Study of the Privacy and Accuracy of the Fuzzy Commitment Scheme
BioKeyS III-Final Report
Table of contents

1 Introduction ................................................................................................................................. 7
  1.1 Protection of biometric information .......................................................................................... 7
  1.2 The BioKeyS III project ........................................................................................................... 8
  1.3 Overview of the document ....................................................................................................... 9

2 Security and privacy in biometric systems ............................................................................. 10
  2.1 Introduction .......................................................................................................................... 10
  2.2 Security and privacy ............................................................................................................. 11
  2.3 Conclusions ......................................................................................................................... 16

3 The Fuzzy Commitment Scheme (FCS) and other BE methods ............................................. 17
  3.1 ISO SC27 24745 – Biometric information protection ......................................................... 17
  3.2 Overview of BE methods .................................................................................................... 18
  3.3 A practical implementation of the FCS ................................................................................ 20
  3.4 Additional techniques ......................................................................................................... 23
  3.5 Conclusions ......................................................................................................................... 26

4 Study of the privacy of a practical FCS .................................................................................. 27
  4.1 Introduction ........................................................................................................................ 27
  4.2 Privacy requirements .......................................................................................................... 27
  4.3 Adversary capabilities ...................................................................................................... 28
  4.4 A framework for privacy assessment ................................................................................... 28
  4.5 Privacy of the BioHASH® FCS system ............................................................................... 28
  4.6 Conclusions ......................................................................................................................... 34

5 Approaches to fusion of private templates ............................................................................ 35
  5.1 Introduction ........................................................................................................................ 35
  5.2 Implications for the different fusion methods .................................................................... 38
  5.3 Conclusion ........................................................................................................................ 38

6 Assessment of the Classification Performance and Privacy of the BioHASH® SDK .......... 40
  6.1 Introduction ........................................................................................................................ 40
  6.2 Biometric testing ............................................................................................................... 41
  6.3 Results for Projekt BioKeyS-PilotDB-Teil1 ......................................................................... 41
  6.4 Assessment for the MCYT database .................................................................................... 41
  6.5 Assessment for the FingerQS databases .......................................................................... 43
  6.6 Measurable privacy .......................................................................................................... 44
  6.7 Conclusions ......................................................................................................................... 46

7 Assessment of the Classification Performance and Privacy using Two Finger Fusion ....... 47
  7.1 Introduction ........................................................................................................................ 47
  7.2 The key level fusion approach ............................................................................................. 47
  7.3 Generating DET curves for key level fusion ....................................................................... 48
  7.4 Biometric testing and results ............................................................................................. 49
1 Introduction

1.1 Protection of biometric information

The field of biometrics is concerned with establishing the identity of individuals by means of unique physiological or behavioral characteristics (modalities). In practical systems, several biometric modalities are used, such as fingerprint, face, iris, finger vein patterns, and so forth. Biometric systems are becoming increasingly popular because they allow for more secure solutions than traditional means for authentication such as PIN codes, passwords and security badges because a biometric is tightly linked to an individual. For the same reason, biometrics can prevent the use of multiple identities by a single individual. Finally, biometrics are often also considered more convenient because, unlike passwords and PIN codes, they cannot be forgotten and are always at hand [Keve 2007].

Biometric technologies are, however, not without their challenges (e.g. [Jain 2006]) and in order to optimally benefit from the increased security and convenience that biometrics have to offer, biometric systems have to be implemented in a proper manner. An important aspect to consider in during implementation is the privacy of the biometric reference information that is stored in biometric systems.

Over the past decades, a number of methods have been developed to protect the biometric information stored in biometric systems ranging from traditional encryption methods to methods specifically dedicated to the protection of biometric information. On a high level of abstraction, all these methods can be separated into two groups.

The first group contains methods that require the secrecy of a piece of information, normally referred to as a key. If this key is compromised and known to an adversary, he can get access to the biometric information that is protected based on the secrecy of the key. This type of approach for protection of biometric information presents the problem of key management: the overall biometric system must be organized such that the key will not become available to an adversary. One possibility is to set up a system of protocols and access rights that limit access to the keys to trusted operators only. An important drawback is that as these biometric systems scale up, the protocols become vulnerable to “incidents” by sloppy execution, change in regulations or legislation, human mistakes or intentional misuse. An other approach is to set up an architecture that, in normal operating conditions, never brings the key 'close to' the protected information. For example, one could think of a client-server architecture where the protected biometric information is stored on the server while the key is stored on the client. If the system then deploys a method that allows for biometric comparison in the encrypted domain, the key will never have to be available at the server. A drawback is that adversaries do not always follow the 'normal operating conditions' and the key might be stolen from the client to get access to the protected information on the server. Nevertheless, all these methods make it more difficult for an adversary to get access to the biometric information.

To circumvent the problem of key management, methods were developed that protect biometric information without using secret information. These methods are called Biometric Encryption (BE) methods. Other terminology that can be found in the scientific literature and in standardization
documents comprises Keyless Biometric Template Protection, Helper Data Systems, Fuzzy Commitment, Fuzzy Vault, Fuzzy Extractor, etc.

The aim of the BioKeyS III project is to assess the privacy and accuracy the Fuzzy Commitment Scheme as a promising representative of the Biometric Encryption methods as described in the following sections.

1.2 The BioKeyS III project

The German Bundesamt für Sicherheit in der Informationstechnik or BSI (in English: Federal Office for Information Security) is the German government agency in charge of managing computer and communication security for the German government. Its areas of expertise and responsibility include the security of computer applications, critical infrastructure protection, internet security, cryptography, counter eavesdropping, certification of security products and the accreditation of security test laboratories.

With the advent of the application of biometrics in both the public and private sectors, the BSI has become more involved in the assessment and development of biometric technologies. For example, in projects like BioFace and BioFinger, the classification accuracy of face and fingerprint systems was investigated. In a more practical setting, the BSI is running the EasyPASS pilot project at Frankfurt airport in collaboration with the German Federal Police to allow faster border passage by using the biometric information stored on electronic passports.

The BioKeyS project is concerned with BE methods and contains several sub-projects. The goal of the BioKeyS I was to optimize a BE prototype system based on the experiences from the BioKeyS-Multi project which investigated the feasibility of a BE systems. The BioKeyS II project gives an overview of the methods and standards of BE approaches and investigates methods to enhance BE systems with the use of passwords and identification systems.

The BioKeyS III project first defines a framework for assessing the privacy of BE methods which was published as "A Reference Framework for the Privacy Assessment of Biometric Encryption Systems", Proceedings BIOSIG 2010 - Biometrics and Electronic Signatures, Gesellschaft für Informatik (GI), Lecture Notes in Informatics (LNI) P-164, Seite 45-55, ISBN 978-3-88579-258-1, Koellen Druck+Verlag, Bonn, Germany, 2010. Furthermore, a new fusion method is presented which significantly enhances the privacy properties and is also easy to implement.

Based on this framework and traditional methods for determining the accuracy of biometric solutions, the project assesses in detail, by way of example, the privacy and accuracy of a commercial product, the BioHASH® SDK, using public and proprietary databases and it discusses the effect of possible attacks.

Furthermore this document presents an overview and the conclusions of the BioKeyS III project and gives recommendations for further research. The project started in November 2009 and was finalized November 2010. The partners that took part in the project were the BSI, secunet and priv-ID.
1.3 Overview of the document

The first part of the document, covering Chapter 2 through Chapter 5, gives a theoretical treatment of methods to protect biometric information. Chapter 2 defines security and privacy in biometric systems. By using the concept of perfectly private biometric systems it arrives at the goal of BE systems defined as preventing relevant biometric information to be obtained from storage facilities in biometric systems without the need for long-term secrets. Where Chapter 2 deals with purely theoretical notions of privacy and security and the goal of BE systems, Chapter 3 gives an overview of practical BE systems with a emphasis on the Fuzzy Commitment Scheme (FCS). Chapter 4 assesses the privacy properties of the FCS in the form of possible attacks. The conclusion in Chapter 4 is that the privacy of a Fuzzy Commitment Scheme (FCS) is close to what can be expected of BE systems. In order to further enhance the privacy properties of BE systems, fusion techniques can be used and Chapter 5 gives an overview of fusion methods that can be used in combination with BE systems. A new fusion method, called key level fusion (KLF), is introduced which significantly enhances the privacy properties and is also easy to implement.

The second part of the document, covering Chapter 6 through Chapter 8, is more practically oriented and gives accuracy and privacy results of the BioHASH® SDK as an example of a Fuzzy Commitment Scheme (FCS). Chapter 6 gives the accuracy of the SDK in the form of a DET curve using a public and a proprietary database, while Chapter 7 gives the accuracy result of key level fusion of two fingers and measures the increase in privacy that can be achieved using fusion. Finally, Chapter 8 reports on the experiences with a practical set-up of key level fusion tested on a population of 25 individuals.
2 Security and privacy in biometric systems

2.1 Introduction

In order to optimally benefit from the increased security and convenience that biometrics have to offer, biometric systems have to be implemented in a proper manner. On the technological side there are challenges such as accuracy, speed, scalability, cost, and interoperability. Besides these, it is important to consider the security of biometric systems as well as privacy issues regarding the information stored in these systems. Many of these challenges are related to the special properties of biometrics as compared to traditional means for authentication:

- biometrics are tightly coupled to an individual which makes revocation and re-issuing of authentication information unfeasible because clearly it is not possible to replace an individual's right index fingerprint or left iris. This is in contrast to PIN codes, passwords, tokens, etc. which can easily be revoked and re-issued;

- biometrics is personal information. For example, it might contain information on the health condition of an individual [Boll 2000][Penr 1965], gender, ethnicity, age, etc. Therefore, in contrast to PIN codes and passwords, in many countries biometrics are considered to be Personally Identifiable Information (PII) and use of biometrics is governed by privacy legislation (e.g. [Euro 2008]);

- each individual has a limited number of biometric modalities (e.g., one face, two irises, ten fingers) while the number of possible passwords or token identifiers is several orders of magnitude higher. As a consequence, an individual will have to re-use the same modality in different applications;

- In a biometric system, biometric information is compared using statistical classification methods due to the inherent variability of biometric measurements. This variability poses a limit to the uniqueness of biometric modalities. Although this limit to the uniqueness is also true for, say, 4-digit PIN codes, passwords and token identifiers allow for a higher level of uniqueness;

- biometric measurements are affected by noise and other forms of variability while authentication protocols based on passwords and the like rely on 'bit-exactness' of the authentication information. This variability limits the distinctiveness of biometric features. Although this limitation also applies to, say, 4-digit PIN codes, passwords and token identifiers allow for a higher level of distinctiveness than single biometric modalities.

These special properties of biometrics when used for authentication have an impact on security and privacy considerations of biometric systems. Biometric Encryption (BE) technologies can make an important contribution in solving some of these vulnerabilities [Cavo 2007]. This chapter is concerned with defining the goal of BE technology. Starting with a general overview of a biometric system, security and privacy vulnerabilities of this system are given. Based on these vulnerabilities, a (conceptual) perfectly private biometric system is defined and it is shown that most vulnerabilities can be solved using traditional cryptographic techniques without the need for long-term secrets. This leads to the ultimate goal of BE technologies and the (conceptual) perfectly private BE system.
2.2 Security and privacy

2.2.1 Overview of a biometric system

In Figure 1 a high-level overview is given of a biometric system which can be used for verification (authentication) and identification of individuals. In verification, a person claims to have a certain identity and the biometric system performs a 1:1 comparison between the offered biometric and the biometric reference information that is linked to the claimed identity and stored in the biometric system. In identification, a 1:m comparison is performed between the offered biometric template and all available reference information stored in the biometric system to reveal the identity of an individual. Without loss of generality we will only consider verification systems as the 1:m comparison in an identification system is, in general, implemented as a sequence of 1:1 comparisons.

During enrollment, a fingerprint sensor SENS generates the image of a fingerprint. After processing the image and extracting relevant features in the feature extraction (FE) block, a template representing the fingerprint is stored in the biometric system (STOR). During verification, an individual claims an identity, and a so-called probe image from this individual is obtained. This image is transformed into a template and compared (COMP) with the template stored in the biometric system corresponding to the claimed identity. The comparison produces a similarity score and applying a threshold T to this score leads to an “Accept” or “Reject” message.

2.2.2 Insertion and eavesdropping

Figure 2 depicts some important vulnerabilities of a biometric system (see also, [Buha 2008] [Cavo 2007] [Jain 2004] [Rath 2001]). As illustrated by the gray rectangle, in many cases the enrollment functionality can be assumed to operate in a secure environment and the most important vulnerabilities are concerned with storage and verification functionality. Although in the literature a
large number of potential attacks is mentioned, we propose to group them into the following categories.

**Insertion**, depicted with ingoing arrows in Figure 2, which contain methods such as
- *spoofing*: a fake biometric sample constructed from biometric data acquired from an individual is presented to the sensor;
- *masquerade*: a biometric sample synthesized from template information is presented to the sensor;
- *similarity*: exploiting the similarity of biometric modalities between individuals;
- *replay*: insert a previously recorded message;
- *tampering*: modifying a message or inserting a synthesized message;
- *substitution*: replacing a biometric template in the storage facilities by another template.

**Eavesdropping**, depicted with outgoing arrows in Figure 2, containing methods such as
- *stealing*: obtaining the fingerprint from an individual using the latent fingerprint from a fingerprint sensor;
- *wiretapping*: obtaining the messages that are exchanged in the system;
- *copying*: obtaining a biometric information from the storage facility.

In the context of biometric systems we associate the **security** of a biometric system with *insertion* vulnerabilities while the **privacy** of a biometric system is associated with *eavesdropping* vulnerabilities.

Thus, the **security** of a biometric system defines how difficult it is to illegitimately be accepted by the system. In contrast, the **privacy** of a biometric system is related to the difficulty to obtain any relevant information from a provided biometric characteristic other than a verification decision. (See [Bree 2009] for similar definitions.)
2.2.3 A Perfectly Private Biometric system

If we assume that enrollment takes place in a trusted environment, a Perfectly Private Biometric (PPB) system can be defined as the gray rectangle in Figure 3 (Note that the concept of a PPB is also used in [Bree 2009] which serves as a basis for a new ISO standard [ISO JTC1]). A PPB system contains a sensor SENS, a feature extractor FE, a comparator COMP, a threshold module T and storage STOR. When offering a fingerprint, possibly in combination with an identity claim, the system outputs an Accept or Reject decision.

This PPB system is perfectly private in the sense that it outputs the minimal required amount of information in the form of a binary Accept/Reject decision. Furthermore, assuming a sensor leaving no latent prints (e.g. a touchless sensor or a swipe sensor), the system has no eavesdropping vulnerabilities. On the other hand, even if the system contains a sensor with perfect liveness detection, if the comparator (COMP) is not perfect the system will occasionally accept an illegitimate finger (due to a wrong decision of the comparator) and in that sense the system is not perfectly secure.

![Figure 3: A Perfectly Private Biometric system](image)

2.2.4 A practical implementation

The broadest goal of biometric system design is to build a perfect biometric system i.e., a system that is perfectly secure and perfectly private. However, biometric system design still contains a number of challenges [Jain 2004]. For example, although biometric comparison algorithms (matchers) are improving, there are no biometric systems exhibiting perfect recognition.

In practical systems, many security and privacy vulnerabilities can be prevented using standard cryptographic protocols. The replay and tampering attacks can be prevented by combining message uniqueness (by, for example, time stamps or sequence numbers) and verifying message authenticity. Standard cryptographic techniques allowing an authenticity check are Message Authentication Codes (MACs) and digital signatures. Although MACs do not provide entity authentication, in the context of a biometric software system where different software modules are part of the same DLL
this can to an extent be achieved by DLL signing and checking the signature of the DLL before it is loaded. MACs do not require long-term secrets but can use a short-term key obtained by, for example, a Diffie-Hellman key exchange protocol [Diff 1976]. A more secure approach is to use MACs based on authenticated key exchange. Although these methods require at least mid-term secret keys, management of these authentication keys is much easier than management of keys used to protect, for example, stored information. A substitution attack can be foiled by adding a digital signature over a template during enrollment which is verified using a public key during verification. Likewise, the wiretapping attack can be thwarted using secure channels which can also be generated by the Diffie-Hellman key exchange protocol.

2.2.5 Protection of stored biometric information

The secure channel to prevent a wiretapping attack can use keys that need only be available for the time the message is in the channel and in that sense they are short-term keys and need not be stored. The copying attack is very similar in nature to a wiretapping attack by interpreting storage as a communication channel. In a (storage) communication channel the message (in this case, the biometric information to be stored) is in the communication channel for as long as the message is in storage. This means that the short-term keys should be retained for the total time the message (template) is in storage. Therefore, the short-term keys become long-term secret keys that must be stored in the biometric system. This means that in order to protect biometric information from a copying attack using standard cryptographic protocols, a long-term secret is required.

2.2.6 The goal of Biometric Encryption

In the previous section it was explained that, in order to prevent abuse of stored biometric information using traditional cryptographic techniques, long-term secrets are required. Because access to the keys means access to the biometric information, it is important to keep these keys secret, even more so because of the special properties of biometric information mentioned in Section 2.1.

In principle, there are two approaches for storing and handling these long-term secret keys. The first approach is that the keys are kept inside the biometric system where the biometric system owner controls the keys. In a practical situation, this will lead to protocols and access rights that are limited to trusted operators only. An important drawback is that if these biometric systems scale up, the protocols become vulnerable to “incidents” by sloppy execution, change in regulations or legislation, human mistakes or intentional misuse.

The second method to handle long-term secrets is that the keys are stored outside the system. For this scenario, in many proposed systems the user will have control over the key. For example, the key can be stored on a smartcard or token, or it can be derived from a password, PIN code, pass phrase, answers to personal questions, etc. An advantage of this method is that the user is in control of the key and thus of his own biometric information, and the system owner does not have access to the decrypted biometric information. A drawback is that the user has to carry a smartcard or token or has to memorize a password or pass phrase. Furthermore, a biometric system based on tokens

1 Note that an alternative method to protect biometric information is to use physically protected storage such as smart cards that perform a biometric comparison on the card (on-Card-Comparison). This document considers software solutions only and physical protection methods will therefore be omitted from the discussion.
does not allow biometric identification (1:m) functionality, thus limiting the scope of biometric applications.

The purpose of Biometric Encryption technology is to eliminate the drawbacks of both approaches by obviating long-term keys. This prevents key abuse by the biometric system owner while at the same time allowing biometric identification. Thus, the goal of Biometric Encryption technology can be formulated as to prevent relevant biometric information to be obtained from storage facilities in biometric systems without the need for long-term keys.

Note that some publications state that the goal of Biometric Encryption is to derive or release a cryptographic key that can be used to encrypt personal data etc. Our notion is that generating a key from a biometric is a secondary effect and that the actual goal of BE is to protect biometric information. A motivation for this is that the derived key can only be as secret as the biometric data, which adversaries could acquire from other sources, e.g., from traces or by physical contact.

Ongoing ISO standardization activities [ISO JTC1] more specifically state that safeguarding the privacy of the data subject comprises the following aspects:

- Preventing anyone to have access to biometric data or derived attributes thereof that are not required and agreed upon by the data subject (e.g. any information other than required for biometric identification or verification in a specific service context);
- Ensuring that third parties and external observers have no access to the biometric references (and the associated identity records).

2.2.7 Perfectly Private Biometric Encryption

The essence of all BE systems (see Section 3.2 for an overview) is that biometric information, before it is stored in the biometric system's storage facility, is first transformed into a representation from which it is impossible to retrieve any biometric information. Clearly, in view of the discussion in Section 2.2.6, this transformation should not rely on long-term secret information as a means to protect biometric information (other than the biometric data itself). A possible candidate for such a transformation is a (keyless) one-way function. The most important property of a one-way function is that, given an input, it is easy to evaluate while, given an output, it is extremely difficult to find a corresponding input.

In that sense, on a high level of abstraction, BE technology mimics the way passwords and PIN codes are protected. When choosing a password, a one-way function in the form of a cryptographic hash function \( h \) is applied to a chosen password \( pwd \) and the hash of the password \( h(pwd) \) is stored in the system's storage facility. During authentication, a candidate password \( pwd' \) is compared with \( pwd \) without first retrieving \( pwd \) from \( h(pwd) \). BE technologies differ in the type of one-way function that is used, the domain in which the comparison takes place (e.g., the image domain, the feature domain, etc.), how the comparison is performed (e.g., using a traditional biometric comparator such as a minutiae matcher [Rath 2001], a cryptographic zero-knowledge protocol [Keve 2007]) and so forth.

2 The most obvious functionality that cannot be token-based is the so-called duplicate check, which prevents a user to enroll multiple times under different identities.
Since conceptually, any type of measurement processing or feature extraction can be thought to be part of the BE technology, a Perfectly Private Biometric Encryption (PPBE) system can be depicted as in Figure 4, where again we assume that enrollment takes place in a secure environment and hence, the enrollment functionality is omitted. It contains a feature extractor FE, a state-of-the-art comparator COMP, a threshold module T and storage STOR. It accepts the image of a fingerprint, possibly in combination with an identity claim, and outputs an Accept or Reject decision.

This (conceptual) PPBE system does not need any secrets while at the same time, it does not leak any biometric (other than the binary Accept/Reject decision) and in that sense is the highest achievable goal of any practical BE system. It is important to notice that vulnerabilities that are common between a PPBE system and a practical implementation thereof cannot be attributed to limitations of the implementation but must be attributed to the limitations of the use of biometrics. These intrinsic biometric vulnerabilities need to be addressed by proper system design such as multi-factor authentication or fusion (see Chapter 5).

2.3 Conclusions

Chapter 2 illustrated that the security and privacy vulnerabilities of biometric systems can be categorized as insertion vulnerabilities and eavesdropping vulnerabilities where the former relate to the security of the system while the latter refer to the privacy of the system. Most of these vulnerabilities can be solved using standard cryptographic techniques without the need for long-term secrets and the related problem of managing access to these secrets. The only vulnerability requiring a long-term secret when using standard cryptographic techniques is related to obtaining biometric information from a biometric storage facility. The goal of Biometric Encryption technology is to enable long-term storage without the need for long-term secrets in such a way that it is impossible to retrieve relevant biometric information from the stored information.
3 The Fuzzy Commitment Scheme (FCS) and other BE methods

3.1 ISO SC27 24745 – Biometric information protection

Currently, the standard “ISO/IEC JTC1 SC27 2ndCD 24745 – Biometric information protection” [ISO JTC1] is being developed. This international standard provides guidance for the protection of biometric information under various requirements for confidentiality, integrity and renewability/revocability during storage and transfer (see also Section 4.2). In addition, this standard provides requirements and guidelines for the secure and privacy-compliant management and processing of biometric information. Figure 5 shows the ISO reference architecture for Biometric Information Protection. In this architecture, a private template is denoted as \((AD, PI)\), where \(AD\) is the Auxiliary Data and \(PI\) is the Pseudonymous-Identifier. During biometric matching (comparison), \(AD\) is combined with a live measurement leading to \(PI^*\) after which \(PI\) and \(PI^*\) are compared to determine if there is a match.

The standard is formulated such that it supports a large range of methods for protecting biometric. These include methods that require secret information such as standard encryption techniques and dedicated methods like, for example, biohashing [Teoh 2004], secret template permutation [Brai 2002][Tibe 2002] and convolution with a secret random kernel [Savv 2004].

As explained in Chapter 1, the purpose of BE methods is to protect biometric information \textit{without} the need for long-term secrets and the new ISO standard is particularly well suited for BE systems. In Section 3.2, an overview will be given of existing BE systems. One of these systems, the so-called Fuzzy Commitment Scheme (FSC) is the underlying method of the BioHASH® SDK and will be explained in more detail in Section 3.3 while additional techniques that can be used in combination with the FCS will be given in Section 3.4. It will be shown in Section 3.3.6 that BioHASH® templates can be expressed as \((AD, PI)\) and the BioHASH® system follows the standard [ISO JTC1].

![Figure 5: The ISO reference architecture for template protection](image-url)
3 The Fuzzy Commitment Scheme (FCS) and other BE methods

3.2 Overview of BE methods

3.2.1 Mytec

The first BE scheme and a practical implementation thereof was developed by a company called Mytec Technologies Inc. which later changed into Bioscript Inc. and was finally acquired by L1. They also coined the term 'Biometric Encryption'. The main ideas of their approach are described in for example [Sout 1998][Sout 1998(2)]. The method works directly on (fingerprint) images and protects the information by multiplying the phase part of the Fourier transform of a (fingerprint) image $F(\omega)$ with a random phase function $\phi(\omega)$ in the frequency domain. The information $H(\omega)=F(\omega)\phi(\omega)$ is stored as helper data and does not reveal information on the fingerprint image because multiplication with random data mimics a one-time-pad. A secret $S$ is embedded by pointing to certain bits in $c(x)$, which is the inverse Fourier transform of the random phase function $\phi(\omega)$ multiplied by the magnitude part of the (fingerprint) image “optimal” filter. For a single bit in $S$, several locations in $c(x)$ are used. All locations to be used in $c(x)$ are stored in a look-up table. During verification, a noisy version of $c(x)$ is obtained from which the secret $S$ can be recovered by retrieving the bits at the locations indicated by the look-up table. The bit retrieval is followed by an error correcting step to remove the noise from the obtained bit string. The weaknesses of the Mytec scheme are due to the use of simple repetition codes rather than more sophisticated error correcting codes.

3.2.2 Cancelable biometrics

In [Rath 2001] the authors introduce an approach known as cancelable biometrics3. During enrollment, the image of a biometric is obtained, for example, the image of a fingerprint. In order to protect its privacy, the biometric image is distorted using a parametrized one-way geometric distortion before storing it in a biometric system. The function is made such that from the distorted image it is difficult to retrieve the original image and comparison can be done using the distorted images. Furthermore, using a different parameter for the distortion function, it is possible to derive several distorted images from a single biometric image (template). The major drawback of this approach is that there is no mathematical foundation that allows assessing the security and privacy properties of the system.

3.2.3 Fuzzy vault

The fuzzy vault method as introduced in [Juel 2002] is a general cryptographic construction allowing to store a secret $S$ in a “vault” that can be locked using an unordered set $X$ over some universe. The secret $S$ can only be retrieved from the vault using a set $Y$ if the sets $X$ and $Y$ have sufficient overlap. The authors mention biometric template protection as a possible application where $X$ is the biometric template obtained during enrollment. During authentication, the rightful owner of the secret can unlock the vault using a measurement of his biometric $Y$ that is sufficiently

3 The term 'cancelable biometrics' is somewhat misleading because clearly the biometric itself cannot be canceled. In the context of BE methods, the terms 'cancelable' and 'revocable' refer to the property that authentication information can be changed and revoked.
similar but not necessarily identical to \( X \). This approach allows deriving different representation of the same set \( X \) by choosing a different secret \( S \). The special property of the fuzzy vault scheme is that it is robust with respect to the order of elements within the sets which, in theory, makes the method well suited to be used with minutiae fingerprints (see e.g. [Ulud 2005]. Recent publications (e.g. [Clan 2003][Sche 2007][Nand 2007]) reveal a number of vulnerabilities when the fuzzy vault is used in real-life biometric systems. More specifically, the correlation attack between several enrollments as presented in [Kohl 2008] is still not satisfactorily solved [Miha 2009].

3.2.4 Biotope™

In the recently proposed Biotope™ scheme [Boul 2006][Boul 2007], each component \( x \) of a feature vector is translated by \( t \) and scaled by \( s \) to obtain \( v = (x-t)/s \) (see www.securics.com). The resulting value \( v \) is separated into the integer \( g = \lfloor v/E \rfloor \) and the residual \( r = v \mod E \) such that \( v = g + r \) where \( E \) is a parameter. The residual part \( r \) is stored as part of the template, but the integer part \( g \) is first passed through a one-way function to obtain \( w \) which is then stored. The one-way function can be a cryptographic hash function or encryption using a public key. The latter choice has the property that it allows controllable reversibility which can be useful in certain applications. The entities \( t, s, r, \) and \( w \) are called a Biotope™ and are stored as a template.

During verification each feature vector component \( y \) is, using the values \( s \) and \( t \) from the stored template, transformed into candidate values for \( g' \) and \( r' \) as \( (y-t)/s = g' + r' \). The value \( g' \) is passed through the chosen one-way function to obtain \( w' \). In order to get a match, \( w \) and \( w' \) must match exactly. For the list of residuals \( r \) and \( r' \) corresponding to the individual feature vector components, a robust distance measure is used to further determine if there is a match\(^4\). The Biotope™ scheme is commercialized by Securics (www.securics.com).

3.2.5 Biocryptics™

The Norwegian company Genkey (www.genkeycorp.com) has developed an approach referred to as Biocryptics™. From the limited information available [Lyse 2006][Duff 2002], the approach is believed to work directly on continuous features. In order to cope with noise and other variabilities, a correction vector is stored aiming to shift a measured feature to the middle of a quantization interval that defines one bit of a binary string to be embedded in the biometric template. In that sense the method is very similar to the so-called shielding functions as proposed in [Linn 2003][Linn 2007] which was inspired by Quantization Index Modulation, a method to embed a pre-defined watermark in an audio or video stream. During enrollment, the embedded binary string is put through a module that generates two large prime numbers. These numbers are used to generate an RSA public and private key. The correction vector, together with the public key is stored as a template while the secret key is discarded. During verification, the binary string recovered from the live biometric and the stored error vector is put through a module to generate an RSA public and private key. In this case the public key is discarded and the private key is used in a standard cryptographic authentication protocol to prove that it is compatible with the public key stored in the template. The attractive property of the approach is that the use of RSA public and private keys facilitates coupling this biometric system with PKIs. Although Genkey claims to be able to generate 1024 bits RSA keys, the entropy in these keys is not reported.

\(^4\) Here, robust refers to outliers in the sequence of distances \( ||r - r'|| \).
3.2.6 Fuzzy Commitment

The Fuzzy Commitment scheme was proposed by Juels and Wattenberg [Juel 1999]. It is the simplest yet the most studied among all BE schemes and is considered most suitable for biometrics that have a template in the form of an ordered string or feature vector. The method is an extension of a cryptographic commitment but it allows some variability in the committed value by using error correcting codes.

The underlying method of the priv-ID BioHASH® system is the Fuzzy Commitment Scheme (FCS) and therefore we will give more details in the rest of this chapter.

3.3 A practical implementation of the FCS

In this section we give an overview of a BE system for fingerprints (an FPBE system) based on the FCS. Although in theory the FCS is well suited to implement a BE system, it implicitly assumes that biometric measurements (in this case fingerprint images) can be represented as binary strings that give good classification results when compared using a Hamming distance measure. That is, if two images are similar, the Hamming distance between the corresponding binary strings should be small. Especially for fingerprints, that are normally characterized by an unordered set of minutiae locations, building a practical BE system from a FCS is not straightforward and requires in-depth knowledge of classification theory, signal processing, information theory, etc. Dedicated processing is required to, for example, assess image quality, align input images, remove variability between images and enhance signal-to-noise ratios. In the following sections an overview is given of a practical BE system and all the individual blocks are discussed. The basic ideas are also used in the priv-ID BioHASH® system.

3.3.1 Overview of the system

In Figure 6 an overview is given of a practical FPBE system. From left to right the system contains the following processing blocks.

- **Feature vector generation**: the first step in the chain is to generate a real-valued feature vector representing the input fingerprint image where it is important that the signal-to-noise (S/N) ratio is sufficiently high such that the following step (Bit string generation) leads to a robust binary representation. The result of this first stage is the feature vector \( f \).

- **Bit string generation**: as mentioned above, the FCS requires the biometric measurement represented as binary string. The Bit string generation block transforms the feature vector \( f \) into a binary string \( X \) that can be classified using a Hamming distance classifier.

- **FCS**: the fuzzy commitment scheme is the core of the BE system and protects the biometric information represented as the binary string \( X \).
- **Hashing**: in the FPBE system, the hashed fingerprint information is the actual reference information. A comparison in the FPBE system will only lead to an Accept if the hash value generated during enrollment and the hash value generated during verification are identical.

The following sections describe the individual blocks in more detail.

### 3.3.2 Feature vector generation

In order to arrive at the best recognition performance, after reading a fingerprint image, most systems start with a variety of processing steps. In many cases, including the BioHASH® SDK, a first step is to align the input images to adjust translation and rotation. In the case of fingerprints, these methods are usually based on the location of the core and other singularities. Next, (digital) signal processing steps are performed on the fingerprint image, such as linear and non-linear filtering to remove noise from an image, further filtering to facilitate extraction of a set of distinguishing features, image alignment, and so forth. The actual processing that is done usually depends on the succeeding processing steps where distinctive features are extracted from the image.

The majority of the fingerprint systems relies on minutiae features which are defined as a ridge ending or ridge bifurcation. Most fingers have about 30-40 minutiae [Malt 2003] and a set of minutia locations is considered to be unique for an individual. Using minutiae, a fingerprint can be well represented as an (unordered) set of minutia locations.

Unfortunately, these unordered sets are not very well suited to be combined with the FCS. For this reason, methods were developed that transform minutiae sets into feature vectors. For example, [Xu 2007] [Xu 2009] present a method that generates a feature vector that is independent of the ordering of the minutia locations and the dimension of the feature vector is fixed and does not depend on the number of minutiae.

In the past decades there also has been a lot of progress in the field of pattern recognition and image processing. For different reasons, this has motivated researchers to treat fingerprints as a pattern of ridges rather than a set of minutiae locations. In many cases, patterns can be described as vectors (e.g. [Baze 2002][Tuyl 2005]). Combining the feature vector generated from the minutiae information with the feature vector from the pattern information results in a real-valued feature vector \( f \) representing the fingerprint image.

### 3.3.3 Bit string generation

In Section 3.3.2 it was illustrated that fingerprints can be represented as feature vectors \( f \). In order to be combined with the FCS, these real-valued feature vectors must be transformed into binary strings such that binary strings derived from similar input images have a small Hamming distance in order to obtain good classification results. Preferably, the bits in the binary string are statistically independent (or at least uncorrelated). The transformation into bit strings is achieved using dedicated quantizers.

In the past decade a number of quantizers was proposed in the literature [Linn 2003][Tuyl 2005] [Zhan 2004][Chen 2007][Chen 2008][Chen 2009]. All these quantizers operate on a single feature. After all separate features have been transformed into a (short) bit string, all the bit strings can be combined to obtain a binary string representation \( X \) of the feature vector \( f \). In more advanced schemes, the number of bit extracted might be different for every feature. By assigning more bits to
good features and less bits to bad features, the overall accuracy of the systems can be improved. Summarizing we have that the output of the bit string generation step consists of the binary string $X$.

### 3.3.4 FCS

The Fuzzy Commitment scheme, proposed by Juels and Wattenberg [Juel 1999], is given in Figure 7. For simplicity, we assume that a binary error correcting code is used (operating on GF(2)) and hence we use XOR operations between bit strings. In the case of a symbol based code, the XOR would be replaced by (symbol-wise) addition and subtraction in the finite field.

During enrollment, a random number $S$ is generated. Using any cryptographic hash function (such as SHA-1, SHA-256, MD5) a hash value $h(S)$ is obtained and stored. Furthermore, $S$ is put through the encoder (ENC) of an error-correcting code (ECC) to obtain the codeword $C$. Finally, the biometric measurement $X$ represented as a binary string, obtained through one of the algorithm in Section 3.3.3, is combined with $C$ through an XOR operation to obtain so-called helper data $W = C \oplus X$ which is also stored. Thus, the input of the FCS is $X$ and the output consists of $(W, h(S))$.

During verification, a new biometric measurement $Y$ (derived using methods described in Section 3.3.3) is combined with $W$ to obtain a candidate codeword $C' = W \oplus Y = C \oplus (X \oplus Y)$. This candidate codeword $C'$ is put through an ECC decoder DEC to obtain a candidate secret $S'$. Finally, the hash of $S'$ is compared with $h(S)$ and if they are identical, an Accept message is generated indicating that $X$ and $Y$ are generated from the same biometric.

Considering the scheme in more detail the following observations can be made. Firstly, the FCS assumes that a biometric can be well represented by a binary string $X$ (or $Y$). Furthermore, assuming the ECC is a binary error-correcting code, the scheme will output an Accept message only if the Hamming weight $\|X \oplus Y\|$ is not larger than the number of errors the ECC can correct. This implies that the biometric representations $X$ and $Y$ are effectively compared using a Hamming distance classifier. Secondly, the biometric $X$ is obfuscated by the randomness in the codeword $C$ mimicking a one-time pad construction. Likewise, the codeword $C$ and hence the secret $S$ is obfuscated by the randomness in $X$. Obviously, the inherent redundancy in $C$ and $X$ implies that the one-time pad is not perfect, and consequently leaks some information. The notion of obfuscation and the quantification of the information leaked will be treated more formally in Section 4.5.4.
The priv-ID BioHASH® system internally uses the FCS to protect biometric information deploying a BCH error-correcting code while the generation of the strings \( X \) and \( Y \) uses methods as described in Section 3.3.3. The hashing that is used in the BioHASH® system is SHA-256. However, the use of the hash algorithm is different from the one proposed in [Juel 1999] and will be explained in the following section.

### 3.3.5 Hashing

The purpose of the hash in the FCS is to prevent access to the secret \( S \). It can be seen from Figure 7 that knowledge of \( S \) (and \( W \)) reveals \( X \) as \( X = ENC(S) \oplus W \). Even partial knowledge on \( S \) could reveal information about \( X \). Therefore, the hash function must be information hiding. Although in theory, no deterministic function can provide this property [Cane 1997], strong cryptographic hash functions are commonly believed to sufficiently conceal information for practical purposes. In order to impede retrieving \( S \) from \( h(S) \) the hashing in the BioHASH® system is implemented as follows.

Firstly, rather than choosing \( S \) as the only argument of the hash function, the argument of the hash consists of the concatenation of \( S \) and \( W \), thus rather than storing \( h(S) \) the value \( h(S|W) \) is stored.

Two important properties of hash functions are one-wayness and collision resistance. In the current setting, one-wayness is required to prevent access to \( S \) and is related to the privacy of the biometric information. On the other hand, a hash collision could lead to a false accept and thus has an impact on the security of the biometric system (See also Chapter 2). By truncating the output of the hash, the collision resistance decreases while the one-wayness increases. For example, if the truncation is such that the length of the hash output \(|h(S)| < |S|\) the uncertainty in \( S \) will be \(|S| - |h(S)|\) bits because there are on average \( 2^{|S| - |h(S)|} \) strings \( S \) that map onto the same hash value. Thus, hash truncation offers a trade-off between privacy and security. It is further interesting to notice that in a practical situation, hash truncation will not affect the overall accuracy of the system unless the collision probability approaches the biometric false accept probability (\( FAR \)).

### 3.3.6 The format of a FCS private template

Referring to Section 3.1, a private template in the generalized framework of the ISO standard SC27 24745 for “Biometric Information Protection” is denoted as \((AD, PI)\), where \( AD \) is the Auxiliary Data and \( PI \) is the Pseudonymous-Identifier [ISO JTC1]. Summarizing Sections 3.3.3 through 3.3.5 we have that a FCS private template contains the following information:

- The helper data \( W \) (see Section 3.3.4);
- The hash \( h(S|W) \) (see Sections 3.3.4 and 3.3.5).

In the context of [ISO JTC1], \( AD \) contains \( W \) and \( PI \) represents \( h(S) \) (or \( h(S|W) \)) such that a FCS private template as for example the BioHASH® private template follows [ISO JTC1].

### 3.4 Additional techniques

The BioHASH® system uses some additional techniques to further enhance the convenience and privacy of the basic FCS. In the following sections these techniques will be explained.
3.4.1 Comparison score

The basic FCS compares two hash values and if these values are identical, an Accept message will be generated. Thus, the basic FCS generates a binary output and does not generate a measure on how similar both biometrics are. This approach is similar to a PPBE system which also generates a binary decision.

In practical situations, however, it might be useful to have a comparison score. For example, a comparison score allows to change the FAR and FRR behavior without the need to re-enroll individuals in the system. In case of a successful verification, the BioHASH® system allows to output a comparison score by performing some extra processing during the comparison process. Referring to Figure 7, in case of a successful verification we have that $S' = S$, $\text{ENC}(S') = C$ and $|C \oplus C'| = |X \oplus Y|$. Thus, by retrieving $C$ from $S'$ it is possible to determine the number of bit differences $D$ in $X$ and $Y$ which can be interpreted as a difference score for the comparison process.

Note that returning a score $D$ is a design choice. On the one hand it facilitates assessing the classification accuracy as well as integration in a real application. On the other hand it allows a hill-climbing attack once an initial match is obtained. An alternative implementation based on the above method that allows for a flexible FAR and FRR behavior is to have a similarity threshold $T_D$ as an input. When the $D > T_D$ an Accept message will be generated. The implications for either choice concerning the privacy of the solution are discussed in more detail in Section 4.5.2.

3.4.2 Privacy bootstrapping

The classification performance of the FCS completely depends on the classification of the binary strings $X$ and $Y$ using a Hamming distance classifier. Although with this approach good classification results can be achieved, it places high demands on the quality of the binary string. In order to easily cooperate with 3rd party vendors, a technique called privacy bootstrapping (PBT) was developed. PBT facilitates the combination of the native-private algorithms available in the BioHASH® system with 3rd party template generation and comparison functionality while still an overall private template is obtained without the need to store long-term secrets. The approach for PBT is depicted in Figure 8. On a high level of abstraction, the BioHASH® functionality is as follows. Given the image of a fingerprint during enrollment, BioHASH® generates $(W, h(S))$. During verification, given a fingerprint image and $(W, h(S))$, BioHASH® generates a score $D$ (see Section 4.4.1).

As explained above, for a successful verification the value $h(S)$ will be regenerated exactly and, using extra processing, can be used to derive a cryptographic key $K$. During enrollment this key $K$ can be used in combination with a standard encryption algorithm (such as AES) to encrypt a template $T$ generated by a 3rd party extractor. After encrypting $T$, the key $K$ is discarded and the encrypted template $E_{K}(T)$ is stored together with $(W, h(S))$ as a private template. During verification a candidate key $K'$ is retrieved to decrypt $E_{K}(T)$ leading to a candidate template $T'$ which can be compared with the 3rd party template $T^*$ generated from the live measurement using the 3rd party extractor leading to a score $Sc$. This 3rd party score $Sc$ can be fused with the score $D$ generated by the BioHASH® algorithms to obtain an overall score.
3.4.3 Binding templates to hardware

The BioHASH® system makes it possible to prevent that BE templates that are stolen in one application are used in another application provided that the other application's software implements the binding technique as well. Furthermore, it allows to trace back templates to the application where they were generated. The main idea is that Biometric Encryption allows to bind BE templates to hardware components. The method is closely related to the way machine-bound software licenses are constructed.

The principle consists of two mechanisms. The first mechanism is the use of an application specific identifier $AID$. As mentioned above, an essential part of the BioHASH® template consists of $h(S)$. By using $AID$ as an extra argument of the hash function $h$, the $AID$ is irreversibly bound to the template and cannot be removed without a re-enrollment. Moreover, without knowledge of $AID$ a BioHASH® comparison will always lead to a Reject message and will render the template useless. Thus, a first approach is to keep $AID$ secret.

However, using a secret $AID$ introduces a long-term secret into the system and although compromise of $AID$ will not lead to revealing information about the biometric, it would allow
BioHASH® templates to be used in other applications. In practical situations, keeping $AID$ secret is difficult, even more so because the secret $AID$ is required in every comparison.

As an alternative, $AID$ can be linked to hardware components in the biometric system. Many hardware components of computer systems have unique hardware identifiers ($HIDs$). Such a $HID$ is tightly linked to the physical hardware and might take the form of a hard disk serial number, a network adapter MAC address, a processor ID, etc. During set-up of the biometric system, using a private key, a digital signature can be generated over ($AID$, $HID$) which is then stored in the system. During verification the software retrieves a candidate application identifier $AID'$, polling the appropriate hardware to obtain a candidate identifier $HID'$ and with this information verifies the digital signature using the corresponding public key. If the signature verification fails the comparison will not be done. Note that the private key is only required during system initialization (or when hardware is replaced) and not during normal operating conditions. This approach effectively links BioHASH® templates to hardware components and renders templates useless when the appropriate hardware is not present thus preventing template theft. Using methods like secret sharing or multiple signatures enables a valid signature verification when only a subset of the hardware components is present allowing for distributed verification terminals.

### 3.5 Conclusions

This chapter described a number of practical BE implementations with an emphasis on the Fuzzy Commitment Scheme. It was explained how a basic Fuzzy Commitment Scheme template is constructed and some additional techniques were given to enhance the accuracy and privacy of the FCS.
4 Study of the privacy of a practical FCS

4.1 Introduction

In Chapter 3 an overview was given of Biometric Encryption in general and the BioHASH® system in particular. In the current chapter we will discuss threats against the privacy of this system. This chapter is structured as follows. First we will discuss some high level notions of privacy goals where we will follow the requirements that are currently being developed in [ISO JTC1]. Next, some high level and low level threats to the system will be described and these will be related to the notions of privacy formulated by the ISO standard. Finally some conclusions will be drawn.

4.2 Privacy requirements

In Chapter 2 a high level notion of privacy was introduced in terms of a (conceptual) PPB and PPBE system that leaks no information about biometric templates. The concept of a PPB system is also described in [Bree 2009] which serves as a basis for a new ISO standard that is currently being developed [ISO JTC1]. This standard provides guidance for the protection of biometric templates under various requirements for confidentiality, integrity, availability and renewability/revocability. More specifically the standard proposes the following privacy requirements concerning biometric information:

- irreversibility: To prevent the use of biometric data for any other purpose than originally intended, biometric data shall be transformed in such a way that the biometric sample or a deductible attribute that does not serve the agreed purpose of the identity management system cannot be retrieved from the transformed representation;
- unlinkability: The stored biometric references shall not be linkable across applications or databases;
- confidentiality: To protect biometric references against access by external observers resulting in a privacy risk, biometric references shall be kept confidential;
- data minimization: minimizing irrelevant and/or undesired processing of personal data, including during the verification of a person’s identity.

The standard does not prescribe the mechanisms of how to achieve these requirements but it is applicable to a range of techniques that is much wider than BE techniques including traditional encryption of the template. Although the standard does not explain in detail the difference between 'irreversibility', 'confidentiality' and 'data minimization' it can serve as a reference in assessing privacy of practical systems. For the purpose of this document we will interpret 'confidentiality' as that when external observers copy biometric references, they should not be able to use the biometric information. When using BE technology like BioHASH® this comes down to 'irreversibility'. The 'data minimization' requirement is defined in the context where biometric data and biographic data are combined and this will be considered outside the scope of the BE assessment. Summarizing we have that in the context of BE systems 'irreversibility' and 'unlinkability' are the most important requirements and consequently, the BioHASH® system will be assessed with respect to these two requirements.
4.3 Adversary capabilities

In Chapter 3 it was shown how a practical FCS can be constructed. Like any BE system, the practical FCS intends to obtain the functionality of a PPBE system and achieve the privacy requirements as formulated in [ISO JTC1].

Before an assessment of the privacy properties is done it is important to define the capabilities of the adversary. It is customary in the assessment of security algorithms such as, for example, symmetric ciphers to adopt the black-box model [Wyse 2009]. This assumes that an adversary knows all the details of the algorithm. During operation, the adversary has access to the inputs and/or the outputs of the algorithm but not to the internal intermediate computation results and secret keys.

In the case of BE we therefore assume that an adversary has access to BE templates and knows both the inner workings of the BE algorithm and the way in which the BE templates are constructed. Given these assumptions it is the purpose of the assessment to determine what a BE method contributes to the privacy objectives.

A second important notion in the assessment of security algorithms is that the best known attack defines the security of the algorithm. For example, if the General Number Field Sieve (GNFS) is the best (i.e., fastest and most efficient) known algorithm to factor the product of two large primes and if we assume that breaking RSA is equivalent to factoring, then the security of the RSA algorithm is directly related to the required effort (in terms of computation time and computation resources) of the GNFS to factor the public RSA modulus in its two primes. The notion of 'best known attack' will also be adopted in assessing BE solutions. The security in terms of bits is normally expressed as the log2 of the required effort for an adversary to achieve a certain goal (at least with sufficiently high probability).

4.4 A framework for privacy assessment

The observations in the previous sections lead to a framework for the privacy assessment of BE systems [Keve 2010]. Such a framework is useful to assess and compare the privacy properties of a BE system, find weaknesses and define counter measures. During the assessment, the following steps should be taken:

- Define privacy goals of BE systems in terms of the privacy properties mentioned in Section 4.2;
- Define adversary capabilities and goals following the notions in Section 4.3;
- Assess the BE system for (known) vulnerabilities and assess the complexity of the attack.

4.5 Privacy of the BioHASH® FCS system

Adversaries can mount attacks against vulnerabilities of security systems. In order to assess the privacy of the BioHASH® system in the sections below we will consider two types of threats:

- High level threats: these threats exploit high level information such as the inputs and outputs of the algorithm. Furthermore, the adversary can do enrollment and verification. High level threats do not exploit knowledge of how templates are constructed;
- low level threats: these exploit low level information such as knowledge of how a template is constructed. Low level threats might be used in conjunction with high level threats.

In traditional cryptanalysis one often describes the adversary capabilities in terms of ciphertext-only, known-plaintext, chosen-plaintext, chosen-ciphertext and adaptive chosen-plaintext. In the context of our current analysis we assume that the adversary has access to a collection of BE templates but, in line with the assumption of Section 2.2.2 that enrollment takes place in a secure environment, he is not capable of generating his own templates. Thus we will exclude the chosen-ciphertext capability of the adversary.

4.5.1 FAR attack

The FAR attack is a high level threat that can be used against any biometric algorithm and exploits the fact that practical biometric systems have a non-zero FAR. The FAR is the probability that the biometric system will incorrectly accept an unauthorized user so that occasionally the comparator falsely declares an image to be genuine. The FAR is normally defined in a verification setting. Thus, given a BE private template, the attack consists of trying sufficient images until an Accept message is obtained. If the comparator is operating at FAR=$\alpha$ then the expected number of trials to obtain a false accept is $1/\alpha$. The false accept rate and the FAR attack can also be defined in an identification setting.

![Figure 9: impostor images at various settings of the similarity threshold](image-url)
setting where templates are stored in a database. If the database contains $K$ records and the comparator is operating at $FAR = \alpha$, then the probability to be accepted by the system is approximately $K\alpha$. In this case, the FAR attack leads to obtaining biometric information from a random user.

It is important to note that the FAR attack is applicable also to the PPBE system introduced in Section 2.2.7. Therefore, if this attack is mounted against a practical BE system, it does not exploit a vulnerability of the BE implementation per se. Still, it allows the adversary to obtain information about the protected biometric information in the sense that a successful trial image is in some sense similar to the image that was used to generate the BE template.

The required effort to exploit the non-zero FAR of a comparator is proportional to the expected number of trial images $1/\alpha$ where every trial requires a comparison. The total required effort for a FAR attack can be split into two components. Firstly, in order to mount the attack, the adversary requires a database of trial images. We define the required effort to collect a single image by $C_{FAR}$. Secondly, if we denote the effort of a single comparison by $S_{FAR}$, the total expected effort for the FAR attack is $(S_{FAR} + C_{FAR})/\alpha$ or $\log_2(S_{FAR} + C_{FAR})/\alpha$ in terms of bits. It should be noted that $S_{FAR}$ is an effort that cannot be avoided. On the other hand, the effort $C_{FAR}$ can be re-used over several attacks. Moreover, $C_{FAR}$ can be reduced by using artificial fingerprint generators such as SFinGe [Sfin].

As an illustration, in Figure 9 some fingerprint images from the MCYT database are given. Every row of Figure 9 contains images from a different finger that result in a false accept on the features extracted by the BioHASH® algorithm for decreasing $FAR$ settings. It is interesting to note that a false accept on a lower FAR setting leads to an impostor image that is more similar to the images used to generate the BE template (at least by visual inspection). This suggests that a larger effort (i.e., a higher $1/\alpha$) results in a more similar image.

As mentioned above, the FAR attack is also applicable to Perfectly Private Biometric Encryption (PPBE) system and allows obtaining some information about the stored biometric characteristics. The FAR attack thus imposes a lower bound on the required privacy properties of BE templates in the sense that the required effort to obtain information directly from the BE template should be substantially higher than the effort for the FAR attack, assuming that both efforts yield the same information. Thus, the FAR attack puts a limit on the irreversibility requirement as mentioned in [ISO JTC1].

The FAR attack can be diminished by increasing $S_{FAR}$ (i.e., slowing down the comparator), at the system level by allowing a maximum number of trials per template and by decreasing $\alpha$ by operating the biometric system on a lower $FAR$ value. The latter option will, however, be at the cost of a higher $FRR$. Considering the classification performance of the BioHASH® SDK (see Chapter 6), the maximum $FAR$ using a single finger (i.e., no fusion of multiple fingers) can be limited to $10^{-4}$ with the $FRR$ remaining below 2%. A very powerful way to effectively decrease the $FAR$ is to match multiple fingers, and/or multiple modalities, such as, for example, a fingerprint/fingervein. For this approach to be effective against a FAR attack, the software needs to be arranged such that an adversary can only determine if there is a match when all the modalities are available. More information about fusion and the impact of privacy can be found in Chapter 5.

If different applications use PPBE systems, the FAR attack can also be exploited to link templates across applications. Using a normal FAR attack in a first PPBE system and using the obtained fingerprint images in a second PPBE system, an adversary can, with a certain error probability depending on the $FAR$ and $FRR$ of both systems, decide if an individual is present in the two PPBE
systems. Thus, the FAR attack also puts a limit on the unlinkability requirement of [ISO JTC1]. Therefore it is essential that the system design incorporates a strategy to prevent the FAR attack.

4.5.2 Hill climbing

The hill climbing threat exploits the continuity of a similarity score as a function of changes in the input image. By observing the similarity score when offering slightly modified input images, an image can be obtained with a high similarity score such that finally a good quality working image (or template) is obtained. Regarding privacy, this threat is similar to the FAR attack: the adversary obtains an image that is in some sense close to the image that was used to generate the BE template.

Referring to Section 3.3.4 it can be seen that BE systems are traditionally implemented such that they do not output a similarity score but just a one-bit Accept/Reject decision which thwarts the high level hill climbing threat. This allows to obtain a working image using the FAR attack but it does not allow to increase the image quality once a working image is obtained.

In Section 3.4.1 it was shown that extra processing allows to produce a similarity score. Using a similarity score facilitates the use of BE methods in practical situations but it is optional. There are two options:

- **Generate a similarity score as an output.** This allows the hill climbing threat but only once a working image is obtained. Because the system returns the similarity score, the adversary can immediately see if his modification resulted in a more similar image. The effort to obtain a working image using the FAR attack is related to the the maximum FAR that can be chosen in the system. Thus, the adversary should first have a matching image that generates a valid score that can then further be refined using the similarity score. Note that this is in contrast to conventional systems that generate a score for any fingerprint image, and the adversary can modify this random image to improve the score.

- **A threshold for the similarity score as an input.** Like in the previous situation the adversary should first have a matching image. When modifying the input image the adversary can most of the time not observe if and how the similarity score changes due to his modification because the system only returns an Accept/Reject decision and the adversary will have to try several threshold settings for the trial image to find the actual score. Thus, the hill climbing threat will require more effort than in the previous case (the actual increase in effort depends on how the adversary implements the hill climbing).

It is doubtful if a hill climbing attack in a traditional system, where a similarity score is generated, is more efficient than a FAR attack because for every new score a comparison is required [Mart 2006]. Although the modifications made to the input images in a hill climbing attack will be constrained (such that input image still looks like a fingerprint) thus reducing the search space, a FAR attacker will use real images from an available database or synthesized images which already look like a fingerprint. In case of a BE system that only generates a Accept/Reject output and where there is a fundamental limitation to the FAR, the hill climbing attack will not be a more serious threat than a FAR attack.
4.5.3 Hash inversion

A first low level threat exploiting knowledge of the template structure is inverting the hash of a private template (see Section 3.3.5). For good hash functions, the best known attack for hash inversion is to perform an exhaustive search (dictionary attack) which means that in the case of a BioHASH® template the required effort for inversion is proportional to \( |S| \) bits (i.e., the actual effort is proportional to \( 2^{30} \)).

It was explained in Section 3.3.5 that when a hash function is used as a part of a BE template, it can be beneficial to truncate the hash value. If \( |h(S)| < |S| \) the adversary is left with uncertainty about \( S \) and thus about \( X \). If \( |h(S)| > |S| \) a hash value \( h(S) \) is likely to have only one pre-image. In this case, if the hash is inverted, the adversary obtains \( X \) which is the binary string representing the biometric.

In many publications, access to \( X \) is considered equivalent to access to the biometric itself. However, \( X \) is not equivalent to a fingerprint image and going from \( X \) to a working image is not trivial, also bearing in mind the processing described in Section 3.3.3 and the considerations in Section 4.5.5. Furthermore, under the black-box assumption, it is impossible to insert the string \( X \) in the processing path and one would have to transform \( X \) into a working image. On the other hand, access to several \( X \) values obtained from different templates would allow cross-matching.

If, in spite of the reasoning in the previous paragraph, we make the assumption that \( X \) is equally useable as a fingerprint image obtained by a FAR attack, then hash inversion should be significantly more difficult than a FAR attack. If we neglect the effort \( C_{FAR} \), motivated by readily available public fingerprint databases and programs like SFinGe [Sfin], the effort for a FAR attack is \( S_{FAR}/\alpha \) while the effort for hash inversion can be expressed as \( S_{hash}2^{30} \) and we require \( S_{hash}2^{30} > > S_{FAR}/\alpha \). If we denote the comparison time excluding the hash process by \( S_{match} \), we obtain \( S_{hash} > > S_{match}/(\gamma-1) \) where \( \gamma=\alpha2^{30} \). Choosing practical values \( \alpha=10^{-4} \), \( |S|=30 \), \( S_{match} = 1\text{ms} \) and \( S_{hash} = 1\mu\text{s} \) it can be seen that hash inversion takes approximately 1000 times longer than a FAR attack. To increase this number, the speed factor \( S_{hash} \) can be increased by, for example, deploying a hash chain. In practical systems, \( S_{hash} \) can be increased by several orders of magnitude without impacting the usefulness of the solution in terms of matching speed. Thus, the threat of inverting the hash can be made significantly more difficult than a FAR attack.

It is further important to notice that a successful hash inversion does not bring the attacker the same information as a FAR attack. This is further discussed in Section 4.5.5.

4.5.4 Using the helper data \( W \)

A second part of the template that can be exploited in an attack is the helper data \( W = C \oplus X \) (see Section 3.3.4). If we assume that the code word \( C \) is chosen from an \((n,k,t)\) ECC then \( k \) bits of \( X \) are obfuscated (protected) by the \( k \) random bits in \( C \) because of the \( k \) bits of randomness in \( C \). Stated differently [Juel 1999][Dodi 2004], in an information theoretic sense \( W \) leaks \( n-k \) bits of information about \( X \). Firstly, it can be shown that if robustness against a certain number of bit errors in \( X \) is required, some leakage cannot be avoided [Dodi 2004]. In order to exploit the leakage, in theory the adversary can set up a linear system of \( n-k \) equations in \( n \) unknowns leaving him with \( k \) degrees of freedom. However, this information theoretic representation of leakage does not indicate to an adversary how this leakage can be exploited to learn dedicated information on the biometric string \( X \) that was used to generate \( W \). Thus, although \( W \) leaks information on the biometric in the
information theoretic sense and presents a threat to the irreversibility requirement, it is not clear
how this information should be exploited.

A more practical threat was recently introduced independently in [Cart 2008] and [Simo 2009]
exploring the use of linear ECCs and the fact that many practical ECCs are not perfect codes. In
short, if two helper data \( W_1 = C_1 \oplus X \) and \( W_2 = C_2 \oplus X' \) are obtained, the adversary can compute
\( W_1 \oplus W_2 = C_1 \oplus C_2 \oplus (X \oplus X') = C_3 \oplus (X \oplus X') \) where the property of a linear code was used that the
sum of two code words is again a code word. If this result can be decoded it is very likely (due to
the non-perfectness of the ECC and the distribution of \( X \)) that \( X \approx X' \) and the ISO requirement of
unlinkability is not satisfied. However, as will be shown in a forthcoming publication [Kelk 2010],
this threat can easily be prevented by introducing an application-specific non-private permutation
matrix which can be implemented as an orthonormal matrix on the feature vector \( f \) (see
Section 3.3.3).

\[
\begin{align*}
p_f(F) &
\end{align*}
\]

\[
\begin{align*}
p_{\mu}(F) &
\end{align*}
\]

\[
\begin{align*}
P(\mu|\mu) &
\end{align*}
\]

\[
\begin{align*}
p(\mu|\mu') &
\end{align*}
\]

\[
\begin{align*}
0 &\quad 1
\end{align*}
\]

Figure 10: Uncertainty of a feature given its binary representation

4.5.5 Deriving feature vectors from binary strings

It is mentioned earlier that many publications (implicitly) assume that access to \( X \) is equivalent to
access to the biometric itself. However, this statement depends on the attack model that is assumed.
Under the assumption of a black-box attack model, the adversary can not exploit any knowledge on
the internal results of intermediate computations, nor can he replace intermediate results of the
computation. Therefore, he cannot insert a string \( X \) during the verification process and the
(unprotected) string \( X \) can only be used for deriving a working fingerprint image. Little research
was done on how difficult it is to, given binary string representation \( X \), arrive at a fingerprint that
will lead to the same \( X \) (or a string close to \( X \))^5. A straightforward method would be to use a FAR
attack possibly in combination with hill climbing directly on \( X \). This would give an attacker a
working image but not the image that was used to generate \( X \). Because the attacker uses the regular
processing chain this can be called a forward approach.

---

5 Note that in the scientific literature several successful attempts were reported to derive a working fingerprint image
from a set of minutiae (as opposed to the a binary representation \( X \)). Please refer to [Hill 2007][Ross 2005][Galb
2010] for more information. However, with respect to general appearance, the reconstructed fingerprints hardly
resemble the original fingerprints.
Alternatively the attacker could deploy a reverse approach where he would try to obtain the feature vector $f$ from $X$ and from there the fingerprint image. It can be shown\(^6\) that given a binary string $X$ leaves much uncertainty about the feature vector $f$. In order to illustrate the mechanism, assume a 1-bit quantizer as depicted in Figure 10 (top-left) a single feature $(F)_i$ is quantized around the mean (see also [Tuyl 2005]). Assume that the following is given: the bit $b$ in $X$ (say $b='1'$), the background distribution $p_B((F)_i)$, the shape of the genuine distribution $p_G((F)_i)$ (but not the mean $\mu$) of the corresponding feature $i$ in $f$. It is then possible, using Bayes’ rule, to compute the distribution of $\mu$ as $p(\mu'|'1')=p('1'|\mu)p_G((F)_i)/P('1')$ where $P('1')=0.5$ because the quantization threshold divides $p_B((F)_i)$ in two equal parts. An illustration is given in Figure 10 showing there is uncertainty regarding the actual value $\mu$ of the feature $(F)_i$ for a given bit $b$ in $X$. Deriving feature vectors from binary string is an interesting topic for further research.

### 4.6 Conclusions

In this chapter, threats to the privacy of a FCS system were discussed under the black-box attack model. High level attacks, exploiting access to the input and outputs of a FCS system, and low level attacks, that also exploit knowledge of the template structure, were described.

It can be concluded that a FAR attack is always easier than any other high level or low level attack. Furthermore, given the black-box model, access to the binary biometric string still leaves uncertainty about the actual biometric features making it difficult to generate a working fingerprint image from a binary biometric string.

---

\(^6\) The work in the remainder of this section is the result of a discussion on March 17, 2010 with Raymond Veldhuis, Tom Kevenaar, Aweke Lemma, Emile Kelkboom and Haiyun Xu and might be published in the near future.
5 Approaches to fusion of private templates

5.1 Introduction

In many biometric applications, fusion techniques are used to enhance system performance [Ross 2006]. For example, in face recognition, fusion techniques can be used to combine the 2D and 3D features of a face leading to better classification results and a more secure biometric system. In a fingerprint-only system, several fingers can be fused to enhance the classification performance.

In traditional biometric systems, fusion can be done at three levels: feature vector, comparison score and decision level. The use of Biometric Encryption techniques, where enrolled templates are only available in encrypted form, makes it necessary to reconsider how the different modalities can be combined. Therefore, in the following sections we will discuss for all of these three levels how fusion can be combined with Biometric Encryption techniques. In order to do so we first give a high-level description of a Biometric Encryption module. Furthermore, we will introduce a fusion method that is dedicated for BE systems.

5.1.1 High level abstraction of Biometric Encryption

On a high level of abstraction, a Biometric Encryption module takes two inputs: a representation of a probe measurement $Y$ in the form of a feature vector or a binary string and a secure template denoted as $[X]$ obtained from a storage facility (e.g. a passport, a database). The output is a binary decision value $B$ indicating if the feature vector $Y$ corresponds to the BE template $[X]$. Thus, a Biometric Encryption module contains the comparison algorithm (see also Figure 4). In some implementations, in case of a match, a comparison score $D$ can be produced indicating how good the match is. If there is no match, no comparison score can be generated. This is depicted in Figure 11.

$$
\begin{array}{c}
Y \\
[\text{Biometric Encryption}] \\
[\text{B} \in \{0,1\}] \\
[\text{D} \in \{0,1\}]
\end{array}
$$

*Figure 11: High level abstraction of a Biometric Encryption module*

In the following sections it will be explained how the different fusion techniques can be combined with BE where we will assume, without loss of generality, that there are two modalities to be fused.

---

7 Some text books like [Ross 2006] also mention sensor (or sample) level fusion where the raw data is fused before any feature extraction is performed. In the context of BE this type of fusion is considered equivalent to feature level fusion.

8 To simplify the notation, in this section we denote a private template $(W,h(S))$ (or $(AD, PI)$) as $[X]$. 
5 Approaches to fusion of private templates

5.1.2 Fusion at the feature level

In feature level fusion, two feature vectors (or binary strings) $X_1$ and $X_2$ obtained during enrollment are combined to form a new feature vector $X_{tot}$. Similarly, during authentication, two feature vectors $Y_1$ and $Y_2$ are combined to form a new feature vector $Y_{tot}$ and both $X_{tot}$ and $Y_{tot}$ are sent to a comparator which produces a comparison score $D$. After applying a threshold $T$ an authentication is accepted or rejected (see Figure 12).

![Figure 12: Traditional fusion on the feature level](image)

In case of BE, the extension is straightforward in the sense that $X_{tot}$ and $Y_{tot}$ can be seen as a new feature vector that is protected by the BE system (see Figure 13).

![Figure 13: Fusion on the feature vector level with BE](image)

5.1.3 Fusion at comparison score level

In comparison score level fusion, two feature vectors $X_1$ and $X_2$ obtained during enrollment are matched individually against corresponding feature vectors $Y_1$ and $Y_2$ obtained during authentication. Every comparator produces a comparison score $D_1$ and $D_2$ indicating the similarity between $X_1$ and $Y_1$, and $X_2$ and $Y_2$, respectively. Both comparison scores $D_1$ and $D_2$ are combined and after applying a threshold $T$, a binary decision $B$ is produced.

![Figure 14: Fusion on the comparison score level with both modalities protected](image)
In case of using BE, comparison score fusion requires that the BE module returns a comparison score $D$. If this is the case, the architecture according to Figure 14.

### 5.1.4 Fusion at the decision level

For fusion at the decision level, the binary outputs $D_1$ and $D_2$ of the two comparators are combined at the Boolean level to obtain the final decision $B$. Since both a BE module and a non-protected module can output a binary decision, extension of fusion at the decision level with BE is straightforward. The resulting architecture is given in Figure 15. The most common decision fusion rules are the OR or the AND rule where in practical situations OR fusion slightly outperforms AND fusion.

![Figure 15: Fusion on the decision level](image)

### 5.1.5 Key level fusion

The use of BE methods give rise to a fusion method that is not possible in traditional biometrics. Referring to Section 3.3.4 it can be seen that the secret $S$ is put through a hash function $h$ and the result $h(S)$ is stored as part of the template. This opens the possibility to do key level fusion. Suppose that two modalities are given that generate a secret $S_1$ and $S_2$, respectively. Key level fusion then consists of storing $h(S_1|S_2)$. In Figure 16 this is worked out in more detail for the FCS for two modalities where the fused private template is represented as $(W_1, W_2, h(S_1|S_2))$. Because all hash arguments should be correct in order to obtain the correct hash value, inherently key level fusion is necessarily a form of AND decision level fusion.

![Figure 16: Key level fusion](image)

---

9 Key level fusion was proposed by Johannes Merkle.
5.2 Implications for the different fusion methods

From the previous sections it follows that, if a BE method can return a comparison score, in principle all traditional fusion methods are possible with protected templates. Which method is best suitable for a certain application depends on several considerations.

Regarding maximum achievable accuracy, in theory, feature vector fusion will lead to the best results. The high dimensional combined feature space gives the optimal freedom to separate the genuine feature vectors from the impostor feature vector. In practice, however, the situation might be different. Feature vector fusion requires an accurate model of probability distributions in a high-dimensional space in order to derive the feature vectors $f$ (see Section 3.3.2). In a BE setting, the accuracy of this model also influences the quality of the binary representation $X$ of a feature vector $f$ (see Section 3.3.3). An accurate model estimation might turn out to be problematic in a practical situation. An inaccurate estimate of the model parameters might lead to over-training resulting in reduced accuracy for measurements that were not in the training set. However, in a practical situation the modalities to be fused can be assumed independent and the probability distributions can be estimated per modality and then be combined.

Score level fusion aims to separate the genuine and impostor scores in a low-dimensional space (the dimension is equal to the number of modalities being fused). As compared to decision level fusion, this approach allows greater freedom in choosing the classification boundary and will therefore lead to better accuracy. Key level fusion can be compared in accuracy to decision level with the limitation that key level fusion inherently uses an AND rule where decision level fusion also allows an OR rule.

In feature vector fusion, two or more modalities are transformed into a single template of which the contributing parts cannot be used separately. In practical applications this can be seen as both an advantage as well as a disadvantage. For example, when fusing fingerprint and finger vein, the inseparability of the fingerprint and finger vein information does not allow to perform a comparison on finger vein only, thus preventing comparison based on latent fingerprints. On the other hand, a multiple fingerprint template requires the use of multiple fingers in all situations which might lead to more inconvenient use cases. Thus, from the usability point of view, it depends on the application which type of fusion would be preferred.

Finally, from the privacy point of view, in feature level fusion the two (or more) modalities will be closely coupled into a single new modality with a higher information content leading to better accuracy (lower $FAR$ at the same $FRR$) and a higher entropy in the binary string. Because the two modalities cannot be compared individually, a FAR attack on a fused template will be much more difficult. The same is true for key level fusion: both inputs for the hash are required to determine if there is a match. The hash inversion attack is related to the sum of the entropy in both modalities and will be similar for feature level and key level fusion. In contrast, a FAR attack on score and decision fused templates will only be a factor two more difficult because every template can be compared individually.

5.3 Conclusion

It can be asserted that in case of feature level and key level fusion the entropy increases linearly and hence the attack effort increases exponentially (in the number of features) since the attack effort is
related to the entropy $H$ according $2^H$. In contrast, score or decision level fusion it increases only linearly (in the number of features). Since the FAR attack is the most important attack, the preferred way of fusion from a privacy perspective would be feature vector level or key level fusion since the attack effort is related to the number of fingers $n$ according $FAR^n$. Taking into consideration the resulting system complexity, it can be seen that key level fusion more easily allows a variable number of modalities (say, 2, 3, or 4 fingers). Thus, overall it can be concluded that key level fusion is the method of choice when implementing BE fusion in application with high security and privacy demands.
6 Assessment of the Classification Performance and Privacy of the BioHASH® SDK

6.1 Introduction

The measurements used in biometric systems are inherently affected by noise and other forms of variability. Apart from the noise present in the measurement system, this is mainly due to varying interaction of the biometric with the acquisition device (sensor). For example, a different pressure on a finger when offering it to a fingerprint sensor will lead to distortion of the fingerprint image. Although a large proportion of the variations can be eliminated when extracting distinguishing features, the generated biometric templates will always contain a certain amount of noise (variability). Consequently, comparison or matching of biometric templates must be treated as a statistical classification process which leads to recognition errors. In biometrics these errors are commonly expressed in terms of the False Accept Rate (FAR) and the False Reject Rate (FRR). The FAR is the probability that the biometric system will incorrectly accept an unauthorized user. Likewise, the FRR is the probability that an authorized user is rejected. Therefore, the lower the FAR and FRR values, the better the (recognition) performance of the biometric system.

Biometric systems can be adjusted towards security or convenience by changing the accept and reject behavior. Usually, high security is associated with a low FAR, whereas high convenience is associated with a low FRR. The possible choices for different system characteristics as well as the quality of the biometric system are commonly depicted in a Detection Error Trade-off curve (or DET curve) which plots the FAR against the FRR. A point on a DET curve presents the combined security and convenience behavior. The point where the FAR is equal to the FRR is called the Equal Error Rate (EER) point and is often used to express the accuracy of a system in a single number. In Figure 17 two example DET curves are given where the solid curve represents better FAR and FRR and hence indicates a better biometric system in term of accuracy.

![Example DET curves of two biometric systems.](image)

Many biometric systems also define a Failure-To-Enroll or FTE rate. An FTE occurs when the biometric system is incapable of handling a certain fingerprint because of, for example, an incomplete scan of the fingerprint.
6.2 Biometric testing

When considering Biometric Encryption (BE) systems, there are two important properties that should be assessed:

- the privacy of the BE templates (this is covered in Deliverable 1.1 of the BioKeyS III project),
- the accuracy of the BE system in terms of FAR, FRR and FTE.

This chapter is concerned with the accuracy of the priv-ID BioHASH® system. Biometric companies and standardization institutes use fingerprint databases to assess the matching accuracy of fingerprint SDKs. An assessment using databases mimics the situation when a fingerprint SDK is used in a real-life situation. The matching accuracy depends on the quality of the images in a database: if a database contains better quality images the matching accuracy of an SDK will be better. Every database should contain at least two images of every finger, one to generate a template and one to perform a comparison. If more images are required to generate a template, more than two images per finger should be available. Furthermore, the fingerprint database should contain sufficient fingers such that statistically relevant results can be obtained.

In this chapter the DET curves of BioHASH® SDK 3.2 will be given and the SDK will be assessed using the public MCYT database and secunet's proprietary FingerQS database. The SDK uses privacy bootstrapping as explained in Section 3.4.2 and will be assessed using the public MCYT database and secunet's proprietary FingerQS database. Moreover, a DET curve for the MCYT database will be given in which only the the Fuzzy Commitment Scheme (see Section 3.3) is used. However, first some prior work is discussed.

6.3 Results for Projekt BioKeyS-PilotDB-Teil1

The goals of the BioKeyS-Multi project was to investigate the feasibility of a Biometric Encryption system. The experience obtained in this project was used in the BioKeyS-PilotDB-Teil1 project where the optimized prototype of Biometric Encryption system was used on a database of 10 individual [BSI 2010]. The setting was based on fusion of six fingers per person. In the assessment, there was no Failure-to-Enroll. In an identification setting, no False Accept events were observed while the FRR was measured to be approximately 10%. Although these results are not statistically relevant [BSI 2010] the reported results show that optimizing the system set-up and the system parameters results in a better behavior of this Biometric Encryption system.

6.4 Assessment for the MCYT database

The public MCYT database (http://atvs.ii.uam.es/databases.jsp) contains good quality fingerprint images of 330 individuals. From all the fingers of these individuals 12 images were collected using a Digital Persona U.are.U 4500 sensor. For the simulations, the first 71 individuals\textsuperscript{10} were chosen using all the fingers of the right hand. The test protocol was as follows:

- The simulated database contains individuals 0-70 (71 individuals)

\textsuperscript{10} The choice for 71 individuals was motivated by simulation time. Other runs show that the performance for all 330 individuals and the first 71 individuals does not show a major difference.
- For each individual the 5 fingers of the right hand were used
- For each finger there are 12 images (thus in total $71 \times 10 \times 12 = 8520$ images)
- Only fingers of the same type are compared (e.g. templates generated from the right index finger are compared only with images from right index fingers)
- Private template generation: out of all 66 possibilities of choosing 2 out of 12 images, 50 possibilities were chosen randomly without duplicates to generate 50 templates per finger
- Genuine comparisons: match the template against all 10 remaining images (thus in total $50 \times 10 \times 71 \times 5 = 177,500$ genuine comparisons)
- Imposter comparisons: match the template against 20 randomly chosen non-genuine images (thus $50 \times 20 \times 71 \times 5 = 355,500$ imposter comparisons)

For the MCYT database, two sets of results are given. The first set uses the privacy bootstrapping approach as explained in Section 3.4.2 and the DET curve is given in Figure 18. During the simulations, no FTE events were observed.

![Figure 18: DET curve for the MCYT database (FTE=0%)](image)

The second set of results is based on the practical implementation of the Fuzzy Commitment Scheme (FCS) as described in Section 3.3. The accumulated results for the Fuzzy Commitment Scheme only based on SDK 3.2 is given in Figure 3. During the simulations, again no FTE events were observed.
Comparing the results in Figure 18 SDK 3.2 and Figure 19 it can be seen that bootstrapping improves the accuracy as compared to an EER of 0.5% for the FCS-only scheme. On the other hand, the current simulations use only two enrollment measurements for generating a private template while the Fuzzy Commitment Scheme benefits from increasing the number of enrollment measurements. Thus, if the application allows, more than two enrollment measurements can be used to enhance the accuracy of the FCS-only approach.

### 6.5 Assessment for the FingerQS databases

The FingerQS databases are secunet proprietary and were collected in 2006 [BSI 2006]. It was the goal to obtain a representative cross-section of the German population both in age groups as well as in gender. In total three different finger print sensors were used: Crossmatch L SCAN 100, Dermalog ZF1 Pre-Release and Sagem MSO 300. For every sensor three images of left and right thumb, index finger, middle finger and ring finger were recorded. The test protocol to obtain the DET curve is as follows:

- Each database contains 1098 individuals
- For each individual, there are 8 fingers (thumb, index, middle, ring for both hands)
- For each finger there are three images (in total $1098 \times 8 \times 3 = 26,352$ images per database)
- Simulations are done per sensor (no cross-sensor matching)
- Private template generation: use the first two (out of three) images per finger

![Figure 19: DET curve for the MCYT database using the Fuzzy Commitment Scheme related to SDK 3.2 (FTE=0%)](image)
6 Assessment of the Classification Performance and Privacy of the BioHASH® SDK

- Genuine comparisons: match the template against the third image (thus 8000 genuine matches per database, $3 \times 8784 = 26,352$ genuine comparisons)

- Imposter comparisons: match template against 20 randomly chosen non-genuine images (thus $20 \times 8784 = 175,680$ imposter comparisons per database, $3 \times 175,680 = 527,040$ imposter comparisons)

The accumulated results for the three sensors are given in Figure 20. During the simulations, no FTE events were observed.

![Figure 20: DET curve for the FingerQS databases (FTE=0%)](image)

### 6.6 Measurable privacy

In Chapter 4, possible attacks on the FCS were discussed to assess the privacy of the FCS. However, a notion that is also commonly used in the privacy assessment of BE systems is the entropy of the binary strings $X$ that are used as a biometric representation in, for example, the FCS (e.g. [Dodi 2004]). Given the entropy (most commonly used notions are the Shannon entropy and the min-entropy) of the strings $X$, properties of privacy leakage can be derived. These methods are very useful in deriving theoretical properties and to build a mathematical foundation for BE methods. Thus, given a large enough database of fingerprint images one could, in theory, determine the entropy.

In practice the situation is more difficult. In order to estimate the entropy, at least an approximation of the distribution of $X$ is required (the maximum of the distribution for the min-entropy and the 'flatness' of the distribution for the Shannon entropy). If we optimistically assume that $X$ consist of, say, 40 bits there are approximate $10^{12}$ possible strings. In order to estimate the entropy, a much larger number of samples is required which far exceeds the number of available human fingers.
Another notion that could be used is to measure the Degrees Of Freedom (DOF) by determining the distribution of the Hamming distance of the strings $X$. Although this does not give an estimate of the entropy, it gives an idea of the amount of freedom that is available in the binary strings $X$. Formulated differently, the DOF indicate how long a bit string with independent and identically distributed (i.i.d.) bits would have to be to obtain the same distribution. In Figure 21, the distribution is given for the Hamming distances of the binary string $X$ for impostor comparisons generated for a random subset of 8 fingers of 330 individuals from the MCYT database [MCYT]. The strings are 257 bit long and each string was generated using two randomly chosen images from the same finger. Using the approach given in [Daug 2003] it can be shown that there are approximately 185 degrees of freedom in the data. This number is also an indication for the dependency of the bits and the amount of redundancy in the bit strings generated by the methods such as described in Section 3.3.3.

![Figure 21: Distribution of the Hamming distances for impostors using the MCYT database](image)

The observed DOF of 185 bits corresponds to the situation where no error-correcting codes are used. In practice this will lead to a system with an extremely low $FAR$ (in the order of $2^{-185}$) and an unacceptably high $FRR$ very close to 100%. The purpose of the FCS is to lower the $FRR$ (while increasing the $FAR$) by storing helper data $W$. Thus, it is interesting to consider the entropy $H(X|W)$ and its relation to the obtained $FAR$ of the system. Some publications state that $|S| \leq -\log_2(FAR)$ which bound is derived assuming perfectly random i.i.d. biometric strings $X$ and an $(n,k,t)$ error-correcting code and by counting the impostor string in the sphere around the target FCS codeword [Buha 2007][Plag 2007]. In practical systems the observed $FAR$ turns out to be significantly higher than this theoretical bound. Thus, in a perfect system there is a direct relation between $|S|$ and the $FAR$ while in practical systems, due to the non-perfect strings $X$, $|S|$ is turns out to be slightly larger than $-\log_2(FAR)$ as is the case with the BioHASH® SDK.

The discrepancy between $|S|$ and $-\log_2(FAR)$ can be explained from the fact that in practical systems, $S$ will not have full entropy or, $H(S|W) \leq |S|$ where $H$ denotes an entropy function. For some special choices of the entropy function (e.g. the average min-entropy [Kort 2008]), it has been shown that $H_\infty(S|W) \leq -\log_2(FAR)$ which holds for arbitrary distributions of the biometric strings $X$. Moreover, it is expected that similar bounds will hold for any BE system. The fact that the length of the key is limited by the $FAR$ motivates the use of biometric fusion which can be used to further enhance the entropy in $S$ (see Chapter 7 for more details on fusion).


6.7 Conclusions

This chapter dealt with the assessment of the practical classification performance of BioHASH® SDK 3.2. The SDK was tested on the public MCYT database and on the proprietary FingerQS database. The assessment shows that the FingerQS databases are more demanding than the MCYT database but the SDK has a good accuracy on both databases. The quality of the binary biometric strings concerning privacy was assessed by simulating the Degrees-of-Freedom.
7 Assessment of the Classification Performance and Privacy using Two Finger Fusion

7.1 Introduction

This chapter describes the properties of the BioHASH® fingerprint system where fusion is applied to two fingers based on key level fusion (KLF). In order to assess the accuracy, the same principles are used as explained in Chapter 6. In that view, this chapter is organized as follows. Section 7.2 presents an overview of key level fusion. The motivation for choosing this particular type of fusion is given and in Section 7.3 it is explained how a DET curve is constructed. In Section 7.4, the practical classification results of the fused system are given along with the databases and protocols that were used to generate the curves. Section 7.7 contains a discussion about the privacy properties of the fused system. Finally, conclusions will be drawn for the BioHASH® KLF approach.

7.2 The key level fusion approach

In this section we briefly repeat the approach and the motivation of using key level fusion in the context of Biometric Encryption methods. For a more elaborate overview of fusion methods, please refer to Chapter 5.

The use of BE methods give rise to a fusion method that is not possible in traditional biometrics. In BE methods, a biometric template takes the form \((AD, PI)\) where \(AD\) is the Auxiliary Data and \(PI\) is the Pseudonymous-Identifier (see also Section 3.3.6). \(AD\) in essence contains variability information and is used during verification to compensate for the variability in a live biometric measurement. The \(PI\) (Pseudonymous-Identifier) is the hashed value \(h(S)\) of a stable secret quantity \(S\) (also referred to as a key) which is chosen during enrollment and regenerated during a verification when a genuine finger is offered. The one-way property of the hash function \(h\) is the mechanism protecting the biometric information that was used to generate the template \((AD, PI)\).

The special structure of a BE template opens the possibility to apply key level fusion. Suppose that two modalities are given that generate keys \(S_1\) and \(S_2\), respectively. Key level fusion (KLF) then consists of storing \(h(S_1 | S_2)\). In Figure 22 this is worked out in more detail for two modalities where

![Figure 22: Overview of key level fusion (KLF)](image-url)
the fused private template is represented as \((AD_1, AD_2, h(S_1|S_2))\). The quantities \(Y_1\) and \(Y_2\) represent live biometric measurements while \(B\) is a binary REJECT/ACCEPT decision.

In BE systems it is important that the entropy of the secret quantity \(S\) is high. Not only makes this an exhaustive search to invert the hash \(h\) more difficult, it also increases the entropy of a cryptographic key that is derived from biometric information. KLF of BE templates has the property that for independent modalities, the entropy in the fused template is the sum of the entropies of all the fused modalities such that the total entropy in the argument of the hash function \(h\) increases significantly. Furthermore, KLF has some additional benefits. In KLF, the modalities that were used to generate the fused template cannot be compared individually because all inputs for the hash are required to determine if there is a match. This makes a FAR attack on a fused template much more difficult. It can be asserted that in case of KLF the attack effort increases approximately exponentially (in the number of fingers). Finally, taking into consideration the resulting complexity of a biometric system implementing fusion, it can be seen that KLF more easily allows a variable number of modalities (say, 2, 3, or 4 fingers). Thus, overall it can be concluded that KLF is the method of choice when implementing BE fusion in application with high security and privacy demands.

### 7.3 Generating DET curves for key level fusion

It can be seen from the explanation in Section 7.2 that KLF is in essence a form of AND fusion: an ACCEPT decision is obtained only if all modalities lead to an ACCEPT. In traditional biometric systems, the ACCEPT/REJECT decision of a single modality is often obtained by applying a threshold \(T\) to a difference score obtained from comparing the stored template with a live biometric measurement. In these systems the thresholds of the individual modalities can be adapted during operation to change the ACCEPT/REJECT behavior leading to a different operating point on the DET curve of the fused system.

![Figure 23: Key level fusion with variable threshold](image)

In BE systems, the ACCEPT/REJECT behavior is determined by the error-correcting code that was chosen during template generation. The threshold \(T\) is related to the number of errors \(t\) the error-correcting code can correct. Hence, the threshold \(T\) is an intrinsic part of the BE template \((AD, PI)\) and cannot modified after the BE template is generated. Thus, conceptually the threshold \(T\) is 'built into' the template and the ACCEPT/REJECT behavior of a KLF system corresponds to a single point on the DET curve.
However, it is still possible to generate a DET curve for BE based on KLF by (conceptually) combining BE templates with a different built-in threshold T. Figure 23 gives a graphical representation of AND fusion for two modalities. The horizontal and vertical axis represent the difference score of the first and second modality, respectively, where a score of 0 means identical templates and a score of 100 means maximum different templates. The variable T is a threshold such that an ACCEPT is generated if a difference score is less than T. For AND fusion, an ACCEPT is generated only if the similarity scores of both modalities are less than T which is indicated in Figure 23 by the light gray area bounded by the solid black lines indicating the AND fusion classification boundary defined by the threshold T. Thus, all score pairs in the light gray area lead to an ACCEPT. To further illustrate the concept, the green dots represent genuine comparison score pairs and the red crosses represent imposter score pairs. Consequently, all the green dots outside the light gray area result in a false reject while all the red crosses inside the light gray area result in a false accept. By increasing T (i.e., on template generation choosing an error-correcting code that can correct more errors), the FRR will decrease and the FAR will increase. This way it is possible to derive the key level fusion FRR and FAR curves as function of the threshold T.

The FARs and the FRRs of the individual modalities (i.e., FAR1, FAR2, FRR1 and FRR2) can be related to the fused false accept and false reject rates FARtot and FRRtot, respectively. In general we have FRRtot = FRR1 + FRR2 - FRR1 * FRR2 where the latter term is zero when both events are mutually exclusive. Likewise we have that FARtot = FAR1 * FAR2 when both events are independent.

It is possible to give some extensions to the scheme given above. Firstly, it is not necessary that both modalities use the same built-in threshold T. This is especially important when both modalities are of a different type (e.g. fusion of a face and a finger). When both modalities are of the same type (e.g. fusion of two fingers) it is usually best to choose the built-in thresholds to be equal11. Furthermore, some BE systems return the difference score when an ACCEPT is obtained. This allows modification of the classification boundary within the gray rectangle in Figure 23.

The results of testing KLF on a public database are given in the following section.

### 7.4 Biometric testing and results

In traditional biometrics systems, fusion is used to improve the overall accuracy. In BE systems, an additional motivation for using fusion is to increase the privacy of the stored biometric information or to derive keys with higher entropy. The increase in accuracy as well as the increase of the entropy depends on the type of fusion that is used (e.g. feature level, score level, decision level, key level). In this work, KLF was chosen and the motivations were given in Section 7.2. Since KLF is a form of AND level fusion, it is to be expected that the FRR increases while the FAR decreases. Therefore it is important to assess the overall accuracy of the BioHASH® system based on KLF. In Section 7.6 the results are given in the form of DET curves for two fingers while Section 7.7 discusses the increase in entropy.

### 7.5 Testing protocol

In this chapter, fusion of the two fingers is studied. Two cases will be studied independently:
- fusion of two index fingers;

11 This is the approach that will be used in the following sections.
7 Assessment of the Classification Performance and Privacy using Two Finger Fusion

- fusion of two middle fingers.

For each of the two cases, the MCYT database was used (http://atvs.ii.uam.es/databases.jsp). This database contains good quality fingerprint images of 330 individuals. From all the fingers of these individuals, 12 images were collected using a Digital Persona U.are.U 4500 sensor. Finger 0 and Finger 5 in the MCYT database represent the right and left index finger, respectively, while Finger 1 and Finger 6 represent the right and left middle finger.

The test protocols for the index fingers were identical to the test protocols for the middle fingers. Therefore, below we will only explain the protocols for the fusion of the two index fingers.

As mentioned above, for each individual in the database there are 12 images of the right index finger and 12 images for the left index finger. For the test protocol, two corresponding images are collected in a set which is illustrated in Figure 24. For a person i, the j-th set contains the j-th image of Finger 0 and the j-th image of Finger 5. For the tests, BioHASH® SDK 4.0 was used in the following protocol (Note that generating a template of a single modality requires two enrollment images. Likewise, two sets are required to generate a fused template.)

- Private template generation (two finger fusion): for each individual, out of all 66 possibilities of choosing 2 out of 12 sets, 50 possibilities were chosen randomly without duplicates to generate 50 fused private templates (\(AD_1, AD_2, h(S_1|S_2)\)) per individual (see Section 7.2 for the definition of a fused private template).

- Genuine comparisons: for each individual, the 50 fused private templates were matched against the 10 remaining sets of the same individual (note that 2 of the 12 available sets were used to generate the private template such that there are 10 sets remaining for genuine comparisons). This results in 330×50×10=165,000 genuine comparisons per database.

- Imposter comparisons: for each individual, the 50 fused private templates were matched against 20 sets where every set is chosen randomly from all the sets of the remaining individuals. This results in 330×50×20=330,000 imposter comparisons per database.

For all the comparisons, the difference scores were collected\(^{12}\) leading to an estimated genuine and imposter distribution. From these distributions, DET, FAR and FRR curves can be computed which will be given in the following section.

\(^{12}\) The difference score \(S_{\text{diff}}\) is computed from the similarity score \(S_{\text{sim}}\) produced by the BioHASH® SDK. This similarity score is normalized on the interval \([0,100]\). The relation between the two scores is \(S_{\text{diff}}=100-S_{\text{sim}}\) such that \(S_{\text{diff}}\) is also in the interval \([0,100]\).
7.6 Classification results

The results of the simulations for the individual fingers and for fusion are given in Figure 25 and Figure 26. During the simulation, no Failures-to-Enrol (FTEs) were observed. When inspecting the figures several observations can be made.

Figure 25 and Figure 26 indicate that the accuracy for the index fingers is better than the accuracy of the middle finger. Secondly, it can be seen that in case of fusion, the DET curve moves up (leading to a higher FRR) and towards the left (leading to a lower FAR). This is to be expected because both fingers must result in a match in order to get a match for the KLF template. However, for good quality fingerprints like in the MCYT database, an FRR of approximately 0.5% for two finger KLF is still acceptable in most practical situations.

The DET curves do not show FAR values below $3 \times 10^{-6}$. This can be attributed to the limited number of 330,000 imposter comparison (see Section 7.5). In order to get a more accurate representation of the DET curve for low FAR values, a larger database is required and/or more imposter comparisons. How to obtain accurate FAR results for low FAR values is a point of further research. In any case, considering the slope of the fused DET curve in Figure 25 and Figure 26, it is possible to choose a low FAR operating point with an acceptable FRR.

It can further be noted that the FRR of KLF is less than the sum of the individual FRR values indicating that, as to be expected, the FRR events are not mutually exclusive. A more interesting observation is that $\text{FAR}_{\text{KLF 0 and 5}} > \text{FAR}_0 \cdot \text{FAR}_5$, indicating that the FAR events are dependent (likewise for Finger 1 and Finger 6). Thus, a false accept for the first finger increases the probability on a false accept for the second finger and vice versa. This can be studied in more detail using Figures 27 through 32 which depict the FAR and FRR curves for fingers 0, 5, 1 and 6 separately, as well as the FAR and FRR curves for KLF for Fingers 0 and 5 as well as Fingers 1 and 6. In Table 1 results are collected illustrating that $\text{FAR}_{\text{KLF 0 and 5}} > \text{FAR}_0 \cdot \text{FAR}_5$.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>$\text{FAR}_0 \cdot \text{FAR}_5$</th>
<th>$\text{FAR}_{\text{KLF 0 and 5}}$</th>
<th>$\text{FAR}_{\text{KLF 0 and 5}} / (\text{FAR}_0 \cdot \text{FAR}_5)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>98</td>
<td>$9.1 \times 10^{-6}$</td>
<td>$3 \times 10^{-6}$</td>
<td>33.1</td>
</tr>
<tr>
<td>98.5</td>
<td>$5.0 \times 10^{-7}$</td>
<td>$3 \times 10^{-5}$</td>
<td>59.7</td>
</tr>
<tr>
<td>99</td>
<td>$2.5 \times 10^{-6}$</td>
<td>$1 \times 10^{-4}$</td>
<td>39.4</td>
</tr>
<tr>
<td>99.5</td>
<td>$1.3 \times 10^{-5}$</td>
<td>$2 \times 10^{-4}$</td>
<td>14.9</td>
</tr>
<tr>
<td>100</td>
<td>$4.9 \times 10^{-5}$</td>
<td>$3 \times 10^{-4}$</td>
<td>6.1</td>
</tr>
</tbody>
</table>

Considering the test protocol described in Section 7.5, there are two possible explanations for this effect. Firstly, it can be expected that primary pattern features of different fingers of the same individual are somewhat dependent. An illustrative example is the ridge frequency which is probably more similar between fingers of the same individual than between fingers of different individuals. Secondly, it is to be expected that secondary image properties (such as wetness or dryness of a print) are correlated between the images of different fingers of the same person. Therefore, if a match occurs due to

---

For the curves of a single finger (no fusion), the same protocol was used as described in Section 7.5 but every set contained only a single image.
secondary image properties it is expected that the matching results are dependent (the same effect will probably occur in minutiae matchers). Additional experiments, where each imposter sets (see Section 7.5) contained images from different individuals, showed that $FAR_{\text{KL}} = FAR_1 \times FAR_2$ which strongly suggests that indeed the matching results of two fingers of the same individual are dependent.
7.7 Privacy considerations

As mentioned in Section 7.2, one of the motivation for using fusion in BE systems is to increase the entropy in the argument of the hash $h$ or, stated differently, to increase the entropy in the key $S$ derived from the biometric. If we combine the general assumption that the fingerprints of an individual are independent with the notion for KLF, the entropy of the individual modalities can be added to obtain the total entropy of the fused BE template. Thus if a single finger generates an entropy of $q$ bits, then $i$ fingers will generate an entropy of $q_i$ bits.

However, it was noted above that in a practical situation, the fingers are dependent. Consequently, the total entropy will be less than the sum of the individual entropies. We will call this effect entropy-increase reduction. If we assume that the entropy $H$ after observation of the template is related to the probability $p(FA)$ on a false accept ($FA$)\(^{14}\) according to $H(S)=-\alpha \log_2(p(FA))$, were $\alpha$ is a constant and $S$ is the stable quantity or key (see also Section 7.2), we can make the following derivation for two fingers\(^{15}\).

For the (Shannon or min-) entropy in general we have

$$H(S_1,S_2)=H(S_1)+H(S_2|S_1)$$

$$=H(S_1)+H(S_2)+(H(S_2|S_1)-H(S_2)).$$

---

14 Note that $p(FA)$ is equal to $FAR$ but to allow for a more regular mathematical notation, we use $p(FA)$.

15 Joint work with Aweke Lemma.
The term \( H(S_2|S_1) - H(S_2) \) can be interpreted the entropy-increase reduction \( (E_1) \) due to the dependency of \( S_1 \) and \( S_2 \). Similarly, in terms of probabilities we have

\[
p(FA_1, FA_2) = p(FA_1)p(FA_2|FA_1)
= p(FA_1)p(FA_2)(p(FA_2|FA_1)/p(FA_2))
\]

such that the entropy-increase reduction is

\[
p(FA_2|FA_1)/p(FA_2) = p(FA_1, FA_2)/p(FA_1)p(FA_2).
\]

If the cause of the dependency is indeed as suggested above, it can be expected that the entropy-increase reduction will mainly occur when a second finger is added. Adding a third, fourth, etc. finger will not lead to additional reduction. Thus, roughly speaking, if the entropy of a single finger is \( q \), the total entropy when fusing \( i \) fingers would be \( q + (i-1)(q-E_1) \). When different types of modalities are used (such as finger print and finger vein) it is expected that entropy in the individual modalities can simply be added. These are, however, points of further research.

Simulations on the FAR data of Finger 0 and Finger 5 indicate a practical entropy loss as function of the operating point (threshold) \( T \) as given in Figure 33 showing that there is an entropy-increase reduction between approximately 2 and 6 bits. These quantitative results should, however, be interpreted carefully. It is important to notice that the estimate of the reduction is based on the tail of the imposter distribution. Especially in this experiment it can be seen that the tail is estimated by only a few measurements and hence the estimate of the shape of the tail will not be accurate resulting in an inaccurate estimate of the entropy-increase reduction in Figure 33. Nevertheless, the explanation and the data given in this section might serve as a point for further research into the entropy loss when fusing modalities.

Although above we only discussed the results for KLF (or AND decision level fusion) it is to be expected that feature level fusion (which also leads to higher entropy keys) will lead to similar entropy loss. The underlying motivation is that the entropy loss seems to result from the dependency of the underlying images rather than from the specific fusion method. This can also be considered as a point of further research.
7.8 Conclusions

This chapter gave an overview of the practical classification performance and a privacy assessment of the BioHASH® system using key level fusion (KLF). The motivation to use KLF is that it is the most convenient fusion approach that results in a significant increase in the entropy of the key derived from the biometric.

The assessment was done using the public MCYT database where DET curves were generated for the fusion of the two index fingers of the individuals as well as for the two middle fingers. The DET curves indicate that the FRR increases and the FAR decreases. For the FRR it was observed that approximately $FRR_{KLF} = FRR_1 + FRR_2$. For good quality images, however, the FRR can easily be kept below 1% at a significantly decreased FAR. An interesting observation was that for imposter comparisons $FAR_{KLF} > FAR_1 * FAR_2$, indicating a dependency between two fingers such that the FAR decreases less than exponentially in the number of fingers. Still, KLF leads to a fast decrease of the FAR at a limited increase for the FRR.

Based on this observation on the FAR, we can conclude that the dependency of fingers also leads to a less than linear increase in the entropy as a function of the number of fingers. A simplified model was given estimating the entropy loss for an added finger.

Summarizing we have that KLF is an excellent fusion approach. Although KLF leads to an increase of the FRR, the overall classification performance improves significantly. Moreover, KLF leads to an almost linear increase in of the entropy in the key derived from a KLF BE template.
8 A Practical Experience using Two Finger Fusion

8.1 Introduction

When using new biometric technologies such as methods that protect the privacy of biometric information, it is important to learn about how users perceive the system as these technologies might cause a different experience for the user. Most notably, accuracy, template extraction times and matching speeds might be different from traditional biometric technologies. Furthermore it is important to assess the practical classification performance as obtained from real users in a practical situation as opposed to the performance figures obtained from standard databases (such as the MCYT databases).

For this purpose, a practical biometric system set-up was built. Using this set-up, 25 individuals were enrolled and verified using KLF of the right and left index finger. This chapter describes in Section 8.2 the experimental set-up that was used for the participants while Section 8.3 gives details about the software behavior. The collected experiences are discussed in Section 8.4 and, based on the collected finger print images and the measurements done in the practical set-up, the practical and detailed classification performance is discussed in Section 8.5 and Section 8.6, respectively.

8.2 Software description

One of the goals of the experiment was to collect user experiences and feedback from a small group of users concerning the properties of the underlying template extraction and matching software of priv-ID. Especially the experiences regarding the timing and the accuracy of the underlying software was important. The practical set-up used key level fusion which was motivated by the fact that key level fusion as opposed to, for example, score level fusion, allows to derive keys with a higher entropy. Key level fusion requires more than one finger both for enrollment and verification. In the practical set-up the choice was made to use key level fusion based on the two index fingers.

![Figure 34: Overview of the GUI](image-url)
Besides modifying the underlying extraction and matching algorithms to allow for key-level fusion, a new Graphical User Interface (GUI) was designed. The design of the GUI was intentionally kept simple with as few graphic enhancements as possible to prevent the participants being distracted from the actual software behavior. The GUI is given in Figure 34.

On the left is the Enrollment area. The top-left two rectangles display the two enrollment images from the right index finger while the bottom-left rectangles display the two enrollment images from the left index finger. On the right is the Verification area where the top-right rectangle displays the verification images of the right index finger and the bottom-right rectangle of the left index finger. The middle part displays the matching result indicated by an icon showing a question mark (no match done yet), a check mark (match) or a cross (no match) (see Figure 35). It further contains statistical information on the number of verification attempts required to obtain a match and it indicates an anonymous subject (participant) identifier.

Part of the user experience of a practical set-up is determined by the type of fingerprint sensor that is used. The experiments used two different fingerprint sensors. In order to easily allow for two different fingerprint scanners, two set-ups were used as depicted in Figure 36, each containing an identical laptop but a different fingerprint scanner.
Throughout the experiments, the total set-up was operated by a priv-ID employee. The required software actions of the priv-ID operator were limited to pushing the “Enter” button on the keyboard to move from Enrollment to Verification mode and vice versa. Each participant that took part in the test went through the following steps.

1. An introduction was given by a priv-ID employee explaining the goal of the set-up;
2. The participant collected four enrollment images in total by putting his right index finger on Sensor A twice followed by putting his left index finger on Sensor A. After four images were collected, the operator pushed the “Enter” button to move to Verification mode which automatically erases the enrollment images from the GUI.
3. Based on the four collected images, the software derived a fused template of the right and left index finger based on key level fusion;
4. The participant put his right index finger on Sensor A followed by his left index finger. The software performed a match using key level fusion presented the matching result in the GUI. This process of collecting two verification images and performing a match is repeated until a correct match is obtained. The number of attempts is automatically recorded and saved as a statistical item and displayed at the bottom of the GUI. The four enrollment images and the verification images of a successful verification are saved to disk by the operator pushing the “Enter” button. This puts the software back into Enrollment mode and the Subject ID number at the bottom of the GUI is automatically increased;
5. Step 2. to 4. are repeated for Sensor B;
6. Each participant is requested to fill out a questionnaire on the experiences with the set-up.

### 8.3 Software details

In order to interpret the results and to be able to estimate the performance of the set-up when it runs on systems other that the system used in the experiments, the technical details of the set-up are given as follows.

- **Laptop details:**
  - HP Compaq 76730b, running Windows(R) XP SP3
  - Intel(R)Core(TM)2 Duo CPU, P8600@2.4GHz, 790MHz, 1.93 GB RAM
- **Sensor types:**
  - Sensor A: digitalPersona U.are.U 4500 Fingerprint Reader
  - Sensor B: Futronic Finger Print Scanner model FS 80

The fingerprint scanners were intentionally chosen so as to give the participants two different experiences. Sensor A has a soft silicon layer on top of the sensing area while the sensing area of Sensor B is a flat, glass-like surface. Earlier experiments show that most individuals prefer Sensor A over Sensor B which is confirmed by this experiment (See Appendix 2). For the used hardware, the approximate timing results are given in Table 2.
Table 2: Approximate processing and operating times

<table>
<thead>
<tr>
<th>Processing times for a single image</th>
<th>Practical operating times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image capture</td>
<td>300ms</td>
</tr>
<tr>
<td>Feature extraction</td>
<td>500ms</td>
</tr>
<tr>
<td>Matching</td>
<td>2ms</td>
</tr>
</tbody>
</table>

Finally, an important factor influencing the software experience for a user is the False Reject Rate (FRR) because it determines the probability for a genuine user to be rejected by the system. Thus, a low FRR will lead to a better user appreciation. On the other hand, in many cases, the owner of the biometric system would like to have a low False Accept Rate (FAR) to reduce the probability of an illegitimate user getting physical or logical access. The trade-off between FRR and FAR is determined by the setting of the internal difference score threshold T in the software such that all comparisons with a difference score lower than T will be accepted. Consequently, increasing T will increase the overall acceptance and thus increase the FAR and decrease the FRR. The difference score can be scaled to take values between 0 (two fingers are identical) and 100 (two fingers are maximum different).

The procedure to set the threshold for the practical set-up was as follows. The underlying priv-ID matching software was applied to the MCYT database (http://atvs.ii.uam.es/databases.jsp). Based on two finger key level fusion (Finger 0 and Finger 5), the FRR and FAR curves were plotted as a function of the similarity threshold T (see Figure 37). The value of the threshold T where the FAR became zero, as indicated by the location of the arrow in Figure 37, was chosen as the operating point of the software before it was rolled out on the two laptops and corresponds to T=98.

8.4 Questionnaire results

As mentioned in Section 8.2, an important goal of this work is to collect user experiences with biometric software, especially when using key level fusion which might lead to an increased false reject rate. In order to assess the participants' experience using the software set-up, a questionnaire...
was prepared containing 14 questions: 6 questions to assess the technical background of the participant and his/her experience with biometrics in general, and 8 questions to evaluate how the participant experienced the software set-up. The questionnaire was made available both in English and in Dutch. Both versions are provided in Appendix 1.

The details of the filled-out questionnaires are collected in Appendix 2. A summary of the questionnaire results is as follows.

The background of participants is mostly male with an age between 25 and 65 years old (Question 1 and 2). They are used to work with computers (Question 3) but do not have regular experience with biometric systems (Question 5). Most of the participants are, to a certain extent, concerned with abuse of the biometric information stored in biometric systems (Question 6).

In the set-up, the majority of the participants prefer Sensor A over Sensor B (Question 8). In general the software is pleasant to use and the delay caused by the capturing and the processing of an image is short enough to have a positive experience (Question 9 to 12). According to all participants the accuracy of the matching was good (Question 13).

### 8.5 Practical classification results

As mentioned in Section 8.2, a second important goal of this work is to assess the practical classification performance. During the experiment, the fingerprint images of the participants were stored in two small databases (one per fingerprint sensor) that can be used to assess the classification performance off-line. The detailed results of the classification performance using the images are given in Section 8.6.

A second, more practically oriented approach is to count the number of times a genuine verification failed. As described in Section 8.2 the software kept track of the number of verification attempts.
that were required to achieve a correct genuine verification. The results are collected in Figure 38 and Figure 39.

From this information, the practical False Reject Rate (FRR) appears to be approximately 25% for Sensor A and 4% for Sensor B. However, these results require some further discussion.

First, the experiment was not completely controlled: although the experiments were guided by a priv-ID employee, the guidance was lenient and the participants were not forced to give good quality finger print images.

Second, a population of 25 individuals is not statistically relevant and therefore the obtained FRR and FAR numbers must be interpreted carefully.

Third, from earlier experiences and from the answers given to Question 8 in the questionnaire, it follows that Sensor A is considered to be a more convenient scanner. One of the reasons is that Sensor A more easily captures an image. In fact, the capture process is so sensitive that it regularly happens that a latent print on the scanner surface triggers a new capture cycle. This new capture cycle is counted by the software as a verification attempt. Therefore the information in Figure 38 does not correctly represent the behavior of the underlying matching software.

Fourth, one individual needed five attempts to be correctly accepted. However, it was observed that this participant first intentionally forced a number of rejects to check if the software indeed sometimes rejected a match. This behavior was permitted (but noted) to retain the participant's trust in the practical set-up. Therefore the rightmost columns in Figure 38 must be discarded.

Summarizing, the results of this test shows that Sensor B gives a better representation of the practical FRR at a value of 4% for this relatively small population. Detailed classification results for the collected data are given in Section 8.6. For key level simulation results on larger databases, please refer to Chapter 7.

### 8.6 Detailed classification results

As explained in Chapter 5 and Chapter 7, key level fusion is in essence a form of decision level fusion (AND fusion) where a match is obtained only if the difference scores of all fused modalities are below a preset threshold $T$. In Chapter 7, an explanation is given how to generate DET, FAR and FRR curves. In this section, the same ideas, concepts and test protocols will be used to generate the classification results for the data collected in the practical experiment.

In the experiment, for both sensors, the six images per user (four enrollment images and two verification images) of all 25 participants were saved resulting a database for each sensor. Thus, each database contains 25 individuals with 3 images of the right index finger and 3 images for the left index finger as shown in Figure 40.
left index finger (i.e., 150 images per database). For each database (sensor) separately, the
classification performance was determined according to the protocol explained below.

As mentioned above, for each individual in the database there are 3 images of the right index finger
and 3 images for the left index finger. For the test protocol, two corresponding images are collected
in a set. This is illustrated in Figure 40. For individual i, the j-th set contains the j-th image of the
right index finger and the j-th image of left index finger. Every individual has three sets (per
sensor).

For the tests, BioHASH® SDK 4.0 was used in the following protocol. Note that generating a
template of a single modality requires two enrollment images. Likewise, two sets are required to
generate a fused template.

- Private template generation (two finger fusion): for each individual, all 3 possibilities of
  choosing 2 out of 3 sets were chosen to generate 3 fused private templates \((AD_1, AD_2, h(S_1|S_2))\)
  per individual (see Section 7.2 for the definition of a fused private template).
- Genuine comparisons: for each individual, the 3 fused private templates were matched against
  the remaining set of the same individual (note that 2 of the 3 available sets were used to generate
  the private template such that there is 1 sets remaining for a genuine comparison). This results in
  \(25 \times 3 = 75\) genuine comparisons per database.
- Imposter comparisons: for each individual, the 3 fused private templates were matched against 2
  sets where every set is chosen randomly from all the sets of the remaining individuals. This
  results in \(25 \times 3 \times 2 = 150\) imposter comparisons per database.

In Figure 41 through Figure 45, the classification performance is given per sensor in the form of the
\(FRR\) and \(FAR\) curve as a function of the threshold \(T\). It can be seen that in this experiment, there is
a perfect separation between genuine and imposter distributions. Note that in the simulations, all
the imposter comparisons lead to a 'no match' result with a difference score of 100 which results in a
vertical line for the \(FAR\) curve. As to be expected in the case of key level fusion, for a chosen
threshold, the \(FRR\) increases as compared to a single modality. However, the threshold of the
system can still be chosen to have both a low \(FRR\) and a low \(FAR\).

If we compare Figure 41 through Figure 45 with the corresponding Figure 27 through Figure 32, it
can be seen that for the \(FRR\) curves, at higher values of the threshold \(T\), the curves in Figure 41
through Figure 45 are less smooth due to a lower number of genuine comparisons. However, the
overall shape of the \(FRR\) curves are similar. Furthermore, due to the small population and the
limited number of imposter comparisons, no false accept events were observed during the
experiments resulting a vertical line for the \(FAR\) curve. In contrast, in Chapter 7 a large number of
imposter comparisons were performed using a larger population such that the \(FAR\) curve has non-
zero width.

---

16 For the curves of a single finger (no fusion), every set contains only a single image.
A Practical Experience using Two Finger Fusion

Figure 41: FAR and FRR curves for sensor A (left index finger)

Figure 42: FAR and FRR curves for sensor A (right index finger)

Figure 43: FAR and FRR curves for sensor A (fusion of left and right index finger)

Figure 44: FAR and FRR curves for sensor B (left index finger)

Figure 46: FAR and FRR curves for sensor B (right index finger)

Figure 45: FAR and FRR curves for sensor B (fusion of left and right index finger)
8 A Practical Experience using Two Finger Fusion

8.7 Conclusion

The purpose of this work was to obtain insight in the practical user experiences with systems based on key level fusion where multiple finger are used both for enrollment and verification. The underlying motivation to use key level fusion as opposed to, for example, score level fusion, is that key level fusion allows deriving keys with a higher entropy. On the other hand, it will lead to a higher FRR which might lead to a lower perceived convenience level for the user of the system.

In order to collect practical user experiences, new software was developed containing a Graphical User Interface (GUI) with an enrollment and verification section. The underlying extraction and matching algorithms were based on key-level fusion. All the fingerprint images were saved to allow off-line simulation of FRR and FAR curves.

The experiments lead to the following conclusions:

- The participant experience the software set-up as pleasant despite the fact that they have to use two fingers in a situation where a higher FRR was to be expected;

- The off-line simulations based on the collected images show a perfect separation of genuine and imposter distribution making it possible to operate the system at low FRR values and thus at a high convenience level while the FAR can still be kept low.

- Comparing the results on public databases as shown in Chapter 7 with the practical performance assessed in this chapter, it was shown both curves are similar.
Biometric Encryption (BE) methods protect biometric information without the need for long-term secrets such as cryptographic keys. This document presents an overview and the conclusions of the BSI BioKeyS III project which was concerned with a detailed accuracy and privacy assessment of a specific BE method: the Fuzzy Commitment Scheme.

While the methods for assessing the accuracy of biometric products is well-known, the approach for determining the privacy properties of BE systems is not yet well defined. Therefore, this document begins with a definition of the goals of BE systems and presents a framework for the privacy assessment of BE systems.

The first part of the framework consists of defining the privacy requirements and in this project the recommendations of the ISO/IEC JTC1 SC27 2ndCD 24745 standard on Biometric Information Protection are followed.

A second important aspect of the framework is the definition of the so-called attack model which describes the capabilities of an adversary attempting to break the ISO privacy requirements. In defining the attack model, similar notions are used as in defining adversary capabilities in traditional cryptographic systems.

Based on the ISO privacy requirements, a first assessment of the privacy properties of the Fuzzy Commitment Scheme is performed based on a number of known attacks. An important attack in this context is the FAR attack. This attack can be successfully mounted even against a perfectly private biometric system and in that sense, it imposes a lower bound on the privacy properties of BE systems. It is concluded that a FAR attack is always easier to perform than any other high level or low level attack (see Section 4.5.1).

A second notion of the privacy of BE systems is the entropy in the key that can be derived from a biometric feature. Based on publicly available databases, an estimation of the entropy is obtained. The entropy is closely related to the FAR of a biometric systems (see Section 6.6).

A method to increase the entropy is to use fusion of several fingers. The BioKeyS III project introduces a new fusion method, called key level fusion, that significantly increases the entropy in the derived key and inhibits a FAR attack (see Section 7.2). Based on public fingerprint databases, the accuracy of key level fusion of two fingers is determined and it is demonstrated that the FRR increases and the FAR decreases. The increase in the entropy is estimated and it is concluded that the entropy increases slightly less than linearly with the number of fingers due to the dependency between fingerprint patterns of the same individual (see Section 7.7).

The theoretical results and methods were applied to a commercial product. The product that was chosen was the BioHASH® SDK provided by priv-ID B.V.

The classification accuracy of the BioHASH® SDK was assessed using a public and a proprietary database. Furthermore the accuracy was confirmed by practical experiments on a population of 25 individuals.

At the end of the project it can be concluded that Biometric Encryption methods and products have become of age. Based on the BioHASH® SDK it was illustrated that the achievable accuracy of BE systems can be similar to methods that do not offer privacy and that the Fuzzy Commitment Scheme can obtain practical privacy levels, in terms of the entropy of the derived key, that are close to what can be expected from a theoretical viewpoint.
10 Glossary

\(|\cdot|\) number of entries in a variable (e.g. \(|X|=N\) means that the length of \(X\) is \(N\) bits, \(|f|=M\) means that \(f\) has \(M\) entries)

\([\cdot]\) a protected template. For example, for the FCS, \([X]\) represents \((W, h(S))\)

AES Advanced Encryption Standard

AID Application-specific identifier

\(B\) Binary comparison decision

BCH Error correcting code invented by Bose, Ray-Chaudhuri and Hocquenghem

BE Biometric Encryption

BioHASH® Commercial name for the priv-ID BE technology

\(C\) codeword of an ECC

\(C_{FAR}\) effort to collect a single image for a database

COMP comparator

\(D\) Hamming distance between an enrollment binary string \(X\) and verification binary string \(Y\)

DEC ECC decoder

DLL dynamic-link library, an executable file that allows programs to share code

DOF Degrees Of Freedom

ECC Error Correcting Code

\(EER\) Equal Error Rate

\(E_{K}(\cdot)\) encryption with key \(K\)

ENC ECC encoder

\(f\) feature vector representation of a biometric measurement

\((f)_i\) \(i\)-th element of vector \(f\)

FPBE system Fingerprint Biometric Encryption system

FCS Fuzzy Commitment Scheme

FE feature extraction

\(FAR\) False Accept Rate, the probability that the biometric system will incorrectly accept an unauthorized use

\(FRR\) False Reject Rate, the probability that the biometric system will incorrectly rejected an authorized user

\(H\) entropy function

helper data generic term indicating the the part of a BE template that is combined with a live biometric measurement to obtain a noise free template
HID  
hardware identifier

i.i.d.  
independent and identically distributed

ISO  
International Organization for Standardization

$K$  
cryptographic key

MAC  
Message Authentication Code

MAC address  
Media Access Control address

$N$  
number of bits in $X$

$(n,k,t)$ code  
Error correcting code with codewords that have $n$ symbols, $k$ message symbols and can correct up to $t$ symbols. In case of a binary code, a symbol is a single bit

$m$  
number of biometric records in a database

PBT  
Privacy Bootstrapping

PII  
Personally Identifiable Information

PIN  
Personal Identification Number

$p_i(F)$  
overall probability distribution of the feature vectors $f$ (impostor distribution)

$p_g(F)$  
probability distribution the feature vectors $f$ derived from an individual biometric instance (e.g., a finger) (genuine distribution)

PKI  
Public Key Infrastructure

PPB system  
Perfectly Private Biometric system (concept)

PPBE system  
Perfectly Private Biometric Encryption system (concept)

RNG  
Random Number Generator

RSA  
Rivest-Shamir-Adleman public key encryption algorithm

$S$  
secret to be linked to a Biometric Encryption template

$S_c$  
comparison score of 3rd party comparator

SDK  
Software Development Kit

$S_{FAR}$  
time required for a private comparison

SHA  
secure hash algorithm

$S_{\text{hash}}$  
time required for evaluating a hash

$S_{\text{match}}$  
comparison time excluding the hash (i.e., $S_{\text{match}} = S_{FAR} - S_{\text{hash}}$)

SENS  
sensor

STOR  
storage

S/N ratio  
signal-to-noise ratio

$T$  
threshold module with threshold $T$

$T_3$rd party template
$T_D$ threshold for $D$

$W$ helper data $C \oplus X$

$X$ binary string representation of an enrollment biometric measurement

$[X]$ shorthand notation for protected binary string $X$ (private template)

$Y$ binary string representation of an verification biometric measurement
Appendix 1 Questionnaires

In order to assess the participants' experience using the software set-up, a questionnaire was prepared containing 14 questions, 6 questions to assess the technical background of the participant and his experience with biometrics in general, and 8 questions to evaluate how the participant experienced the software set-up. The questionnaire was made available both in English and in Dutch. For completeness, both versions are provided in this Appendix.
Questionnaire

Background
7. Gender
   □ Male
   □ Female

8. What is your age group?
   □ <25
   □ 25-40
   □ 40-65
   □ >65

9. Please indicate your experience with software in general
   □ I do not work with computers
   □ I use email several days per week
   □ I use Office applications several days per week (Word, Powerpoint, Excel, etc.)
   □ I use other PC applications several days per week

10. Please indicate which one or more of the following biometric technologies you have used before.
    □ Fingerprint
    □ Voice
    □ Palm
    □ Face
    □ Iris
    □ None

11. How often do you use biometric technology
    □ Every day
    □ Once per week
    □ Once per month
    □ Seldom

12. In a biometric system like you just used, your fingerprint is stored in the system. Are you concerned about abuse of the stored biometric information?
    □ Not at all
    □ A bit
    □ Very much
    □ Not when it is properly protected and the image cannot be abused
Appendix 1 Questionnaires

Experience with the equipment

13. The explanation that was given on how to use the equipment was clear
   - Agree
   - Neutral
   - Disagree
   - Because

14. Which sensor has your preference?
   - Sensor A
   - Sensor B
   - Because

15. General impression is that the software is pleasant to use.
   - Agree
   - Neutral
   - Disagree
   - Because

16. The software is easy to use
   - Agree
   - Neutral
   - Disagree
   - Because

17. The software reacts intuitively
   - Agree
   - Neutral
   - Disagree
   - Because

18. The software reacts
   - Too slowly
   - About right
   - Too quickly
   - Because

19. How do you rate the accuracy of the software?
   - Good
   - Medium
   - Bad
   - Because

14. Do you have any further comments on the use of biometrics or the software you just used?
Enquête

Achtergrond

1. Wat is uw geslacht?
   - Man
   - Vrouw

2. Wat is uw leeftijdscategorie?
   - <25
   - 25-40
   - 40-65
   - >65

3. Hoeveel ervaring heeft u met computers en software in het algemeen?
   - Ik werk niet met computers
   - Ik gebruik email meerdere dagen per week
   - Ik gebruik Microsoft Office meerdere dagen per week (Word, Powerpoint, Excel, etc.)
   - Ik gebruik andere PC applicaties meerdere dagen per week

4. Welke biometrische technologieën heeft u wel eens gebruikt?
   - Vingerafdrukken
   - Stem
   - Handpalm
   - Gezicht
   - Iris
   - Geen van deze technologieën

5. Hoe vaak gebruikt u biometrische technologie?
   - Elke dag
   - Eenmaal per week
   - Eenmaal per maand
   - Bijna nooit

6. In een systeem zoals u dat zojuist heeft gebruikt wordt uw vingerafdruk opgeslagen. Maakt u zich zorgen over mogelijk misbruik van de opgeslagen informatie in dit soort systemen?
   - Helemaal niet
   - Een beetje
   - Veel
   - Niet zolang de informatie goed is beschermd en misbruik van de informatie daarmee onmogelijk wordt.
Ervaringen met de gebruikte opstelling

7. De uitleg over het gebruik van de opstelling was duidelijk.
   - Eens
   - Neutraal
   - Oneens
   - Omdat..................................................

8. Welke sensor heeft uw voorkeur?
   - Sensor A
   - Sensor B
   - Omdat..................................................

9. De algemene indruk is dat de opstelling prettig is in het gebruik.
   - Eens
   - Neutraal
   - Oneens
   - Omdat..................................................

10. De software is eenvoudig te gebruiken.
    - Eens
    - Neutraal
    - Oneens
    - Omdat..................................................

11. De software reageert op een intuitieve manier.
    - Eens
    - Neutraal
    - Oneens
    - Omdat..................................................

12. De snelheid waarmee de software reageert is
    - Te langzaam
    - Goed
    - Te snel
    - Omdat..................................................

13. Wat vind u van de nauwkeurigheid van de software om u te herkennen?
    - Goed
    - Matig
    - Slecht
    - Omdat..................................................

14. Heeft u nog verdere opmerkingen over gebruik van biometrie in het algemeen of de opstelling in het bijzonder?
Appendix 2 Detailed questionnaire results

This appendix gives the distribution of the answers as provided by the 25 individuals that took part in testing the software and filled out the questionnaire. The questionnaires were made available in both Dutch and English. In total 23 Dutch and 2 English questionnaires were used. Most of the questions allowed for additional remarks. Where required, these optional remarks were translated into English and are added to the corresponding answer distributions below.

Question 1

![Figure 47: Answer distribution for Question 1]

Question 2

![Figure 48: Answer distribution for Question 2]
Appendix 2 Detailed questionnaire results

Question 3

Q3: Please indicate your experience with software in general

Figure 49: Answer distribution for Question 3

Question 4

Q4: Please indicate which one or more of the following biometric technologies you have used before.

Figure 50: Answer distribution for Question 4

Question 5

Q5: How often do you use biometric technology

Figure 51: Answer distribution for Question 5
Appendix 2 Detailed questionnaire results

Question 6

Q6: Are you concerned about abuse of biometric information stored in biometric systems?

Not at all | A bit | Very much | Not when it is properly protected, and the usage cannot be abuse
---|---|---|---
6 | 11 | 3 | 6

Figure 52: Answer distribution for Question 6

Question 7

Q7: The explanation that was given on how to use the equipment was clear

Agree | Neutral | Disagree
---|---|---
25 | 0 | 0

Figure 53: Answer distribution for Question 7

Additional remarks: -

Question 8

Q8: Which sensor has your preference?

Sensor A | Sensor B
---|---
21 | 9

Figure 54: Answer distribution for Question 8
Appendix 2 Detailed questionnaire results

Additional remarks:
- Answer given “Sensor A” and “Sensor B”: “No preference.” (three times)
- Answer given “Sensor B”: “Sensor does not have to be cleaned.”
- Answer given “Sensor B”: “Sensor is easier to use”
- Answer given “Sensor A”: “Don't have to push so hard, quicker to capture, clearer.”
- Answer given “Sensor B”: “Because sensitivity for dirt.”
- Answer given “Sensor A”: “Faster.”
- Answer given “Sensor A”: “Because it reacts faster.”
- Answer given “Sensor A”: “Less pressure required, reacts faster.”

Question 9

![Figure 55: Answer distribution for Question 9](image)

Additional remarks:
- Answer given “Agree”: “Does what it is supposed to do.”

Question 10

![Figure 56: Answer distribution for Question 10](image)

Additional remarks:
- Answer given “Agree”: “Does what it is supposed to do.”
Appendix 2 Detailed questionnaire results

Question 11

![Figure 57: Answer distribution for Question 11](image)

Additional remarks:
- Answer given “Neutral”: “Because oral explanation of the software was required.”
- Answer given “Disagree”: “I made a wrong move, then we had to restart the software to reset it.”

Question 12

![Figure 58: Answer distribution for Question 12](image)

Additional remarks:
- Answer given “About right”: “Reacts almost immediately.”

Question 13

![Figure 59: Answer distribution for Question 13](image)

Additional remarks: -
**Question 14** Do you have any further comments on the use of biometrics or the software you just used?

Additional remarks:

- “Sensor A sometimes detected a finger while it was not there.”
- “An automatic cleaning of the device would be nice.”
Bibliography


BSI 2010  Projekt BioKeyS-PilotDB-Teil1, Testbericht, Version 1.1 vom 31.03.2010.


Bibliography


Sfin http://biolab.csr.unibo.it/research.asp?organize=Activities&select=&selObj=12&pathSubj=111%7C%7C12&.


Bibliography


