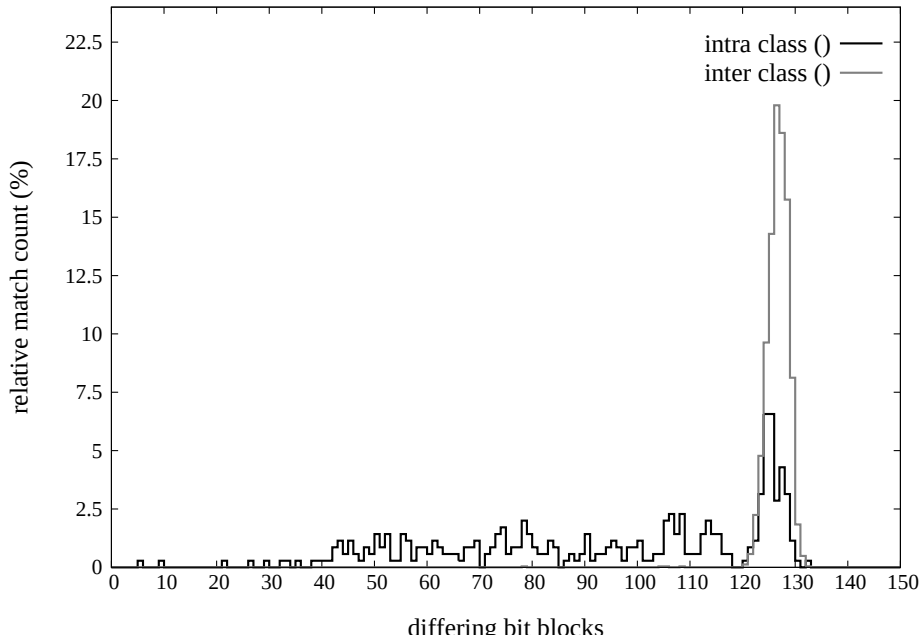

Figure 4.18: The inter-class and intra-class distances of the above algorithm without rearranging the bits of the extracted iris code with regard to the total number of block level errors. For evaluation about 100 persons of the CASIA v3 iris database were used.

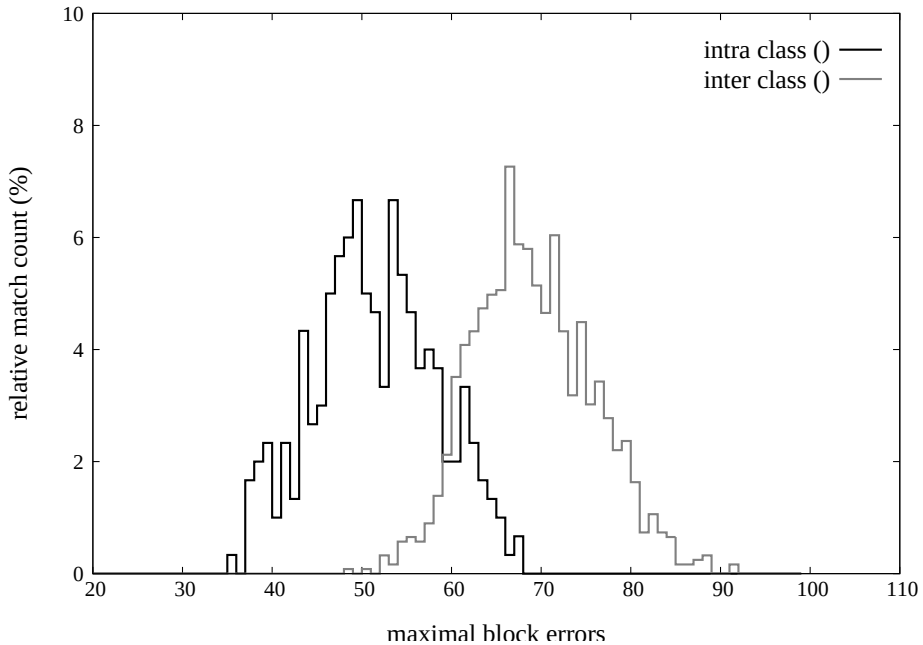

Figure 4.19: The inter-class and intra-class distances of the above algorithm according to the maximal number of errors within 128-bit blocks. For evaluation about 100 persons of the CASIA v3 iris database were used.

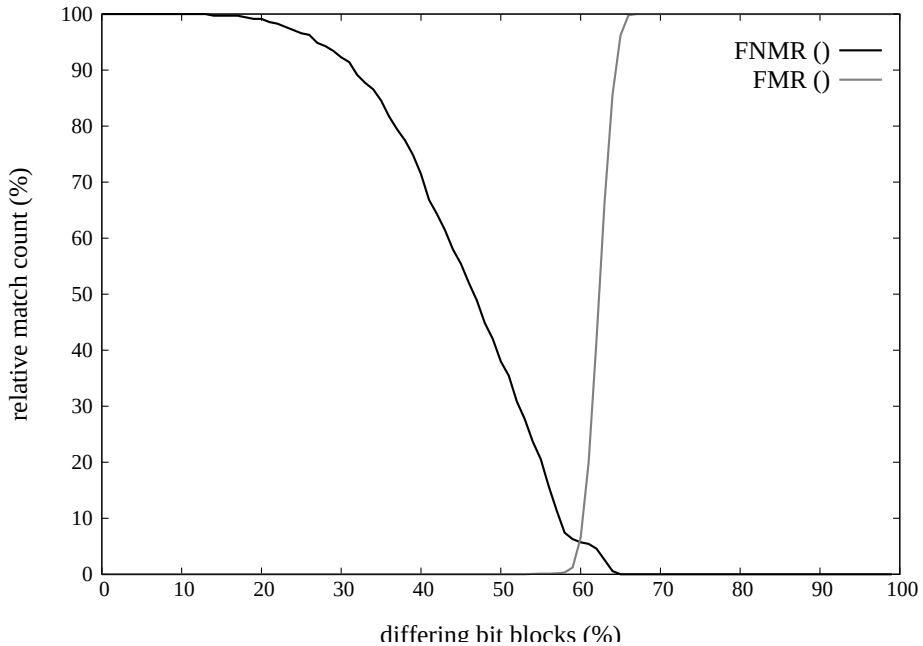

Figure 4.20: The FMR and the FNMR of the fuzzy commitment scheme based on the above algorithm. For evaluation about 100 persons of the CASIA v3 iris database were used.

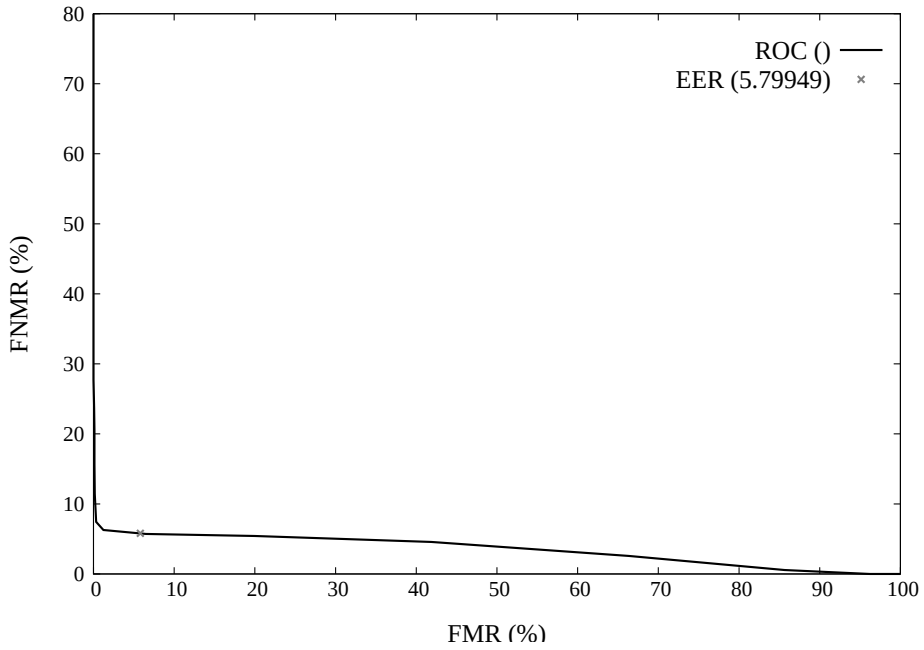

Figure 4.21: The ROC and the EER of the fuzzy commitment scheme based on the above algorithm. For evaluation about 100 persons of the CASIA v3 iris database were used.

4.7 An Iris Recognition Algorithm using Characterization of Key Local Variations

L. Ma, T. Tan, Y. Wang and D. Zhang [45] proposed an iris recognition algorithm which is invariant to translation, scale and rotation. In this approach the iris texture is treated as a kind of transient signal which is processed using wavelet transform. The local sharp variation points, which denote important properties of transient signals, are recorded as features. In the following subchapters the preprocessing, feature extraction and the matching procedure of the algorithm are summarized:

4.7.1 Preprocessing

After the acquisition of the iris image first the iris has to be localized. This means the inner and outer boundaries (two non-concentric circles) of the iris have to be detected. First the iris region is roughly determined and subsequently edge detection and Hough transform is used to calculate the detailed parameters of the two circles. This process comprises several steps:

1. First the image is projected in vertical and horizontal direction to approximately estimate the center coordinates of the pupil, denoted by (X_p, Y_p). Since the center of the pupil is mostly the darkest point of the pupil these coordinates are estimated by calculating the minima of the two projection values so that

$$\begin{aligned} X_p &= \arg\min_x \left(\sum_y I(x,y) \right) \\ Y_p &= \arg\min_y \left(\sum_x I(x,y) \right) \end{aligned} \tag{4.30}$$

   so that X_p and Y_p denote the center coordinates of the iris image $I(x, y)$.

2. In the second step more accurate center coordinates are calculated. Therefore a 120×120 region, centered at (X_p, Y_p), is binarized by adaptively selecting reasonable thresholds using the grey level histogram of this region. The centroid of the region becomes the new center of the pupil. This step can be repeated several times to improve the calculated center.

3. In the third step the parameters of the two circles are calculated by using the edge detection and Hough transform.

After the boundaries of the iris have been detected, the determined iris is normalized. Thus the annular iris is unwrapped counter-clockwise to a rectangular iris block with a fixed size. The normalization process reduces the distortion of the iris caused by pupil movement and furthermore simplifies subsequent preprocessing.

To get a more well-distributed texture the intensity variations across the whole image are approximated. This is done by calculating 16×16 blocks of background illumination

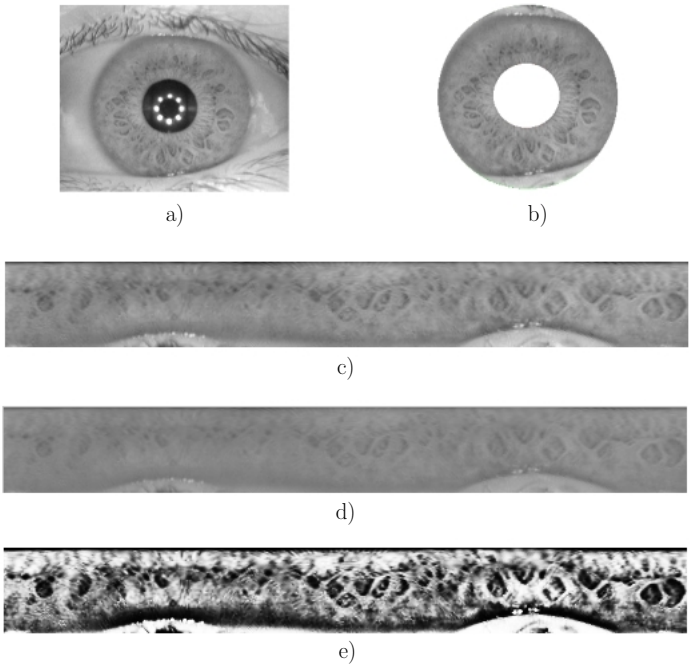

Figure 4.22: The preprocessing procedure: a) image acquisition b) iris localization c) iris normalization d) estimation of the local average intensity e) image enhancement

(= the mean of each grey scale block) and subtracting it from the original image. Once the lightning is corrected the image is enhanced by means of histogram stretching.In Figure 4.22 the whole preprocessing procedure is illustrated.

4.7.2 Feature Extraction

The details of the iris, the intensity signals, are generally spread along the radial direction which corresponds to the horizontal direction in the normalized image. For feature extraction special 1D-wavelets are used where the wavelet function is a quadratic spline of a finite support. The image I is decomposed so that:

$$S_i = \frac{1}{M} \sum_{j=1}^{M} I_{(i-1)\cdot M+j} \qquad i = 1, 2, ..., N \tag{4.31}$$

where S is a set of 1-D intensity signals and the normalized image I consisting of K rows of gray values and can be defined as:

$$I = \begin{pmatrix} I_1 \\ I_2 \\ \vdots \\ I_K \end{pmatrix} = (I_1^T, I_2^T, ..., I_K^T)^T \tag{4.32}$$

The total number of rows used to form a signal S_i is denoted by M, which means that a combination of M horizontal rows is used to detect local variations. Since iris regions close

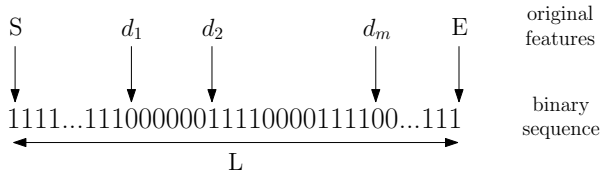

Figure 4.23: The feature transform: The length L of a binary sequence at a scale is the same as the length of the 1-D intensity signal. Furthermore, at each feature component d_i the binary signal changes form 1 to 0, or vice versa (the first $d_1 - 1$ bits are set to 1 if $p_1 = -1$, and 0 if $p_1 = 1$.

to the sclera do not contain much information only the top-most 78% section is used for feature extraction. Therefore a relation between the total number of rows K and the total number of signals N can be denoted by $K \times 78\% = N \times M$. For the experimental results M was set to 5 and N to 10.

As wavelet transform dyadic wavelets are used which is able to decompose signals into detail components. Using these wavelets local sharp variations can be detected. Dyadic wavelets vary along the dyadic sequence $(2^j)_{j \in z}$ and the dyadic wavelet transform of the signal $S(x)$ at scale 2^j is defined as:

$$WT_{2^j} S(x) = \frac{1}{2^j} \int S(X) \psi \left(\frac{x - X}{2^j} \right) dX \tag{4.33}$$

where $\psi \left(x/2^j \right)$ is the wavelet function at scale 2^j, which is a quadratic spline which has a compact support and one vanishing moment. The local minimum of this wavelet denotes the appearing of an irregular block and a local maximum of this wavelet denotes the vanishing of an irregular block. Furthermore, two scales are used. For each signal S_i the so-called position sequences at two scales are concatenated to form the features so that,

$$f_i = \{d_1, d_2, ..., d_m; d_{m+1}, d_{m+2}, ..., d_{m+n}; p_1, p_2\} \tag{4.34}$$

where the first m positions result from the first scale and the second n position result from the second scale of the intensity signal and p_1 and p_2 denote the property of the first variation points. If, for example, the first variation point d_1 is a local minimum then p_1 is set to 1 and otherwise to -1. The concatenation of all features of all intensity signals form the resulting feature vector f, which is defined as,

$$f = \{f_1, f_2, ..., f_N\} \tag{4.35}$$

where f_i denotes the features of the ith intensity signal at two scales. In the end the resulting feature vector consists of a total number of 660 components. Additionally, to save memory for each feature component d_i only the difference d_i' to the previous feature component is stored.

4.7.3 Matching

For the matching process first a binary sequence of each feature vector is generated. The generation of such a binary sequence for one intensity level is illustrated in Figure 4.23. By

generating a binary sequence for the whole feature vector the feature vector can be written as,

$$Ef = \{Ef_{(1,1)}, Ef_{(1,2)}, ..., Ef_{(N,1)}, Ef_{(N,2)}\} \tag{4.36}$$

where $Ef_{(i,1)}$ and $Ef_{(i,2)}$ are the resulting binary sequences for the ith intensity signal at the first and second scale. Since the length of $Ef_{(i,j)}$ is L, the resulting binary feature vector consists of a total number of $2L \times N$ bits. Thus a similarity function for two different feature vectors Ef^a and Ef^b can be defined as:

$$D = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2L} \sum_{j=1}^{2} \left(Ef^a_{(ij)} \oplus Ef^b_{(ij)} \right) \tag{4.37}$$

As mentioned before, scale invariance is achieved by normalization of the image. To additionally achieve rotation invariance the normalized iris image is circularly shifted to a certain degree and the minimal matching score is calculated. This can be done very easily since a rotation of the original iris texture is just a translation in the normalized iris texture.

4.7.4 Experimental Results

For the performance evaluation of the proposed scheme the entire CASIA iris database was used, resulting in a total number of 7223 intra-class comparisons and 2297019 inter-class comparisons. For an implementation of this algorithm an EER of 0.09% has been claimed.

The performance of an own implementation of the proposed algorithm was evaluated using the entire CASIA v3 iris database [1] to confirm the proposed results. In Figure 4.24 the distribution of the intra-class distance and the inter-class distance are plotted. Compared with the proposed distributions, these results are slightly worse. In Figure 4.25 the FMR and the FNMR are plotted and Figure 4.26 shown the ROC resulting in an EER of 1.07%. Nevertheless, the implementation of the proposed algorithm shows good results and the generated feature vectors are well suited for building up a biometric cryptosystem.

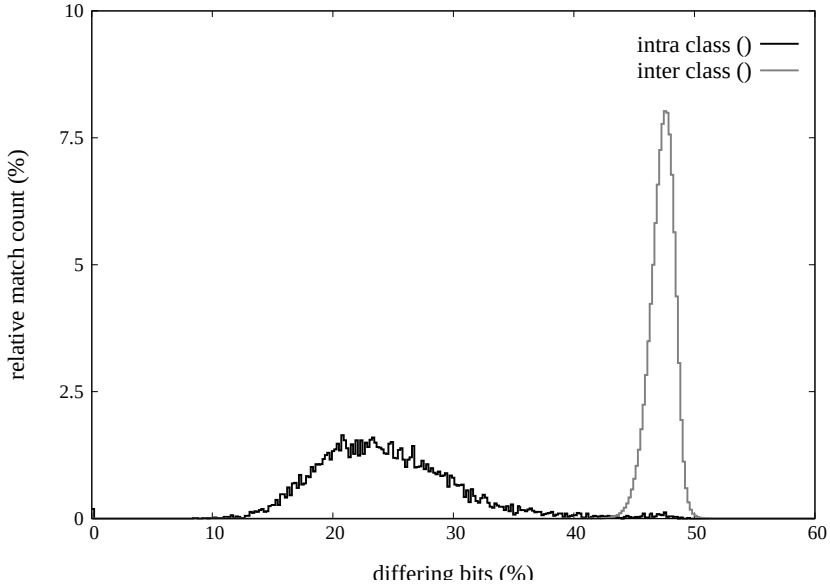

Figure 4.24: The distribution of the intra-class distance and the inter-class distance of the proposed algorithm which are slightly worse than those proposed. For evaluation the entire CASIA iris database was used.

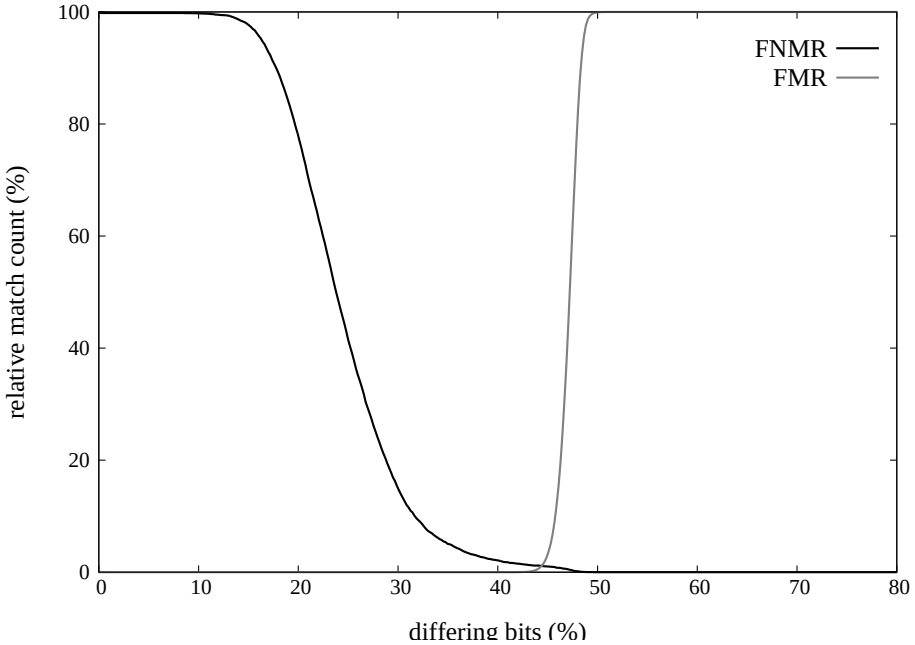

Figure 4.25: The FMR and the FNMR of the implementation using the above algorithm. For evaluation the entire CASIA iris database was used.

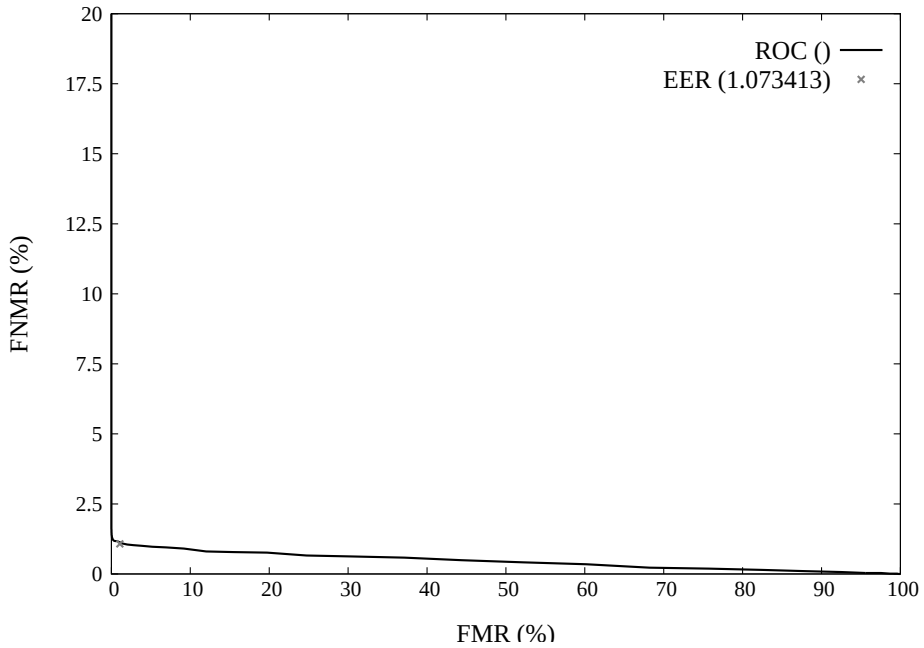

Figure 4.26: The ROC and the EER of the implementation using the above algorithm. For evaluation the entire CASIA iris database was used.

4.8 A Fuzzy Commitment Scheme using Concatenated Error Correction Codes

The algorithm of L. Ma, T. Tan, Y. Wang and D. Zhang [45] seems to fit well into a fuzzy commitment scheme using a concatenation of block level and bit level error correction codes. In Figure 4.24 the distribution of the intra class and inter class distances of the proposed algorithm are shown. These distances correspond to the Hamming distance. This means that single bit errors are counted. While the intra class distance for single bit errors ranges to approximately 48%, this value drastically increases if bit blocks are compared to each other. In Table 4.2 the intra class distances for the according block sizes are summarized. The results for these distances show that a one-level error correction of block level error correction codes does not suffice to correct the occurring errors. Furthermore, the cumulations of the intra class and the inter class distances interfuse with increasing block size.

This means another way of error correction is required. Bit level error correction codes operate on single bit blocks. Unfortunately, these codes are not capable of correcting that many errors. Additionally, these bit errors are randomly distributed and clusters of errors cannot be decoded using bit level error correction codes.

Therefore a combination of both, block level and bit level error correction code, must be used to handle all occurring errors. In the following subchapter the concatenation of these two types of codes is described generally and in detail as it is used in a fuzzy commitment scheme.

4.8.1 Concatenation of Error Correction Codes

Using an iris recognition algorithm for which Hamming distances up to approximately 42% are allowed, a concatenation of error correction codes is meaningful. While one error correction code can be used to correct bit errors, the other code can be used to correct block level errors. Thus a concatenation of these two types of codes is used to encode and decode a cryptographic key in a fuzzy commitment scheme.

While bit level error correction codes work on single bit blocks and do not employ information besides this bit block, block level error correction codes rely on the information provided by a set of bit blocks. Therefore it is suggested to first encode the key using a block level error correction code and subsequently encode the result using a bit level block level error correction code, just like in the approach of Hoa *et al.* [31]. Thereby in the decoding process first the bit level decoding is applied which means that the block level error correction code is able to make use of already decoded information. In Figure 4.27 the basic working flow of a concatenated error correction code is illustrated.

Another aspect which is very important is that it is necessary for both error correction codes to operate on the same blocks. This means that the bit level error correction code should encode and decode bitblocks which form the values used in the block level error correction code, just like it is shown in Figure 4.27. If the error correction code operates on different blocks, one defective block in the bit level could cause several errors in the block level. This phenomenon is illustrated in Figure 4.28.

Hence it is suggested that if a block level error correction code operates on bit blocks of length n, the bit level error correction code should consist of at least 2^n codewords. This guarantees that in the encoding process the bit level error correction code is capable of

Max. Intra-Class Distance (%)	Bit Block Size
48.3	1
53.3	2
67.2	4
79.5	6
85.1	8
90.2	10

Table 4.2: The maximum of the intra class distances of the iris codes, using the algorithm of Tan *et al.* [45], to the according size of the bit block which are compared to each other.

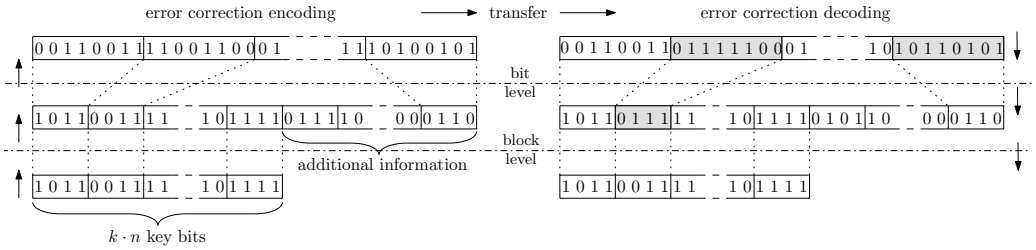

Figure 4.27: The common encoding and decoding flow of a concatenated error correction using bit level and block level error correction codes.

mapping bit blocks of length n to codewords of an arbitrary length m ,where $m > n$, and vice versa in the decoding process.

In the proposed fuzzy commitment scheme the applied iris recognition of Tan *et al.* [45] generates an iris code which consists of 2560 bytes where the second half of the iris code is a bit mask. Thus only the first 1280 bytes are used. Now the parameters of the concatenated error correction codes have to be calculated. This can be done easily by calculating reversely: the result of the second error correction encoding step, namely the bit level error correction, should produce a total number of 1280 bytes as output, denoted by H_{res}, which is XORed with an iriscode afterwards. Applying Hadamard codes as bit level error correction codes implies that codewords of length n are mapped to codewords of length 2^{n-1} (Note: an Hadamard matrix H_k generates an Hadamard code consisting of $2k$ codewords of length k). This means, if codewords of length n are used, an Hadamard code has to be generated out of a Hadamard matrix H of dimension $n-1$ (Hadamard matrices of higher dimension could be used as well). As block level error correction Reed Solomon codes are applied. The length of the result of the block level error correction encoding, denoted by RS_{res}, can be defined as $(1280 \cdot 8)/2^{n-1}$. For a cryptographic key K of length $l \cdot n$ the block level error correction is defined by $RS(l \cdot n, (1280 \cdot 8)/2^{n-1})$. Thus l n-bit information blocks are encoded using $(1280 \cdot 8)/2^{n-1}$ additional bits. In summary, the magnitudes of all bitstreams can be defined as,

$$|K| = l \cdot n, \ |RS_{res}| = \frac{1280 \cdot 8}{2^{n-1}}, \ |H_{res}| = 1280 \cdot 8. \tag{4.38}$$

To apply a Reed Solomon it is required that $1280 \cdot 8 < 2^{2n-1}$ because the Reed Solomon code uses code words of length n. Otherwise several Reed Solomon codes would have to be used for different parts of the bitstream. In Figure 4.29 the whole encoding procedure is summarized.

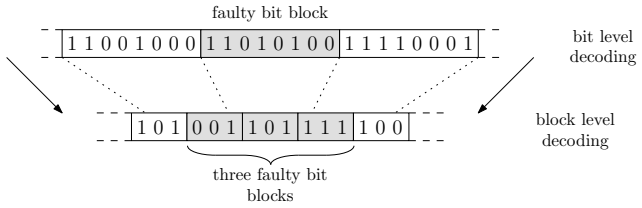

Figure 4.28: If the concatenated error correction codes do not operate on the same blocks, one error in the bit level may cause several errors in the block level.

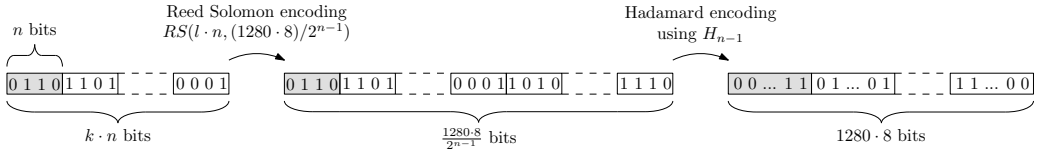

Figure 4.29: The encoding procedure: Reed Solomon codes are used for block level error correction, Hadamard codes are used for bit level error correction.

At this point it becomes clear that one type of error correction code would not suffice. For instance, if Hadamard codes would be applied as bit level error correction codes, large burst error could not be corrected. In detail, using only Hadamard codes a total number of 10 bit blocks, consisting of 11 bits, could be mapped to $10 \cdot 10^2 = 1280 \cdot 8$ bits to bind a 110-bit cryptographic key. However, in the best case only $1024/4 - 1 = 255$ errors could be corrected for each of the 1024-bit blocks. Thus, at most 24.9% of all errors could be corrected which would result in an unsatisfying FMR of 42.15% and an FNMR of 0% according to Figure 4.25.

On the other hand applying only block level error correction codes would not suffice either. For instance, by using Reed-Solomon codes bit block of length 10 would have to be used, because $10 \cdot 2^{10} = 1280 \cdot 8$. Thus, binding a 128 bit cryptographic key, a total number of $(1024 - 128)/2 = 506$ bit blocks ($= 49.4\%$) could be corrected. However, if a single bit error occurs within a 10-bit block the whole block is faulty. According to Table 4.2, by using bit blocks of length 10 the maximal intra-class distance raises beyond 90%. Besides that, using bit blocks of length 10, the distribution of the intra-class and the inter-class distances would overlap more than for single bits. Therefore by using the above algorithm it is not adequate to use block level error correction codes as the only layer of error correction.

4.8.2 Preprocessing

In the preprocessing step the algorithm of Tan *et al.* [45] is applied to generate an iris code which consist of 2560 bytes. The second half of the iris code is dropped. This is because the second 1280 bytes define a bit mask to detect distortions produced by eye lashes. Furthermore, very fast changes in the binary signal are ignored. This means binary sequences of 0s and 1s, which are shorter than a defined threshold, are not used. Instead, these short sequences of 0s are replaced by 1s and vice versa.

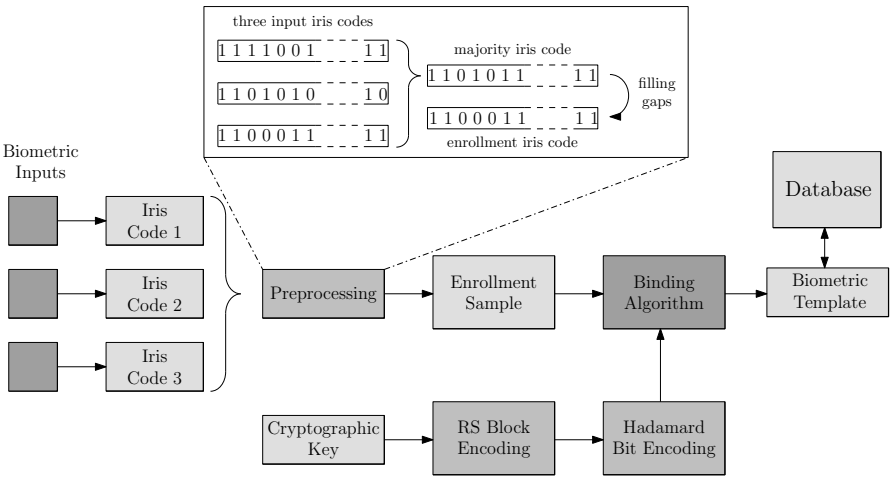

Figure 4.30: The enrollment procedure: the enrollment iris code is bound with an error correction encoded cryptographic key to form a secure biometric template. The generation of the enrollment iris code: the majority of three input samples is calculated and subsequently gaps are filled to produce a natural iris code.

4.8.3 The Enrollment Process

At the time of enrollment a total number of three input images are used to generate the enrollment sample, which is combined with the encoded key to form the biometric template. First the algorithm of Tan *et al.* [45] is applied for each iris image. Subsequently the majority of the three iris codes is calculated. Then the preprocessing step is applied to the majority iris code resulting in a representative iris code for the specified person. The generation of the enrollment iris code is shown in Figure 4.30 as part of the whole enrollment procedure.

At the same time an arbitrary chosen cryptographic key is encoded using a Reed Solomon block level code and a Hadamard bit level code. Here codewords of length $n = 8$ are used to encode an 128 bit key using a $RS(16, 80)$ block level code. These 80 8-bit blocks are then mapped to 80 128-bit blocks so that,

$$80 = \frac{1280 \cdot 8}{2^{n-1}},\ n = 8. \tag{4.39}$$

The resulting bitstream of length $1280 \cdot 8$ bit is bound with the enrollment iris code of the same length by simply XORing these bitstreams. The resulting $1280 \cdot 8$ bit, which represent a secure biometric template in which the cryptographic key is hidden, are finally stored in a database. Here a hash value of the cryptographic key could be stored additionally to verify generated keys in the authentication process (this is suggested in a basic fuzzy commitment scheme but in general a successful Reed Solomon decoding implies a successful authentication). The whole enrollment process is illustrated in Figure 4.30.

4.8.4 The Authentication Process

In the authentication process a single input image is used to generate an iris code which is then reprocessed. The result is then XORed with the biometric template and subsequently error correction decoding is performed. Using an Hadamard code, which is generated out of an Hadamard matrix H_{128} provides 2^8 codewords and is capable of correcting up to $128/4-1$

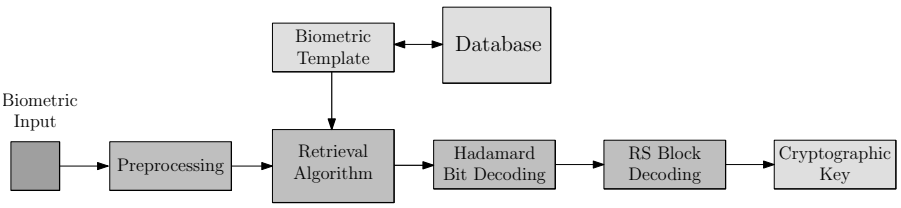

Figure 4.31: The authentication procedure: a single biometric input is processed, XORed with the template and in the end error correction decoding is performed.

bit errors. In the second level of error correction, the $RS(16, 80)$ Reed Solomon code corrects up to $(80 - 16)/2$ 8-bit block level errors. This suffices for the above intra class distances. In Figure 4.31 the authentication process is shown.

4.8.5 Experimental Results

For the evaluation of the fuzzy commitment scheme the CASIA iris database [1] was used. A total number of about 100 persons were used, namely the persons for who at least 8 image were available. The first three images were used for the enrollment procedure and the other five images were used for authentication. In Figure 4.32 the distribution of intra-class and inter-class distances with respect to the number of block level errors is shown. This distribution seems to be quite similar to that of the originally scheme but obviously the FNMR increases since some of the legitimate users are not accepted due to the one-sided occurrence of block level errors or bit level errors. In Figure 4.33 the FNMR and the FMR of the system are plotted and Figure 4.34 shows the ROC resulting in an EER of 1.85%. For a FMR of 0% a FNMR of 4.642% was achieved.

4.8.6 Conclusion

The proposed fuzzy commitment scheme is an example of a biometric cryptosystem in which a concatenation of bit level and block level error correction codes are applied. An 128-bit cryptographic key is hidden in a biometric template so that neither information about the user's biometrics nor information about the cryptographic key is revealed. The applied iris recognition algorithm generates a pretty long iris code which offers the opportunity of using a Hadamard code generated by a Hadamard matrix of high dimension which is capable of correcting a great number of bit level errors. Subsequently a Reed Solomon code is used to correct remaining block level errors. These two error correction codes operate on bit blocks of the same size.

Compared with the results of the original algorithm the FNMR increases. Nevertheless the fuzzy commitment scheme is capable of hiding and retrieving a cryptographic key in and out of a biometric template. This key is 128 bit long (sufficiently long to be used in a common cryptographic system). Additionally, the biometric template is stored in a secure way.

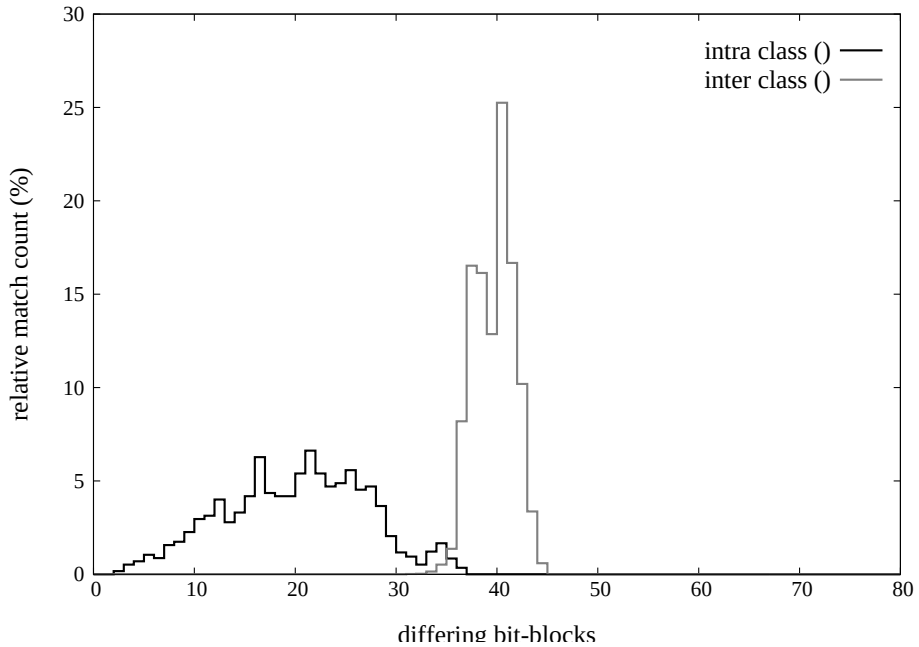

Figure 4.32: The inter-class and intra-class distances of the proposed fuzzy commitment scheme with regard to the total number of block level errors remaining after Hadamard decoding. For evaluation about 100 persons of the CASIA iris database was used.

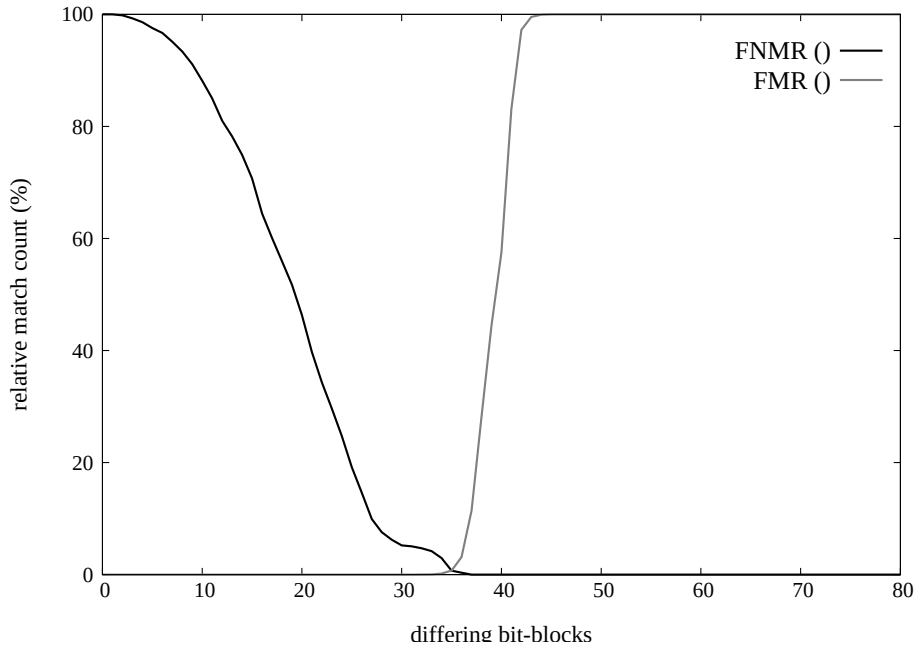

Figure 4.33: The FMR and the FNMR of the fuzzy commitment scheme using the above iris recognition algorithm. For evaluation about 100 persons of the CASIA iris database was used.

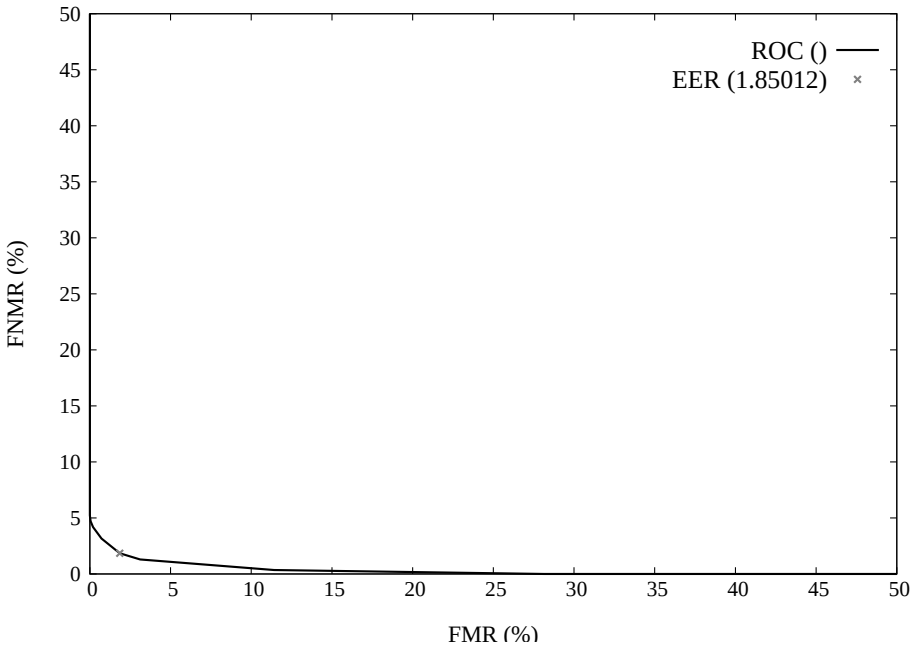

Figure 4.34: The ROC and the EER of the fuzzy commitment scheme using the above iris recognition algorithm. For evaluation about 100 persons of the CASIA iris database was used.

Chapter 5

Implementation of an Iris based Key Generation Scheme

In this chapter an implementation of an iris based key generation scheme is proposed which generates a arbitrary chosen cryptographic key out of iris images. The proposed scheme makes use of an iris recognition algorithm which is based on texture analysis using wavelet transform. The scheme uses defined intervals out of which a cryptographic key is generated. First of all the preconditions for a basic interval scheme are defined (Section 5.1). Then the basic working flow of interval schemes is explained (Section 5.2). Subsequently the applied iris recognition algorithm is summarized and experimental results are presented (Section 5.3). The construction of the proposed implementation is outlined (Section 5.4) and the security of the whole system is analyzed (Section 5.5) as well as the cancellability (Section 5.6). Finally experimental results are presented and discussed (Section 5.7, 5.8).

5.1 Preconditions for Schemes which use Intervals

First the preconditions for the construction of an iris based interval scheme must be defined. Most iris recognition algorithms are geared to the approach of John G. Daugman with respect to the output of the algorithm. Common iris recognition algorithms generate some sort of iris code. These iris codes are then matched using metrics such as the lower hamming distance or similar. By defining a boundary for the inner class distance and the inter class distance the algorithm decides whether to accept or reject a given sample. Such algorithms operate on the bit level. To create biometric cryptosystems using such an algorithm cryptographic keys are hidden in the template like in a classic key binding scheme. This could be done using a fuzzy commitment or a fuzzy vault scheme. Algorithms which operate on the bit level are not suitable for the construction of an interval scheme since the feature vector consists of single bits. There is no sense in defining intervals for single bits. If several bits are combined to form feature values, minimal bit errors change the values drastically as shown in Figure 5.1 where a decimal feature value is calculated out of some feature bits. Of course, using a gray code or simply counting the number of 1s and 0s would in most cases decrease the difference between calculated feature values. However, applying such codes is only practical if a block of bits define a feature in the iris code which is not the case in

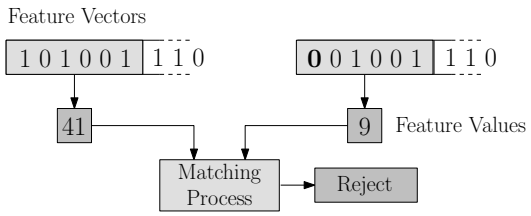

Figure 5.1: Classic feature vectors which consist of single bits. These feature vectors are not suitable for an interval scheme. If several bits are combined to form values minimal errors change the feature values drastically. This means it is not possible to define tight boundaries.

generic iris recognition algorithms. Thus applying such codes to common iris codes won't solve this problem.

This means the construction of an interval scheme requires an iris recognition algorithm, which generates a feature vector consisting out of several feature values (not single bits). Furthermore, these feature values should be very distinctive and should only change slightly from one measurement to another. If this is the case, tight boundaries can be defined for each feature. Additionally it is required that enough of such feature values can be extracted. Every defined interval encodes an extracted feature value with a bitcode. If only a small number of features can be extracted, the resulting cryptographic key will be very short.

In summary the first precondition for the construction of an interval scheme is a suitable iris recognition algorithm. This algorithm should produce a feature vector which consists of feature values (not single bits). Secondly, it is required that a sufficiently large number of feature values can be extracted to produces a cryptographic key which is long enough to be used in classic cryptosystems.

5.2 Preface

If the above preconditions are fulfilled, which means a sufficiently long feature vector consisting of feature values is extracted, intervals can be defined for each feature value. This is done by measuring several biometric inputs at the time of enrollment. Out of the extracted feature values means and deviations are calculated out of which either discrete or adapted intervals are constructed. In the end each interval is encoded, which means a bitcode is assigned to each interval. In the following subchapters the enrollment process and the authentication process of a general interval scheme are described:

5.2.1 Enrollment in an Interval Scheme

As mentioned above, several biometric inputs are necessary during the enrollment procedure. If more biometric inputs are used the boundaries of the calculated intervals will become more precise. Once enough biometric inputs are acquired, feature extraction is applied to each biometric input (for example, an iris image).

In the next step the feature values of all extracted feature vectors are used to set up boundaries for each feature. In general, this can be done by first calculating a representative value which could be the mean, median or a similar value. Then the boundaries around this mean value are set up, which means some deviation is calculated, for example, the mean deviation,

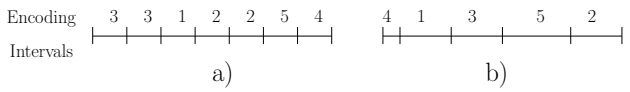

Figure 5.2: Types of intervals: a) discrete intervals where each interval has the same size b) adapted intervals where each interval is encoded with a different codeword

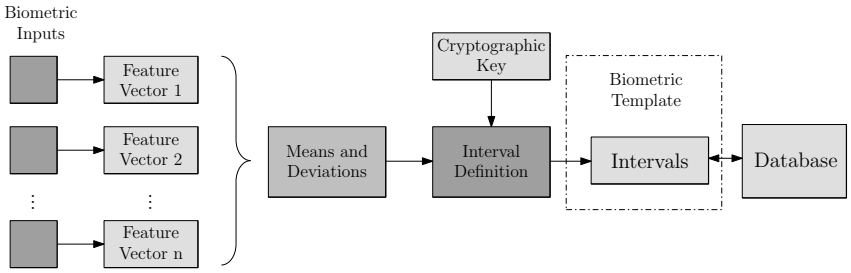

Figure 5.3: The basic enrollment process of an interval scheme: first several samples are used to calculate means and deviations and afterwards intervals are defined which form the biometric template.

standard deviation or a similar deviation.

Now that the interval of each feature is calculated, additional intervals have to be generated to fill up the remaining feature space for each single feature interval. There are two ways to fill up the feature space with intervals. This can be done by dividing the feature space into discrete parts (each part should be considerably smaller than the smallest interval).

The second way to set up intervals in the feature space is to create adapted intervals. This means, the boundaries of all other intervals are created randomly. These fake intervals are calculated just like the chaff points in a fuzzy vault scheme, altough here only one dimension has to be considered. By using this method it suffices to only use one codeword for each interval. In Figure 5.2 these two methods of setting up intervals are illustrated.

Then all the intervals have to be encoded with codewords which together form the cryptographic key. If discrete intervals are used, the intervals which are within the calculated boundaries are encoded with the desired codeword and all other intervals are randomly encoded with other values. This means, some intervals have to be encoded with the same codeword. If adapted intervals are used, it suffices to use one codeword for each interval.

The encoding procedure can be seen as setting up functions which map received feature values to distinct codewords. In the end a hash value of the correct key could be stored as part of the template. This hash should be generated using a secure hash function. The whole enrollment process is illustrated in Figure 5.3.

5.2.2 Authentication in an Interval Scheme

The authentication process is much simpler than the enrollment process. Once a given biometric input is acquired and a feature vector is extracted, the feature values only have to be mapped to codewords. If every feature value is mapped to a codeword, these codewords form the resulting cryptographic key. In the end a hash of the generated key could be tested against a stored hash value of the correct key. The whole process of authentication is illustrated in Figure 5.4.

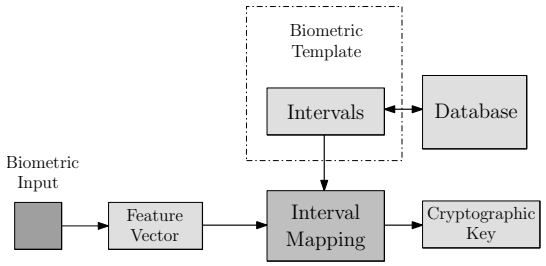

Figure 5.4: The basic authentication process of an interval scheme: first a feature vector is extracted and the feature values are mapped into intervals. The resulting codewords form a cryptographic key.

5.3 Applied Iris Recognition Algorithm

As mentioned above, one precondition for building up an interval scheme is that the iris algorithm which is used for feature extraction should generate a feature vector which consists of a sufficient large number of feature values. The algorithm of Zhu *et al.* [98] fulfills these requirements. The algorithm is based on texture analysis using either Gabor filtering or wavelet transform for feature extraction. By using a 2D wavelet transform a total number of 26 floats are calculated for each biometric input. Therefore each float can be used to calculate a codeword which is at least 5 bits long. This would result in a cryptographic key with a length of at least 130 bits usable in classic cryptographic systems. In the following subchapters the applied iris recognition algorithm is summarized:

5.3.1 Image Acquisition and Preprocessing

For the acquisition of the iris images Zhu et al. have designed an own device [97], which is capable of capturing iris images of sufficiently high quality (this device has been filed for a Chinese Patent). After iris images are captured preprocessing is performed which is composed of three steps. In the first step the iris is localized. The result of this step are two (mostly co-centric) circles which define the inner and outer boundaries of the iris. While the inner boundary is detected by means of thresholding the outer boundary is detected by maximizing changes of the perimeter-normalized sum of gray level values along the circle (this part of the preprocessing is very similar to Daugman's "exploding circles" method). In the next step normalization is performed which means the extracted iris ring is mapped to a rectangle of fixed size. This is done by transforming polar coordinates to cartesian coordinates. Due to the reason that the original images mostly have low contrast and may have non-uniform illumination caused by the position of the light source local histogram equalization is applied. The whole preprocessing operation is illustrated in Figure 5.5.

5.3.2 Feature Extraction

After image acquisition and preprocessing the feature extraction is performed using two different texture analysis methods: multi channel Gabor filtering and wavelet transform:

Gabor filtering is based on a multichannel spatial filtering approach [75]. In this approach a convenient model has been described for the hypothesized visual cortical channels

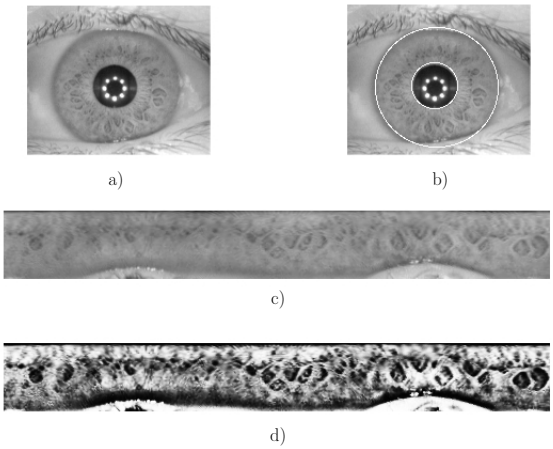

Figure 5.5: Image acquisition and preprocessing: a) in the first step the raw image is acquired b) in the second step the iris is localized c) in the third step the extracted iris ring is normalized d) in the last step the resulting image is enhanced.

for the purpose of texture feature extraction. Each channel is tuned to a specific narrow-band of spatial frequency and orientation and realized by a pair of Gabor filters $h_l(x, y; f, \theta)$ and $h_o(x, y; f, \theta)$. These two Gabor filters are of opposite symmetry and are given by

$$h_l(x, y) = g(x, y) \cdot \cos[2\phi f(x \cos\theta + y \sin\theta)] \tag{5.1}$$

$$h_o(x, y) = g(x, y) \cdot \sin[2\phi f(x \cos\theta + y \sin\theta)] \tag{5.2}$$

where $g(x, y)$ is a 2-D Gaussian function, f and θ are the central frequency and orientation, which define the location of the channel in the frequency plane. It is recommended to use frequencies of power 2. The central frequencies used in their approach are 2, 4, 8, 16, 32 and 64 cycles/degree. Furthermore, for each frequency f, filtering is performed at $\theta = 0^o$,45^o,90^o and 135^o. This leads to a lot of 24 output images (4 for each frequency), from which the iris features are extracted. As features means and standard deviations are calculated for each output image resulting in 48 rotation and translation invariant features. In the performance evaluation either all of the 48 features or subsets of these features are used.

The second texture analysis method used for feature extraction is a 2-D wavelet transform. A 2-D wavelet transform can be treated as two separate 1-D wavelet transforms. At first wavelet transform is applied on original images. Then a set of sub images are obtained at different resolution levels. As in the method using Gabor filtering the means and standard deviations of each wavelet sub-image are extracted as texture features. As basis for the wavelet transform DAUB4 is used. Due to the fact that information at finer resolution level is noisy, only five low resolution levels excluding the coarsest level, are used. The result of this method are 26 features which are robust in noisy environment.

5.3.3 Enrollment and Identification

For the enrollment process it is suggested to use five enrollment samples. Using one of the above methods for all of these five samples the means and standard deviations of all features

Features	Classification Rate
All	93.8
Means	90.6
Standard Deviations	91.9
All at f=2,4,8,16,32	93.8
Wavelet Transform	82.5

Table 5.1: The best proposed results for the identification accuracy of the Gabor filtering using several subsets of the extracted central frequencies and wavelet transform.

are calculated. Thus as result of the enrollment process the means of these feature values are stored in a template. To identify a person the weighted Euclidean distance is used as classifier. Features of an unknown person are compared with those of a set of known iris images. A person is identified as person k if the following weighted Euclidean distance is a minimum at k:

$$WED(k) = \sum_{i=1}^{N} \frac{(f_i - f_i^{(k)})^2}{(\delta_i^{(k)})^2} \tag{5.3}$$

where f_i denotes the ith feature of an unknown person and $f_i^{(k)}$ and $\delta_i^{(k)}$ denote the ith feature and its standard deviation of person k, N is the total number of features extracted from a single iris image. With this classifier an authentication process of one persons involves more than one comparison. A given iris sample has to be compared to every stored template. If a large number of persons is registered this could take a long time. A simple authentication of one person is not possible because it is not possible to set fixed boundaries for the calculated weighted Euclidean distance.

5.3.4 Experimental Results

For the proposed experimental results a total number of 16 different persons were tested [98]. For each person 10 iris images were captured which makes a total of 160 iris images. Using such a small set of iris images the proposed results do not seem to be very meaningful. As mentioned above, for each person 5 samples were used for enrollment and 5 samples were tested. The proposed testing was conducted using different combinations of features. The best proposed identification results are summarized in Table 5.1.

To confirm the presented results an implementation of the proposed algorithm was tested using the CASIA database. All persons for which at least ten iris images were available were tested, which makes a total number of 41 persons. The implementation uses wavelet transform to extract a total number of 26 features.

In Figure 5.6 the FMR and the FNMR are plotted. As it can be seen the results of the implementation are much worse than the proposed ones resulting in a classification rate of 65% which means that for a FMR of 0% the FNMR is 35%. The ROC is shown in Figure 5.7 resulting in an EER of 7.4%. These rates could probably result from the fact that in the results shown in Table 5.1 only 16 different image classes were used. Obviously a subset of 16 persons of the tested could be used to receive better results. Furthermore the method used to capture the iris images could also have an impact on the rates because the presence of eyelids and eyelashes is not handled in the above scheme in the sense of using a bit mask or similar. Therefore the whole test was run through again using only half of the iris image, namely from the right side [45^o to 315^o] and the left side [135^o to 225^o] to

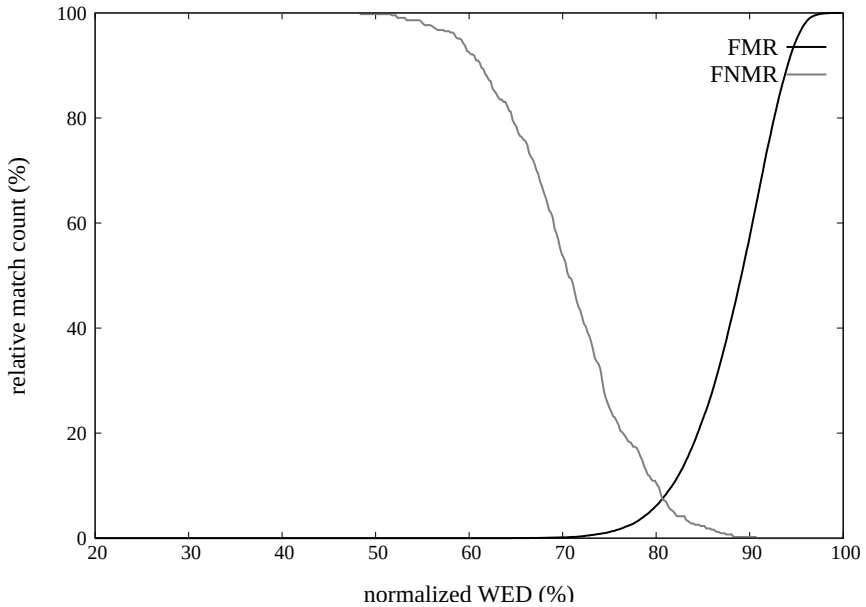

Figure 5.6: The FMR and the FNMR of the implementation using the above algorithm on a total number of 41 persons of the CASIA database. Here the whole iris texture was used for the feature extraction.

get rid of most eyelids and eyelashes. But results which are shown in Figure 5.8 and Figure 5.9 are only slightly better than the aboves. Nevertheless these results should only give a relative comparison to the interval scheme which is using this algorithm. This means the aim of constructing the interval scheme is to retain the above results with the addition of returning a cryptographic key.

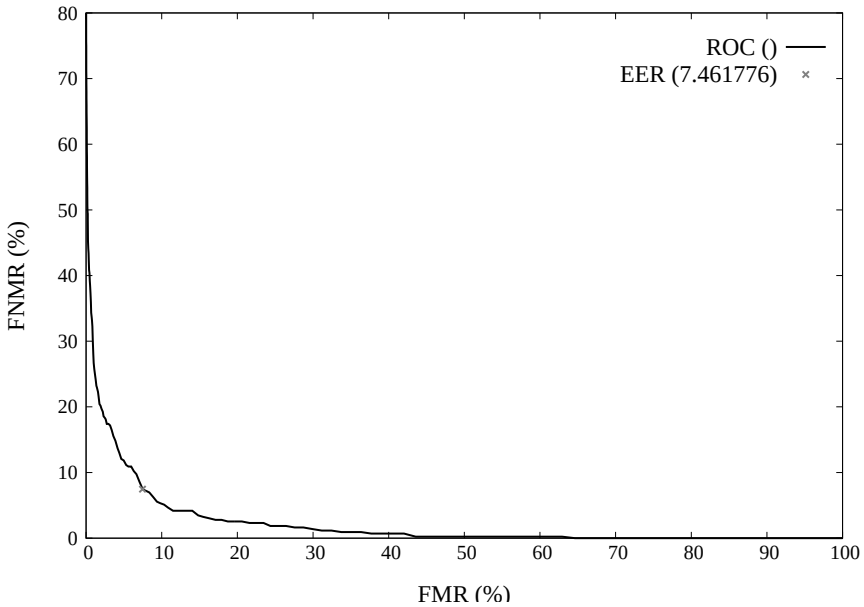

Figure 5.7: The ROC and EER of the implementation using the above algorithm on a total number of 41 persons of the CASIA database. Here the whole iris texture was used for the feature extraction.

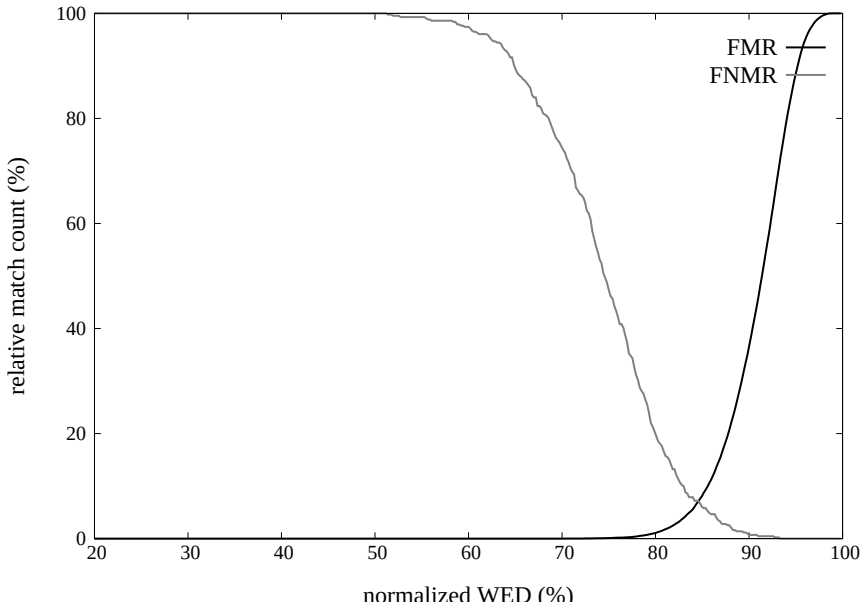

Figure 5.8: The FMR and the FNMR of the implementation using the above algorithm on a total number of 41 persons of the CASIA database. Here only the iris texture from the right side 45^o to 315^o and the left side 135^o to 225^o was used for the feature extraction.

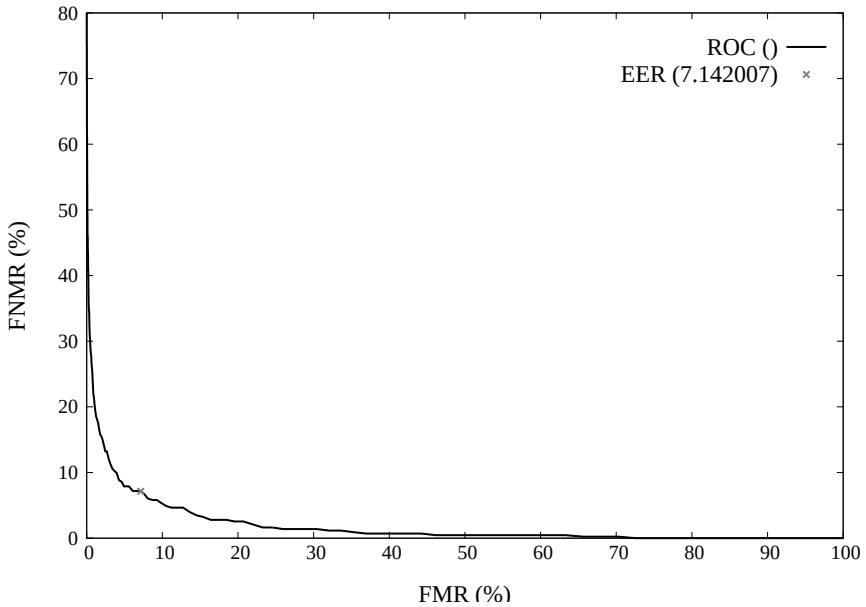

Figure 5.9: The ROC and EER of the implementation using the above algorithm on a total number of 41 persons of the CASIA database. Here only the iris texture from the right side 45^o to 315^o and the left side 135^o to 225^o was used for the feature extraction.

5.4 Construction of the Interval Scheme

Using above iris recognition algorithm with wavelet transform as feature extraction method, 26 features can be used to create a key which has to be at least 128 bit long (so that it can be used in common cryptosystems). This means, the challenge is to construct intervals and extract at least 5 bits out of each interval resulting in a key which is at least 130 bits long. The catchy part of the implementation is the enrollment process in which intervals are set up and encoded. In the authentication process the extracted features only have to be fitted into these intervals.

5.4.1 The Enrollment Process

As mentioned above, the enrollment process is the most challenging part of the system because within the enrollment process the boundaries of the intervals are set up. As recommended in the applied iris recognition algorithm 5 iris samples are used to set up the intervals.

In the first step features are aligned. To align the extracted features the ith feature f_i of each person is divided through a previously approximated mean m_i of this feature resulting in $\hat{f}_i$ so that for every user k

$$\forall i, i \leq N, \hat{f_i^{(k)}} = \frac{f_i^{(k)}}{m_i} \tag{5.4}$$

where N is the number of extracted features (in the applied algorithm N is 26). This is

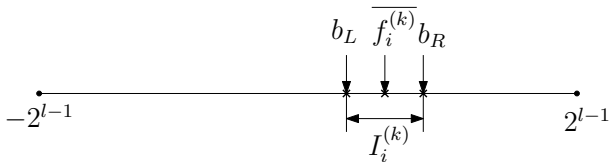

Figure 5.10: The calculation of the initial interval is the result of the first step. In the next step the Interval has to be encoded and furthermore other intervals have to be added.

done because not all the calculated features of one person are in the same range. In the above scheme this problem is solved by dividing the difference of the features through the quadratic standard deviation $(\delta_i^{(k)})^2$ of each feature f_i. The quadratic standard deviation $(\delta_i^{(k)})^2$ of each person k is stored in the template of person k. Since in an interval scheme no other information besides the defined intervals, the codewords and hash of the correct key should be stored for each feature f_i fixed values are used. These approximated values are calculated previously and can also be generated using eyes of non registered users. This method suffices to align the extracted features.

In the next step for each person k the means of all the extracted features are calculated, denoted by $\overline{f_i^{(k)}}$ where

$$\overline{f_i^{(k)}} = \frac{1}{n}\sum_{j=1}^{n} f_{ij}^{(k)} \tag{5.5}$$

and n denotes the number of enrollment samples. Furthermore, for all features the quadratic standard deviation $(\delta_i^{(k)})^2$ is calculated where

$$(\delta_i^{(k)})^2 = \sqrt{\frac{1}{n-1}\sum_{j=1}^{j=n}(f_{ij}^{(k)} - \overline{f_i^{(k)}})^2} \tag{5.6}$$

Once this deviation and the mean of each feature are calculated the boundaries of the initial interval can be set up for every feature of every person as follows:

$$b_L = \overline{f_i^{(k)}} - (\delta_i^{(k)})^2 \bmod 2^{l-1} \tag{5.7}$$

$$b_R = \overline{f_i^{(k)}} + (\delta_i^{(k)})^2 \bmod 2^{l-1} \tag{5.8}$$

where b_L denotes the left border and b_R denotes the right border of the interval and l is the number of bits which should be extracted out of one feature ($l = 5$ leads to a 130 bit key). In Figure 5.10 an example of a result of these first two steps, an initially calculated interval $I_i^{(k)}$ for the ith feature of person k, is illustrated .

In the next step, the encoding of the interval has to be performed. The most primitive way of encoding intervals would be to just assign one codeword to each interval as it is shown in Figure 5.2. A more secure way of encoding the calculated interval would be to define a function, which, if it is evaluated within the boundaries of the interval, returns the right codeword. For example, a Gaussian function could be used for this purpose as proposed by Sutcu *et al.* [74]. The Gaussian function only has to be fitted into the calculated interval. An example of such a Gaussian function which is fitted into a calculated interval is shown in Figure 5.11.

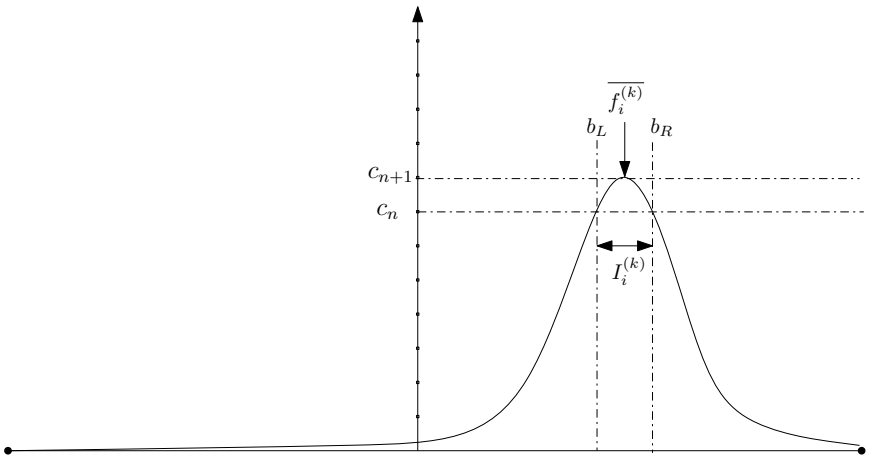

Figure 5.11: A Gaussian function which encodes each feature value which is within the defined interval $I_i^{(k)}$ with the codeword c_n which is part of the cryprographic key.

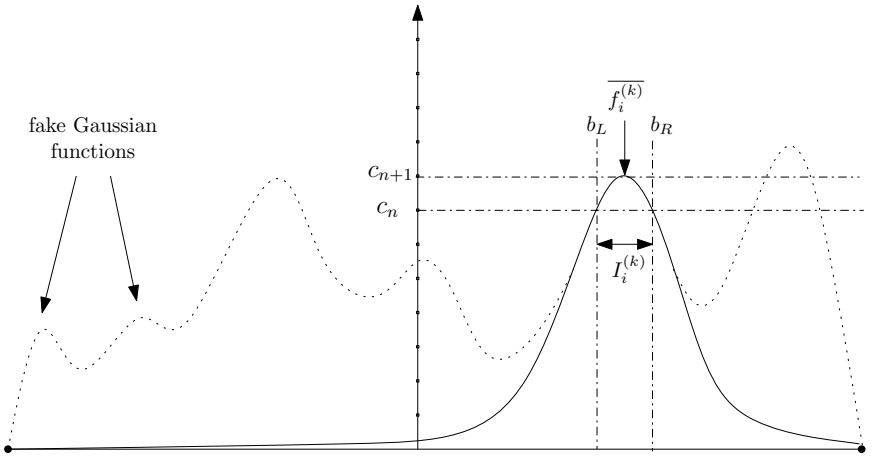

Figure 5.12: The calculation of the initial interval is the result of the first step. In the next step the Interval has to be encoded and furthermore, other intervals have to be added.

Now that the first interval is set up and encoded, other intervals have to be added to hide information. By analogy to chaff points in the fuzzy vault approach here fake intervals are added. As mentioned earlier, global boundaries and means can be approximated for all 26 features. Thus the length of the interval could reveal information. It is advisable to add other intervals of similar length. Once all these so-called "fake intervals" are set up for one feature, these can be encoded randomly. Using functions to encode feature values would mean to add other fake functions over the whole feature space. In Figure 5.12 an example of adding fake Gaussian functions is shown.

If the fake intervals are added properly, an attacker cannot extract much information out of the given functions. Nevertheless, defining Gaussian functions and furthermore, adding fake Gaussian functions does not seem to be a trivial algorithm. Defining a simple polygonal chain would serve the purpose, Figure 5.13 illustrates the analog polygonal chain of a Gaussian function with added fake Gaussian function.

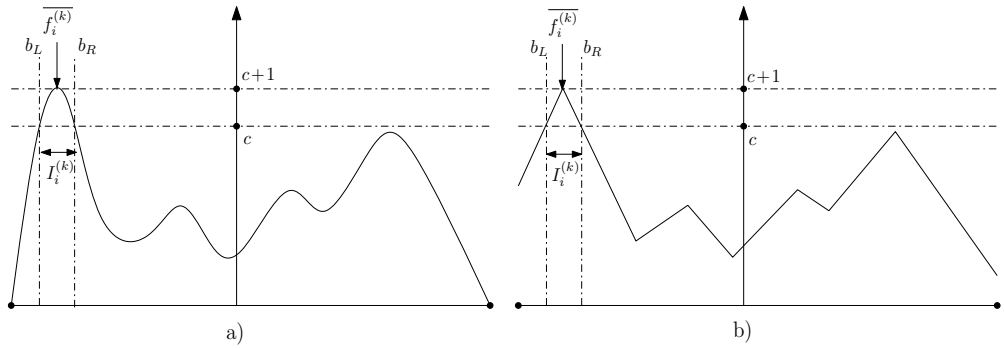

Figure 5.13: a) a Gaussian function with added fake Gaussian functions b) the analog polygonal chain which can be stored as a set of points.

Thus the result of the enrollment procedure would be a polygonal chain that is a set of points (x, y) for each of the 26 features. This scheme is much simpler than one using Gaussian functions. A simple algorithm can be defined for constructing such a polygonal chain:

1. Calculate the initial interval and define a codeword c_n.
2. Construct lines through $(\overline{f_i}, c_{n+1})$ and (b_L, c_n), and through $(\overline{f_i}, c_{n+1})$ and (b_R, c_n).
3. Choose a random point on both lines below $(\overline{f_i}, c_{n+1})$. Then define another two lines which intersect with this point so that the angle of incidence equals the angle of reflexion.
4. Choose a random point (with discrete $y - value$) on both lines.
5. Repeat the procedure for these two points analog to $(\overline{f_i}, c_{n+1})$ in step 2. until enough points are calculated.

Since the features were aligned previously, a fixed number of points can be defined for all features. The whole algorithm for creating the points for one feature is summarized in Figure 5.14.

At the end of the enrollment procedure a hash value $h(K)$ of the correct key K could be stored as part of the template where h is a secure hash function.

5.4.2 The Authentication Process

At the time of authentication one biometric sample, the iris image, is acquired. Again 26 feature values are extracted, which have to be mapped into intervals. Thus each feature value f_i is first divided through a previously approximated mean m_i of this feature resulting in $\hat{f}_i$ (just like in the enrollment phase).

Let $\{P_{i1}, P_{i2}, ..., P_{iN}\}$ be the N points which are stored in the biometric template to form the intervals for the ith feature of a registered person. Assume that $\{P_{i1}, P_{i2}, ..., P_{iN}\}$ is an ordered set with respect to the x-value of each point. Then the two points P_{in} and P_{in+1} of the point set which is stored for the ith feature have to be found so that

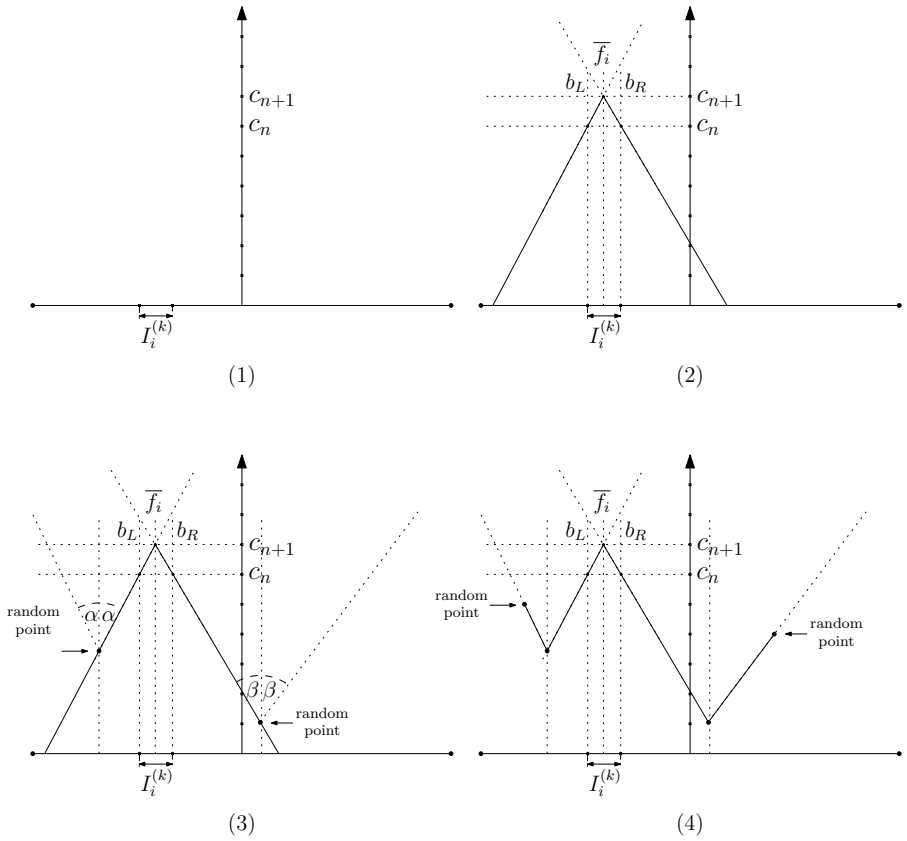

Figure 5.14: The whole algorithm out of which a set of points is generated which defines the simple polygonal chain for one feature

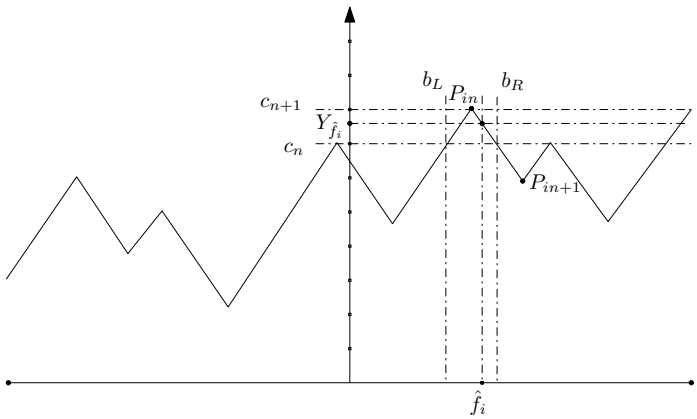

Figure 5.15: The y-value for a feature value f_i is calculated by interpolating two points. The lower absolute value of the result is the codeword for the feature f_i. To avoid that $\lfloor (Y_{\hat{f}_i}) \rfloor = c_{n+1}$ a small δ should be subtracted from the peak.

$$x_{P_{in}} \leq \hat{f}_i \leq x_{P_{in+1}} \tag{5.9}$$

where $x_{P_{in}}$ is the x-value of P_{in} and $x_{P_{in+1}}$ is the x-value of P_{in+1}. If such two points can be found, the corresponding y-value of the extracted feature value is calculated by means of interpolation such that

$$Y_{\hat{f}_i} = Y_{P_{in}} + (\hat{f}_i - X_{P_{in}}) \cdot Y_{\overrightarrow{P_{in}P_{in+1}}} \tag{5.10}$$

where $Y_{P_{in}}$ is the y-value of the point P_{in}, $X_{P_{in}}$ is the x-value of the point P_{in} and $\overrightarrow{P_{in}P_{in+1}}$ is the vector between the points P_{in} and P_{in+1}. If the y- value of the feature value is calculated, the codeword $C_{\hat{f}_i}$ is returned where

$$C_{\hat{f}_i} = \lfloor (Y_{\hat{f}_i}) \rfloor \tag{5.11}$$

This is done for all feature values as described in Figure 5.15 and the resulting key $\hat{K}$ is the concatenation of all returned codewords. To test the key, a hash value $h(K)$ of the correct key K could be stored as part of the template and if $h(\hat{K}) = h(K)$, the user is accepted and otherwise the user is rejected.

5.5 Security Analysis

In most biometric systems the security of the template is very critical. In the proposed scheme the template consists of a set of simple polygonal chains and a hash value of the correct cryptographic key. This hash value is created using a secure hash function. If enough points are involved and these points are chosen intelligently, the simple polygonal chains do not reveal any information about the actual intervals out of which the correct key can be constructed. The security is scalable in the sense of how many fake points are stored for each feature. If more points are stored for each feature, the more difficult it becomes for an attacker to get any information about the correct intervals. One problem resulting from the proposed method is that several feature values could lead to a correct codeword. This weakness is illustrated in Figure 5.16.

The proposed system acts like a simple private template scheme. Out of extracted feature values codewords are generated which form a cryptographic key. But in the encoding step the cryptographic key can be chosen randomly. This means, the key is not directly generated out of a biometric sample but bound into the template in the encoding step.

Longer keys could be generated if a greater set of codewords would be used. But then more fake points must be added which means the feature space has to be enlarged as well.

To make the whole system collision-free depends on the applied feature extraction method. The interval scheme itself does not have any impact on this issue.

5.6 Cancellable Interval Scheme

The encoding function (the polygonal chain) maps the extracted features of one person to a previously generated key. In case a cryptographic key is stolen, a new cryptographic key

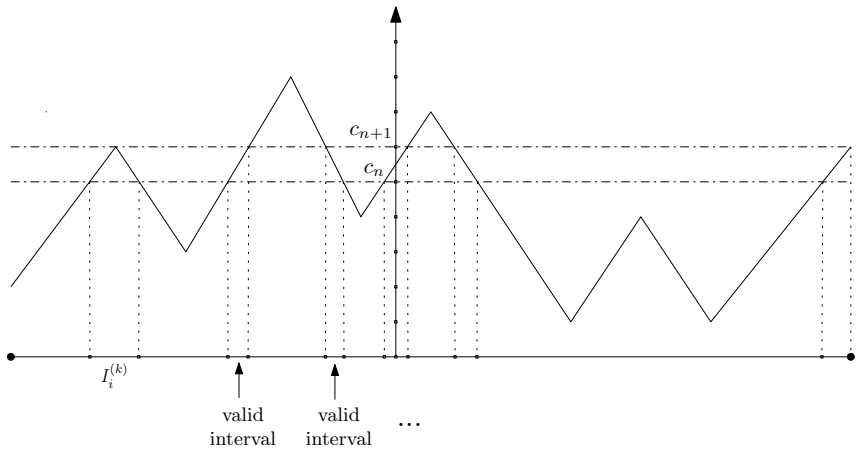

Figure 5.16: For each generated interval several other feature values, which are outside the defined boundaries, could possibly generate the same codeword.

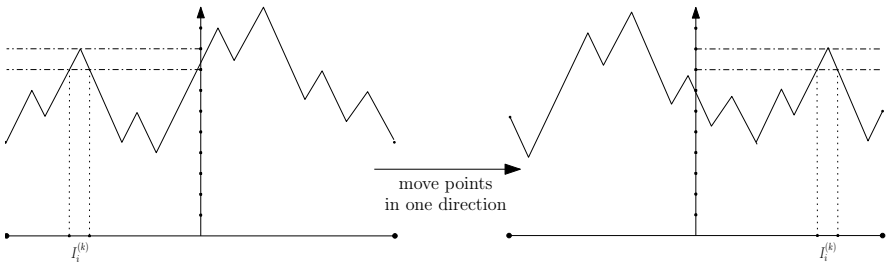

Figure 5.17: For each generated interval several other feature values which are outside the defined boundaries could possibly generate the same codeword.

can be generated and the polygonal chain is adapted. Therefore the scheme is cancellable in case the cryptographic key is guessed, stolen or comprised. Furthermore, different keys can be assigned to one person for different applications.

In case the biometric itself is stolen, the scheme could be extended (note that if an interval scheme like the proposed is used for online signatures one could learn to imitate another user's signature - this would mean that this person is capable to generate a valid key at any time for any application). To avoid this leakage, a physical token could be introduced to increase security. For example, a pin-code could be stored on a smartcard, which for example, moves the boundaries of all intervals in one direction as shown in Figure 5.17. Thus only the biometric input in combination with the right physical token would return a valid key. Nevertheless, if the biometrics itself is compromised the whole system is just as secure as the pin-code.

5.7 Experimental Results

The whole system was tested on the CASIA database. All persons for which 10 images were available were tested, which makes a total number of 41 persons. As expected the achieved results are slightly worse than those of the original algorithm. This is because the

Measurement	Whole images	Slitted images
FNMR	43.78	36.21
FMR	1.87	0.07

Table 5.2: The results for the interval scheme using entire iris images and the result of the interval scheme using iris images from the right side 45^o to 315^o and the left side 135^o to 225^o.

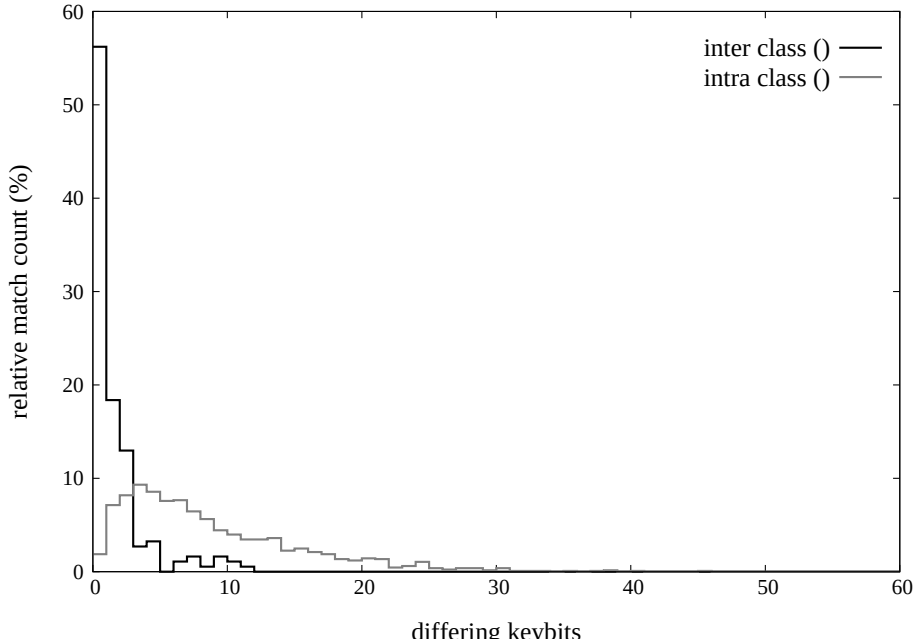

Figure 5.18: The distribution of the extracted 130 bit cryptographic keys with respect to the Hamming distance to the correct cryptographic key. Here the entire iris images were used for feature extraction.

original algorithm aims at a classification of an iris image, which means a minimum value of the weighted Euclidean distance has to be found (notice that a big WED value can still represent a minimum). Thus in the original scheme no boundaries are set, while in the interval scheme boundaries have to be set in which the extracted features are fitted into.

The experimental results for the proposed interval scheme are illustrated in Figure 5.18 (only the distance to the original key is plotted - a ROC of FMR/FNMR plot makes no sense in this case, because only keys which match the original key suffice). Here the whole iris image is used for the feature extraction. Both the FMR and the FNMR become worse. The results for the FMR and the FNMR are summarized in Table 5.2. Thus in a second experiment all iris images were slitted from the right side 45^o to 315^o and the left side 135^o to 225^o. The rest of the iris images which does not contain as much distortion (eyelids and eyelashes) is used for feature extraction. The results are shown in Figure 5.19. Using images with less distortion thightens the boundaries of the intervals and thus the FNMR and the FMR are improved. In the original algorithm slitting the image does not have much impact on the results, because in most cases the former minimum remains the minimum. But in the interval scheme, which uses this algorithm the performance is increased greatly resulting in a success rate of almost 63.5%. This means that the proposed interval scheme does not decrease the performance of the iris recognition scheme drastically.

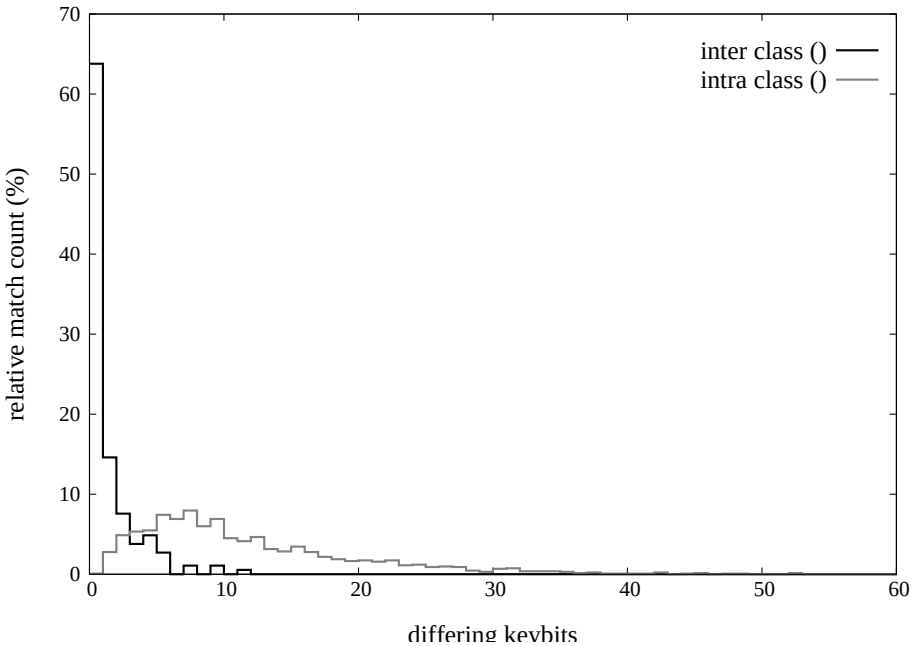

Figure 5.19: The distribution of the extracted 130 bit cryptographic keys with respect to the Hamming distance to the correct cryptographic key. Here the entire iris images from the right side 45^o to 315^o and the left side 135^o to 225^o were used for feature extraction.

5.8 Conclusion

The proposed interval scheme uses an algorithm used for the classification of iris images. As features this algorithm extracts float values which are used together with deviations to build up an interval scheme. The performance of the interval scheme is only slightly worse (about 1.5%) than the performance of the original scheme. Additionally, an arbitrary 130 bit cryptographic key is hidden in the biometric template during the enrollment process and is retrieved during the authentication process. The hiding of the cryptographic key is done by creating simple polygonal chains which provide high security (which is scalable as well).

In summary, a biometric cryptosystem was presented which does not drastically decrease the performance of the algorithm used for feature extraction. Still, this biometric cryptosystem hides and retrieves a cryptographic key into and out of the biometric template which is sufficiently long to be used in a classic cryptographic system. Additionally, high security is achieved through simple means.

Bibliography

[1] The Center of Biometrics and Security Research, CASIA Iris Image Database, http://www.sinobiometrics.com.

[2] A. Adler, "Vulnerabilities in Biometric Encryption Systems," *Audio- and video-based Biometric Person Authentication (AVBPA)*, pp. 1100–1109, 2005.

[3] S. Agaian, *Hadamard Matrix and Their Applications*, ser. Lect. notes in math. Springer Verlag, 1985, vol. 1168.

[4] L. Ballard, S. Kamara, F. Monrose, and M. Reiter, "On the requirements of biometric key generators," *Technical Report TR-JHU-SPAR-BKMR-090707*, 2007, submitted and available as JHU Department of Computer Science Technical Report.

[5] E. Berlekamp, "Factoring Polynomials Over Finite Fields," *Bell Systems Technical Journal*, vol. 46, pp. 1853–1859, 1967.

[6] A. Bodo, "Method for producing a digital signature with aid of a biometric feature," 1994, german patent DE 42 43 908 A1.

[7] K. Bowyer, K. Hollingsworth, and P. Flynn, "Image understanding for iris biometrics: a survey," *Computer Vision and Image Understanding*, no. 110, pp. 281–307, 2008.

[8] X. Boyen, "Reusable cryptographic fuzzy extractors," *CCS 2004 Proceedings of the 11th ACM Conference on Computer and Communications Security*, pp. 82–91, 2004.

[9] A. Burnett, F. Byrne, T. Dowling, and A. Duffy, "A Biometric Identity Based Signature Scheme," *Applied Cryptography and Network Security Conference, New York*, 2005.

[10] J. P. Campbell, "Speaker Recognition: A Tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[11] R. Canetti, "Towards realizing random oracles: hash function which hide all partial information," *Advances in Cryptology. proc. of Crypto'97 (LNCS: 1294)*, pp. 455–469, 1997.

[12] A. Cavoukian, A. Stoianov, and F. Carter, "Biometric Encryption: Technology for Strong Authentication, Security and Privacy," *IFIP International Federation for Information Processing*, vol. 261/2008, pp. 57–77, 2008.

[13] Y.-J. Chang, W. Zhang, and T. Chen, "Biometrics-based cryptographic key generation," *Proceedings of 2004 IEEE International Conference on Multimedia and Expo (ICME-2004)*, vol. 3, pp. 2203–2206, 2004.

[14] R. Chellappa, C. Wilson, and S. Sirohey, "Human and Machine Recognition of Faces: A Survey." *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–740, 1995.

[15] B. C. Chen and V. Chandran, "Biometric based cryptographic key generation from faces," *In Proceedings Digital Image Computing: Techniques and Applications (DICTA)*, pp. 394–401, 2007.

[16] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcard-based fingerprint authentication," *Proc. ACM SIGMM 2003 Multimedia, Biometrics Methods and Applications Workshop*, pp. 45–52, 2003.

[17] T. Connie, A. Teoh, M. Goh, and D. Ngo, "Palmhashing: a novel approach for cancelable biometrics," *Inf. Process. Lett.*, vol. 93, no. 1, pp. 1–5, 2005.

[18] J. Daugman, "The importance of being random: Statistical principles of iris recognition," *Pattern Recognition*, vol. 36, no. 2, pp. 279–291, 2003.

[19] ——, "How Iris Recognition Works," *IEEE Trans. CSVT*, vol. 14, no. 1, pp. 21–30, 2004.

[20] G. Davida, Y. Frankel, and B. Matt, "On enabling secure applications through off-line biometric identification," *Proc. of IEEE, Symp. on Security and Privacy*, pp. 148–157, 1998.

[21] ——, "On the relation of error correction and cryptography to an off line biometric based identication scheme," *Proc. of WCC99, Workshop on Coding and Cryptography*, pp. 129–138, 1999.

[22] D. L. Delivasilis and S. K. Katsikas, "Side channel analysis on biometric-based key generation algorithms on resource constrained devices," *International Journal of Network Security*, vol. 3, no. 1, pp. 44–50, 2005.

[23] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," *Proc. Eurocrypt 2004 (LNCS: 3027)*, pp. 523–540, 2004.

[24] S. Draper, A. Khisti, E. Martinian, A. Vetro, and J. Yedidia, "Secure storage of fingerprint biometrics using slepian-wolf codes," *in Inform. Theory and Apps. Work. (UCSD)*, 2007.

[25] S. Y. E. Martinian and J. S. Yedidia, "Secure biometrics via syndromes," *in 43rd Annual Allerton Conference on Communications, Control, and Computing, Monticello, IL, USA*, 2005.

[26] H. Feng and C. C. Wah, "Private key generation from on-line handwritten signatures," *Information Management and Computer Security*, vol. 10, no. 18, pp. 159–164, 2002.

[27] M. R. Freire, J. Fierrez, J. Galbally, and J. Ortega-Garcia, "Biometric Hashing Based on Genetic Selection and Its Application to On-Line Signatures," *International Conference on Biometrics '07 (LNCS: 4642)*, pp. 1134–1143, 2007.

[28] A. Goh and D. C. L. Ngo, "Computation of cryptographic keys from face biometrics," in *Communications and Multimedia Security (LNCS: 2828)*, 2003, pp. 1–13.

[29] A. Goh, A. B. J. Teoh, and D. C. L. Ngo, "Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1892–1901, 2006.

[30] A. Graps, "An introduction to wavelets," *IEEE Computational Science and Engineering*, vol. 2, no. 2, 1995.

[31] F. Hao, R. Anderson, and J. Daugman, "Combining Cryptography with Biometrics Effectively," *IEEE Transactions on Computers*, vol. 55, no. 9, pp. 1081–1088, 2006.

[32] F. Hoa and C. W. Chan, "Private key generation from on-line handwritten signatures," *Information Management and Computer Security*, vol. 10, no. 2, pp. 159–164, 2002.

[33] G. S. I. Reed, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, pp. 300–304, 1960.

[34] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based Fingerprint matching," *In Proc. of IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, 2000.

[35] A. Juels and M. Sudan, "A fuzzy vault scheme," *Proc. 2002 IEEE International Symp. on Information Theory*, p. 408, 2002.

[36] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," *Sixth ACM Conference on Computer and Communications Security*, pp. 28–36, 1999.

[37] S. Kamara, B. de Medeiros, and S. Wetzel, "Secret locking: Exploring new approaches to biometric key encapsulation," *Proceedings of ICETE*, pp. 254–261, 2005.

[38] J.-G. Ko, Y.-H. Gil, and J.-H. Yoo, "Iris Recognition using Cumulative SUM based Change Analysis," *Intelligent Signal Processing and Communications, 2006. ISPACS '06*, pp. 275–278, 2006.

[39] A. Kong, K.-H. Cheunga, D. Zhanga, M. Kamelb, and J. Youa, "An analysis of BioHashing and its variants," *Pattern Recognition*, vol. 39, pp. 1359–1368, 2006.

[40] Y. W. Kuan, A. B. J. Teoh, and D. C. L. Ngo, "Secure hashing of dynamic hand signatures using wavelet-fourier compression with biophasor mixing and 2^N discretization," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, pp. 32–32, 2007.

[41] C. Lee, J. Choi, K. Toh, S. Lee, and J. Kim, "Alignment-free cancelable fingerprint templates based on local minutiae information," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 37, no. 4, pp. 980–992, 2007.

[42] Q. Li, M. Guo, and E.-C. Chang, "Fuzzy extractors for asymmetric biometric representations," *IEEE Workshop on Biometrics (In association with CVPR)*, pp. 1–6, 2008.

[43] Q. Li, Y. Sutcu, and N. Memon, "Secure sketch for biometric templates," *Advances in Cryptology - ASIACRYPT 2006 (LNCS:4284)*, pp. 99–113, 2006.

[44] A. Lumini and L. Nanni, "An improved BioHashing for human authentication," *Pattern Recognition*, vol. 40, no. 3, pp. 1057–1065, 2006.

[45] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Efficient Iris Recogntion by Characterizing Key Local Variations," *IEEE Transactions on Image Processing*, vol. 13, no. 6, pp. 739–750, 2004.

[46] E. Maiorana and C. Ercole, "Secure Biometric Authentication System Architecture using Error Correcting Codes and Distributed Cryptography," *Gruppo nazionale Telecomunicazioni e Teoria dell'Informazione (GTTI'07)*, 2007.

[47] M. Malek, "Hadamard Codes," http://www.mcs.csueastbay.edu/malek/Class/Hadamard.pdf.

[48] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition.* Springer-Verlag, 2003.

[49] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel, "Cryptographic Key Generation from Voice," *SP '01: Proceedings of the 2001 IEEE Symposium on Security and Privacy*, 2001, 12 pages.

[50] ——, "Using Voice to Generate Cryptographic Keys," *Proc. 2001: A Speaker Odyssey, The Speech Recognition Workshop*, 2001, 6 pages.

[51] F. Monrose, M. K. Reiter, D. P. Lopresti, and C. Shih, "Toward speech-generated cryptographic keys on resource constrained devices," *Proc. 11th USENIX Security Symp.*, pp. 283–296, 2002.

[52] F. Monrose, M. K. Reiter, and S. Wetzel, "Password hardening based on keystroke dynamics," *Proceedings of sixth ACM Conference on Computer and Communications Security, CCCS*, pp. 73–82, 1999.

[53] A. Nagar and S. Chaudhury, "Biometrics based Asymmetric Cryptosystem Design Using Modified Fuzzy Vault Scheme," *18th International Conference on Pattern Recognition (ICPR'06)*, vol. ICPR (4), pp. 537–540, 2006.

[54] V. S. Nalwa, "Automatic on-line signature verification," *Proceedings of the IEEE*, vol. 85, pp. 215–239, 1997.

[55] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based Fuzzy Vault: Implementation and Performance," *in IEEE Transactions on Information Forensics And Security*, vol. 2, pp. 744–757, 2007.

[56] T. Olzak, "Keystroke Dynamics: Low Impact Biometric Verification," 2006, http://www.infosecwriters.com/text_resources/pdf/ Keystroke_TOlzak.pdf.

[57] N. Poh and J. J. Korczak, "Hybrid biometric person authentication using face and voice features," *AVBPA '01: Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication (LNCS: 2091)*, pp. 348–353, 2001.

[58] N. K. Ratha and R. M. Bolle, *Automatic Fingerprint Recognition Systems.* Springer-Verlag, 2003.

[59] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, pp. 614–634, 2001.

[60] N. K. Ratha, J. H. Connell, and S. Chikkerur, "Generating cancelable fingerprint templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 561–572, 2007.

[61] E. Reddy and I. Babu, "Performance of Iris Based Hard Fuzzy Vault," *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, no. 1, pp. 297–304, 2008.

[62] A. Ross, A.K.Jain, and S.Pankati, "A prototype hand-geometry based verification system," *Conference on Audio- and Video-based Biometric Person Authentication*, pp. 166–171, 1999.

[63] V. Roy and C. V. Jawahar, "Feature Selection for Hand-Geometry based Person Authentication," 2005, 7 pages.

[64] M. Savvides, B. Kumar, and P. Khosla, "Cancelable biometric filters for face recognition," *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04)*, vol. 3, pp. 922–925, 2004.

[65] W. Scheirer and T. Boult, "Cracking Fuzzy Vaults and Biometric Encryption," *in Biometrics Symposium, 2007*, pp. 1–6, 2007.

[66] A. Shamir, "How to share a secret," *Comm ACM 22*, pp. 612–613, 1979.

[67] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, pp. 471–480, 1973.

[68] O. T. Song, A. B. Teoh, and D. C. L. Ngo, "Application-specific key release scheme from biometrics," *International Journal of Network Security*, vol. 6, no. 2, pp. 122–128, 2008.

[69] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar, "Biometric Encryption - Enrollment and Verification Procedures," *Proc. SPIE, Optical Pattern Recognition IX*, vol. 3386, pp. 24–35, 1998.

[70] ——, "Biometric Encryption using image processing," *Proc. SPIE, Optical Security and Counterfeit Deterrence Techniques II*, vol. 3314, pp. 178–188, 1998.

[71] ——, "Biometric encryption," *ICSA Guide to Cryptography*, 1999.

[72] ——, "Method for secure key management using a biometrics," 2001, U.S. Patent 6219794.

[73] Y. Sutcu, Q. Li, and N. Memon, "How to Protect Biometric Templates," *SPIE Conf. on Security, Steganography and Watermarking of Multimedia Contents IX*, vol. 6505, 2007, Proceedings of SPIE, 11 pages.

[74] Y. Sutcu, H. T. Sencar, and N. Memon, "A secure biometric authentication scheme based on robust hashing," *MMSec '05: Proceedings of the 7th Workshop on Multimedia and Security*, pp. 111–116, 2005.

[75] T. Tan, "Texture Feature Extraction Via Visual Cortical Channel Modelling," *Proc. 11th IAPR Inter. Conf. Pattern Recognition*, vol. III, pp. 607–610, 1992.

[76] C. Taswell, "The what, how, and why of wavelet shrinkage denoising," *Computing in Science & Engineering*, vol. 2, pp. 12–19, 2000.

[77] A. B. J. Teoh, D. C. L. Ngo, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition 37*, pp. 2245–2255, 2004.

[78] ——, "Personalised cryptographic key generation based on FaceHashing," *Computers And Security*, vol. 2004, no. 23, pp. 606–614, 2004.

[79] ——, "Biometric Hash: High-Confidence Face Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 6, pp. 771–775, 2006.

[80] A. Teoh and J. Kim, "Secure biometric template protection in fuzzy commitment scheme," *IEICE Electron. Express*, vol. 4, no. 23, pp. 724–730, 2007.

[81] G. J. Tomko, C. Soutar, and G. J. Schmidt, "Fingerprint controlled public key cryptographic system," 1996, U.S. Patent 5541994.

[82] V. Tong, H. Sibert, J. Lecoeur, and M. Girault, "Biometric fuzzy extractors made practical: a proposal based on fingercodes," *In International Conference on Biometrics (LNCS: 4642)*, 2007.

[83] S. Tulyakov, F. Farooq, and V. Govindaraju, "Symmetric hash functions for fingerprint minutiae," *In: International Workshop on Pattern Recognition for Crime Prevention (LNCS: 3687), Security and Surveillance*, pp. 30–38, 2005.

[84] S. Tulyakov, F. Farooq, P. Mansukhani, and V. Govindaraju, "Symmetric hash functions for secure fingerprint biometric systems," *Pattern Recogn. Lett.*, vol. 28, no. 16, pp. 2427–2436, 2007.

[85] M. Turk and A. Pentland, "Face recognition using eigenfaces," *Computer Vision and Pattern Recognition*, pp. 586–591, 1991.

[86] U. Uludag and A. K. Jain, "Fuzzy fingerprint vault," *in Proc. Workshop: Biometrics: Challenges Arising from Theory to Practice*, pp. 13–16, 2004.

[87] U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain, "Biometric cryptosystems: issues and challenges," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 948–960, 2004.

[88] C. Vielhauer, "Biometric User Authentication for IT Security," *Advances in Information Security*, vol. 18, 2006, iSBN: 0-387-26194-X.

[89] C. Vielhauer and R. Steinmetz, "Handwriting: feature correlation analysis for biometric hashes," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 1, pp. 542–558, 2004.

[90] C. Vielhauer, R. Steinmetz, and A. Mayerhöfer, "Biometric hash based on statistical features of online signatures," in *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1*. Washington, DC, USA: IEEE Computer Society, 2002, p. 10123.

[91] K. Voderhobli, C. Pattinson, and H. Donelan, "A schema for cryptographic key generation using hybrid biometrics," *7th annual postgraduate symp.: The convergence of telecommunications, networking and broadcasting, Liverpool*, 2006.

[92] X. Wu, N. Qi, K. Wang, and D. Zhang, "A Novel Cryptosystem based on Iris Key Generation," *Fourth International Conference on Natural Computation (ICNC'08)*, pp. 53–56, 2008.

[93] S. Yang and I. Verbauwhede, "Automatic secure fingerprint verification system based on fuzzy vault scheme," *in Proceedings of IEEE ICASSP*, vol. 5, pp. 609–6012, 2005.

[94] W. K. Yip, A. B. J. Teoh, and D. C. L. Ngo, "Replaceable and securely hashed keys from online signatures," *IEICE Electron. Express*, vol. 3, no. 18, pp. 410–416, 2006.

[95] W. Zhang, Y.-J. Chang, and T. Chen, "Optimal thresholding for key generation based on biometrics," *Int. Conf. on Image Processing (ICIP'04)*, pp. 3451–3454, 2004.

[96] G. Zheng, W. Li, and C. Zhan, "Cryptographic key generation from biometric data using lattice mapping," *18th International Conference on Pattern Recognition (ICPR 2006)*, vol. 4, pp. 513–516, 2006.

[97] Y. Zhu, T. Tan, and Y. Wang, "Iris Image Acquisition System," *Chinese Patent Application, No.99217063.X*, 1999.

[98] ——, "Biometric personal identification based on iris patterns," *15th International Conference on Pattern Recognition (ICPR'00) - Volume 2*, p. 2801, 2000.