Enhancing the performance and security of Helper Data Systems

Enhancing the performance and security of Helper Data Systems

Taras Stanko

Printed by Gildeprint Drukkerijen, Enschede, The Netherlands
ISBN: 978-90-386-4944-3
A catalogue record is available from the Eindhoven University of Technology Library.

Enhancing the performance and security of Helper Data Systems

This research was supported by the Netherlands Organization for Scientific Research NWO through Cyber Security project 628.001.019 (ESPRESSO).

Cover: The cover shows a fingerprint on an analog electronic physical unclonable function.

Enhancing the performance and security of Helper Data Systems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op donderdag 23 januari 2020 om 11:00 uur

door

Taras Stanko

geboren te Boryslav, Oekraïne

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof. dr. M.T. de Berg
1e promotor:	dr. B. Škorić
2e promotor:	prof. dr. S. Etalle
leden:	prof. dr. S. Katzenbeisser (Universität Passau)
	prof. dr. ir. J.P.M.G. Linnartz
	prof. dr. ir. R.N.J. Veldhuis (Universiteit Twente)
	prof. dr. ir. F.M.J. Willems

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Acknowledgments

Ohh, what a journey it was! These last four years were the most challenging and yet the most fulfilling years of my life till now. I challenged myself in many ways, I grew up professionally and mentally. This theses is the result of hard and persistent work. Fortunately, I was lucky to receive tremendous help and support during this journey, and I am more than sure without this help I would have never made it. I am very grateful to all the people who shared this journey with me.

Thank you to my promoter Boris Škorić, for all your guidance, patience and all the knowledge you shared with me. You manage to explain the most difficult things in an easy and understandable way. You are a true talent and professional. I am really lucky that I had a chance to work under your supervision.

Thank you to the head of our research group, Sandro Etalle, to make this project happen and for teaching me the brilliance that 'Slow is fast'. This is one of the best pieces of advice I received during my PhD. I would like to express my gratitude to the Doctorate Committee Stefan Katzenbeisser, Jean-Paul Linnartz, Raymond Veldhuis and Frans Willems who spent their valuable time and expertise to improve the quality of my thesis by providing helpful comments and critical feedback.

A special thanks goes to Davide, Sowmya, Guillaume, with whom I shared the office throughout the last few years. Thank you for working from the library all the time, thus helping me concentrate better on my work. Also, I want to thank Jolande, Mirjam and Anjolein for creating a warm and friendly atmosphere in the group. Organization of the process is a key, which was done at the highest level. A big thank you for many debates, lunch breaks, coffees, cakes, sweets and other social activities to (in no particular order): Gustavo, Mahdi, Laura, Stefan, Christine, Alessandro, Lorenz, Milan, Alex, Ömer, Mahdi, Laura, Dion, Davide, Daan, Pavlo, Andy, Daniel, Dominik, Harm, Leon, Frank, Manon, Manos, Niels, Jeroen, Chloe, Meilof, Matthew, Guillaume, Huyn, Luca,

Nicola, Wil, Alessio, Sowmya, Sokratis, Serena, Umbreen and Estuardo.

Lastly, I want to express big gratitude to my family. I want to thank my daughter Lisa and my wife Olya. Lisa, you came in my life during the last year of my PhD. You quickly taught me better time management skills as well as the art of sleeping faster. Lisa, you gave me a big motivation and strength for my work. Olya, I have found in you everything that I was missing myself, we are a great team and I am grateful for being able to share my life you. Thanks for all the patience, support, love and friendship I have received from you. I'm grateful to my parents, who allowed me to pursue my goals. Thank you mom and dad for coming to us and all your help with taking care of Lisa.

Off I go to new adventures. Good luck to all of you with yours.

Eindhoven, January 2020

Taras Stanko

Abstract

In the past decade biometrics and Physical Unclonable Functions (PUF) have become a popular alternative to standard authentication methods, such as passwords. Protection of secrets generated by biometrics and PUFs is done similar to password protection: a hash of the secret is stored. The hash is resilient to different types of attacks (e.g. hacks and insider attack). Secret keys generated by biometrics and PUFs are prone to noise, thus an error correction step is required. However, error correction data has to be stored somewhere and may leak sensitive information. A standard approach to reduce leakage is to use a Helper Data System (HDS), and a two-stage HDS in particular. At the first stage quantization of the measurement data is performed and a binary string is obtained. The helper data reduces the discretization errors. However, the binary string may still contain errors. An error-correction code (ECC) is applied at the second stage. The HDS approach requires biometric data to have a fixed-length representation after the first stage. This is not an easy task since a biometric measurement does not always reliably produce the same number of features (e.g. fingerprint minutiae).

The device that performs the biometric verification or the PUF key reconstruction is often resource-constrained, e.g. a smartcard. The bottleneck of the HDS is the computationally expensive ECC decoding step. The error correction can be outsourced to a more powerful second party. An eavesdropper then learns the error pattern. The error pattern can be data dependent, which leads to potential security issues. Additionally experiments have shown that some PUFs are prone to drift. Thus, the PUFs become recognizable when the outsourcing is used. This has a potential impact on privacy.

To improve state-of-the-art on HDSs and biometric template protection we have introduced the following four improvements:

- We have optimized the quantization procedure in the first stage. At an intermediate level of noise the improvement in terms of mutual information

is a few percent (up to 7%) and the bit error rate can be reduced by as much as 50%. The result is generic and not limited to biometric data only. (Chapter 2)

- We have introduced a new fixed-length representation for fingerprint features. The representation is based on differences between *pairs* of features. We achieved similar recognition performance as state-of-the-art template protection schemes but with smaller template size, which makes our approach faster and more practical. (Chapter 3)
- We have built a two-stage helper data system for fingerprints from the above mentioned fixed-length representation combined with the optimized quantization scheme. The scheme was implemented with Polar Codes as the error correction code. The best results were obtained by combining three enrollment images. The performance degradation due to added privacy is minimal in the case of high-quality fingerprints. (Chapter 4)
- We have introduced two modifications to the ECC outsourcing scheme which together eliminate both leakage problems: leakage about PUF keys due to data-dependent noise and identifiability of PUFs due to drift. (Chapter 5)

Contents

Chapter 1

Introduction

1.1 — Security with noisy data

1.1.1 – Digital security is not only digital. Two of the main concepts in security are *confidentiality* and *authenticity*. Confidentiality means that data, be it stored or transmitted, should be accessible only by authorized parties. Authenticity of data means that someone who is viewing the data can verify that the origin is as stated. Authenticity of a party means that others can verify that this party is who it claims to be. Authenticity of a physical object means that someone who inspects the object can ascertain that it possesses certain special properties that are difficult to create or, even better, that it is exactly the same object as one that has been enrolled at some previous time.

In the digital domain, confidentiality and authentication are typically taken care of by cryptographic means. Data encryption is used for confidentiality; message authentication codes and digital signatures for authentication. The secure deployment of these cryptographic primitives crucially depends on the secrecy of keys. In the case of symmetric cryptography, the sender (Alice) and receiver (Bob) of a message have the same key. In asymmetric cryptography, each party has its own private key and public key. A private key is used for decryption of ciphertexts and for signing data. A public key is used for encryption and for verification of signatures. Symmetric keys and private keys need to be kept secret. This sounds simple, but actually it is a major concern. Keeping digital keys out of the hands of adversaries is notoriously difficult. For instance, commercially available devices (smartphones, computers etc) are vulnerable to malware. This vulnerability is caused by a combination of factors such as system complexity, legacy issues and user-hostile business models. Governments are reluctant to force a solution. On the contrary, many governments demand backdoors. It is widely known that certain governments are stockpiling vulnerabilities to develop offensive cyber-weapons [69] and then

sometimes lose control of them. To make the problem of key protection worse, *the theft of digital keys can easily go unnoticed*, which prolongs the period during which an adversary is able to do damage.

Digital devices are embedded in the physical world. To hardware and software engineers the physical world is often a nuisance: it causes noise and glitches, and it imposes constraints. However, the interaction between the digital and the physical realm is more than a hindrance. In fact, it can be harnessed in several ways to ameliorate some of the security problems mentioned above. For instance, authentication does not have to depend on digital keys. As the saying goes, authentication is based on what you know, what you have, or what you are. The 'what you have' may be a physical object that is difficult to clone. Research into Physical Unclonable Functions (PUFs) has yielded materials that are cheap to manufacture but infeasible to clone even for the manufacturer. Authentication of non-secret physical properties has the advantage that it does not require any digital data to be kept secret. Furthermore, theft of a physical token is far more conspicuous than theft of a digital key. The 'what you are' refers to biometrics. Every person's body has distinguishing features such as fingerprints, ear prints, irises, vein patterns, DNA, and face structure. Some of these features are completely unique, differing even between identical twins. Biometrics have been used for criminal forensics since the 19th century. Automated fingerprint recognition is gaining popularity, with millions of smartphones recognizing the fingerprint or face of their owner.

Key storage is another application where interaction with the physical world can be exploited to improve security. Several types of PUF are suitable for use as a so-called Physically Obfuscated Key (POK). Here a physical structure is subjected to a measurement, and a cryptographic key is derived from the measurement outcome. Effectively, the cryptographic key is present in analog form in the nano-scale physical structure of the object. POKs have a number of advantages over digital storage. When a device is in its switched-off state, a POK is less vulnerable to key extraction attacks than digital forms of storage such as flash memory. Furthermore, a POK key can be wiped from digital memory immediately after use, and re-generated on the fly when needed, so that it exists in (vulnerable) digital form only for brief periods of time. A third advantage is Public Key Infrastructure logistics. In e.g. military scenarios, the authority that handles key management must program keys into many devices. This involves people physically transporting private keys to many destinations, which introduces a vulnerability. When the private keys are implemented as POKs, on the other hand, the vulnerability can be avoided. The (random) private key is already inside the device; the corresponding public key is computed by the device and communicated to the authority. The private key never leaves the device.

This thesis is about privacy-preserving biometric authentication and about Physically Obfuscated Keys. As will be explained below, these topics have a lot

in common regarding security requirements, technical difficulties, and solution methods.

1.1.2 – Privacy-preserving biometric authentication. In the past decade biometrics has become a popular alternative to standard authentication methods, such as passwords and tokens. Passwords are possible to forget, tokens can be left behind. Biometrics is now widely used. Fingerprint-based authentication systems are deployed on numerous smartphones and smartcards. Many countries apply biometrics for voter registration. More then 150 countries issue a passport containing biometric information; more then one billion biometric passports have been issued worldwide.

It is infeasible to keep biometric information completely secret. We leave latent fingerprints behind on smooth surfaces. Our faces and irises can be photographed. Fortunately, biometric authentication does not require secrecy. The only requirement is that the verifier does a measurement on an actual living human. This is called 'liveness detection'.

Even though biometric information cannot be kept secret, there are good reasons to minimize access to biometrics. Unrestricted access to biometric data would lead to serious privacy and security problems. (i) Biometric data can reveal medical conditions. (ii) Storing biometric data in unprotected form in multiple databases makes it possible to cross-match an individual across databases. (iii) Easy access to biometric information makes it possible for attackers to target an individual, and to create high-quality fake biometrics matching that person. Even though liveness detection should catch such spoofing, it is best not to make the attackers' life easy. (iv) It becomes possible to leave fake forensic evidence at a crime scene, e.g. in the form of latent fingerprints or synthesized DNA.

Authentication consists of two steps: enrollment, and verification. During the enrollment the characteristic is provided and stored[1]. The characteristic is checked with the stored one during the verification, which leads to yes/no decision. It depends on the circumstances how well the enrollment data can be protected. We distinguish between two attacker models. In the first case, special hardware is available for storing secrets. This trivially solves the problems. In the second attacker model there is no such hardware. The system is vulnerable to 'insider attacks', where the adversary knows shared secrets, such as encryption keys, has access to all stored data but does not know the enrollment and verification value. This means that simply encrypting biometric data is not a solution, since the attacker is able to decrypt it. Additionally, if a third party is involved, this party should learn nothing about the enrollment and verification values.

[1] Here we are concerned about confidentiality. It is straightforward to ensure integrity and authenticity of the enrollment data by signing it. If the integrity in not protected, an adversary may do a denial of service attack by modifying the stored data.

We are interested in the second attacker model, which is more or less 'standard' in the biometric privacy community. In the scientific literature there are the following methods for protecting biometric enrollment data:

- Helper Data System with one-way hash function.
  A cryptographic hash function maps data of arbitrary size to a bit string of a fixed size (a hash) and is designed to be a one-way function, that is, a function that is infeasible to invert. Comparison of hashed data is a well-known technique for password protection against insider attacks. In this setting it is very hard for an adversary to find a pre-image of secured template. One would like to do the same with biometrics. A one-way hash function assures that similar inputs result in completely different outputs. Thus error correction is required when we want to hash biometric data. We need special error correction with little leakage, since in the standard attacker model the adversary sees the redundancy data. A so-called Helper Data System (HDS) performs error correction, while making sure that the redundancy data does not leak anything important. Roughly speaking this is achieved by revealing only the noisy part of the enrollment measurement.
  The following methods are alternatives to the HDS approach and are designed to tolerate noise without correcting it.
- Homomorphic encryption.
  Homomorphic encryption [73], [107] is a form of cryptography that allows computation on encrypted data (ciphertexts). The enrollment data is encrypted with a public key. During the verification a computation is performed on the encrypted enrollment data and the fresh measurement, yielding the encryption of the similarity between the two measurements. The holder of the private key performs the decryption. The scheme provides confidentiality even when the biometrics is compromised or has low entropy. However, the computation on encrypted data is time consuming, and the involvement of the third party causes communication overhead.
- Locality sensitive hashing.
  Locality sensitive hashing [42] (LSH) reduces the dimensionality of the data, while making sure that similar inputs result in similar outputs. A subcategory of LSH is the random projection method. The core idea of the method is given by Johnson-Lindenstrauss lemma [46]. A set of points in a high-dimensional space can be embedded into a space of much lower dimension in such a way that (Euclidean) distances between the points are nearly preserved. A disadvantage of the random projection method is that the output leaks information about *specific components* of the input, which can lead to privacy problems.
- Sparse coding with ambiguization.
  This approach also applies random projections, but without the dimen-

sional reduction. The result of the projections is quantized in such a way that a vector is obtained that contains mostly zeroes. Finally artificial noise is added [72] to obfuscate the input values. This approach has been proposed very recently. It is not yet known how well specific components of the input are protected.

- Cancellable biometrics.
  Cancellable biometrics [70] refers to an irreversible transform that alters the input. If a template is compromised, a new template can be re-generated from the same biometrics. A disadvantage is that during the transform a significant amount of information is lost. Some types [67], [68] of cancellable biometrics use the random projections method with a secret projection matrix; this requires a shared secret, which falls outside the standard attacker model.

The above mentioned privacy-preserving techniques are summarized in Table 1.1.

Table 1.1: *Comparison of privacy-preserving techniques*

Approach	Advantages	Disadvantages
Cancellable biometrics	New template can be re-generated	Information is discarded; Requires token/shared secret
Homomorphic encryption	Provides privacy even for compromised biometrics; Secure even for low-entropy key	High computational complexity; Communication overhead
Locality-sensitive hashing	Low computational complexity	Unknown security of the system
Sparse coding with ambiguation	Low computational complexity	Unknown privacy
Helper Data System +OWF	Only noise pattern is leaked; Precisely known security level	Requires error-correction codes

A biometric authentication system should tolerate noise, have a low computational complexity, and store the enrollment data in a secure way. If emphasis is put on discarding as little information as possible (because fingerprints does not have much entropy), low computational complexity, and well understood security level, then the Helper Data System approach scores highly. This thesis focuses on the HDS approach.

Helper Data Systems.

Functionality. The Helper Data approach was introduced in [22, 24, 47, 54]. A HDS in its most general form is shown in Fig. 1.1. The `Gen` procedure takes as input a measurement X. `Gen` outputs a secret S and (public) Helper Data W. The helper data is stored in insecure memory. In the reproduction phase, a fresh measurement Y is obtained. Typically Y is a noisy version of X, close to X (in terms of e.g. Euclidean distance or Hamming distance) but not necessarily identical. The `Rep` procedure takes Y and W as input. It outputs $\hat{\mathsf{S}}$, an estimate of S. The scheme has two requirements: it is infeasible for an attacker to learn

S (concealing), and $\hat{S}$ should equal S with high probability (binding). If W leaks nothing about S then the HDS is called a *Zero Leakage* HDS (ZLHDS).

Two special cases of the general HDS are the Fuzzy Extractor (FE) and the Secure Sketch (SS) introduced in [24]. A FE extracts nearly uniform randomness S from its input X. Thus, S can be used as a key in any cryptographic application. The Secure Sketch has $S = X$. If X is not uniformly distributed, then S is not uniform. The SS is suitable for privacy-preserving biometrics; high entropy of S (given W) is required, but not uniformity.

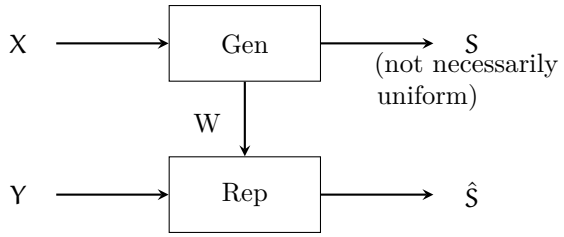


Figure 1.1: *Data flow in a generic Helper Data System.*

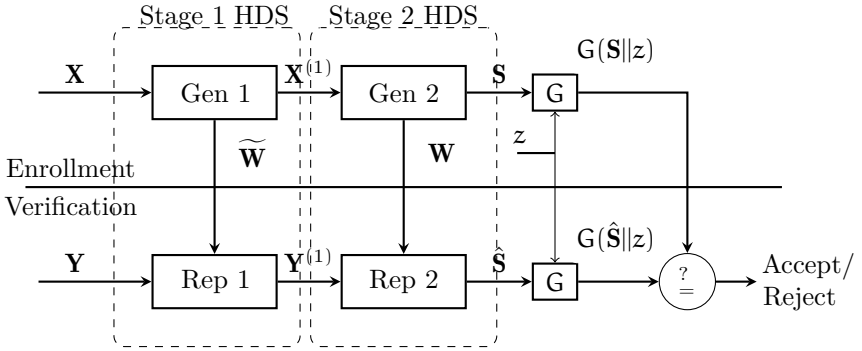


Figure 1.2: *Two-stage Helper Data System with one-way function* G.

Two-stage architecture. A HDS for a continuous source consists of two stages (see Fig. 1.2).

At the first stage quantization is performed. For this step discrete helper data [96] and continuous helper data [19] were proposed. An example of quantization in [19] is depicted in Fig. 1.3.

The range of X is divided in N intervals corresponding to S (the division does not have to equiprobable). Each interval is divided in M equiprobable regions which determine the helper data (In the figure $N = 4$, $M = 3$). The value of S and W follow from X by determining in which interval X lies. Given

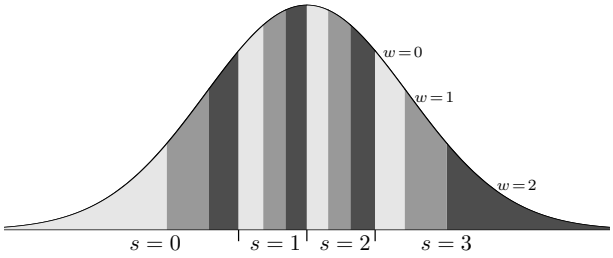


Figure 1.3: *Quantization intervals and helper data regions.*

X, we obtain a secret and a helper data. The helper data W, leaks nothing about the secret. During the reconstruction one looks for the interval closest to Y that has correct W. It is intuitively clear why such a HDS has good security. The secret S can be thought of as the most significant digits of X and the helper data W as the least significant, noisy digits. On one hand, the equiprobable intervals ensure the W leaks nothing about S. On the other hand, by revealing W one reveals information that mostly consists of measurement noise and thus does not pose a privacy risk.

The helper data reduces the discretization errors. However, some noise still may be present. An error-correction code (ECC) is applied at the second stage. The commonest approach is a Code-Offset Method (COM) [47] which utilizes a linear binary error-correction code. The COM in its simplest form functions as a Secure Sketch. The syndrome-only COM works as follows. Let $\mathbf{X}^{(1)}$ be a binary string. Let Syn and SynDec be the syndrome encoding and decoding functions of the code. The helper data is simply the syndrome of the noisy string.

$$\mathbf{W} = \mathsf{Syn}(\mathbf{X}^{(1)}); \quad \hat{\mathbf{S}} = \mathbf{Y}^{(1)} \oplus \mathsf{SynDec}(\mathbf{W} \oplus \mathsf{Syn}(\mathbf{Y}^{(1)})). \tag{1.1}$$

Intuitively the syndrome of $\mathbf{X}^{(1)}$ can be thought of as the least significant digits of $\mathbf{X}^{(1)}$. The syndrome of $\mathbf{X}^{(1)}$ inevitably leaks some information about $\mathbf{X}^{(1)}$. If the COM needs to be deployed as a Fuzzy Extractor, a secure key can be derived from $\mathbf{X}^{(1)}$ by using privacy amplification [8].

Fixed-length representations. The usage of an error correcting code requires a fixed-length representation of biometrics. This is not an easy task since fingerprint minutiae (see Chapter 3), a widely used local feature for verification and identification, tend to appear/disappear. Several attempts have been made to come up with a fixed-length representation, for example [27], [45], [102], [64]. Most of the representations produce binary data, which is sub-optimal for the HDS approach, since it does not allow us to utilize a first stage HDS. A very

useful fixed-length representation called spectral minutiae was developed by Xu et al. [102]: a Fourier-like spectral function is built up on a fixed grid, in such a way that each detected fingerprint minutia contributes to the function. Their approach obtains translation invariance by taking the absolute value of the Fourier transform, discarding significant amount of information. Additionally the approach requires multiple rotations of the verification image to optimize the matching procedure. This increases the complexity of the template protection scheme.

1.1.3 – Physically Obfuscated Keys. As mentioned before this thesis is focused on key storage application by using PUFs. Similarly to humans some (electronic) devices have their own 'fingerprints', which are the unique noisy properties from steps in the manufacturing process that are uncontrollable. These unique properties are called Physical Unclonable Functions (PUF). The notion of PUF was introduced in [66]. A PUF is a physical structure that consists of many random, uncontrollable components. One is looking for physical structures with the following useful properties: 1) It is easy to measure the response of PUF given the stimulus (usually called challenge) 2) Given the challenge, the response is unpredictable. An attacker, not having the PUF at hand, should have a significant amount of uncertainty about the response of a PUF to a given challenge. 3) Physical unclonability. The PUF is very hard to reproduce even by the manufacturer. 4) Read-proof. Measurements (both destructive and non-destructive) must not reveal accurate information about the composition of the physical structure. 5) Tamper evidence. When the PUF is attacked in an invasive way, the PUF will be damaged to such an extent that its challenge-response behavior drastically changes. 6) Integration. Preferably the PUF is inseparably bound to the device it is meant to protect.

Numerous PUF-like constructions have been developed [12, 30, 37, 66, 92], [20], [41], [98], [55], [38], [36], [53], [34], [52], [85], [57]. Only optical PUFs have most of the desirable properties. However, the name PUF is used for any construction that has a subset of these properties. There are also practicality requirements for a PUF in order to be really useful. The PUF should be cheap and easy to integrate into the production process of a device. Finally, the PUF needs to be very robust to various environmental changes such as temperature and humidity.

Various applications have been proposed for PUFs: anti-counterfeiting, device authentication, tamper protection, and key storage. We will concentrate on the key storage, i.e. Physically Obfuscated Keys. Of the above list of desirable PUF properties, POKs require only ease of readout, unpredictability, read-proofness and integration. Several PUF types are suitable for building POKs that are integrated with electronics.

- Coating PUF [92] is an IC covered with an opaque layer that contains a random mixture of particles with different dielectric properties. The PUF

readout is a capacitance measurement of the bottom side of the coating. The coating is chemically inert offering strong protection against physical attacks. Damage to the coating results in a noticeable change of the PUF properties.

- Delay-based PUFs [30] such as Arbiter PUFs and Ring Oscillator PUFs [4]. The unique property is the delay time which occurs when a signal is sent through a series of components. In the Arbiter PUF a signal is sent through two different configurable paths. There is one output bit, which signals which path is fastest. In the RO PUF the signal is looped back from the output to the input, and the oscillation time constitutes the PUF output.
- Memory-based PUFs such as SRAM (Static RAM) PUFs [35] and DRAM (Dynamic) PUFs [50]. The unique property in SRAM PUFs is the startup value of memory cells; in DRAM PUFs it is the speed at which memory cells lose their electric charge.

There are strong similarities between the data processing of biometrics and POKs. As with biometrics, there is an enrollment phase and a verification (reconstruction) phase. Biometrics and POKs both need a HDS. In the case of POKs this is not for privacy but for security reasons: the cryptographic key has to be secure even when the adversary has access to the helper data. Some POKs (e.g. Coating, Ring Oscillator, Optical, DRAM) have an analog response, thus a two-stage helper data system is required. Other POKs (e.g. SRAM) have a binary response, and there is only need for a 1st-stage HDS.

There are also notable differences between POKs and biometrics. Perhaps the most important difference, from a security point of view, is that it is much easier to keep POK keys secret than biometric data. A POK can be embedded in an electronic device that controls access to the POK, but a biometric feature of the body is always vulnerable to surreptitious measurements by attackers. It makes perfect sense to derive secret keys from PUF responses, whereas doing such a thing with biometric data is perilous. Furthermore, if biometric data is stolen, one cannot re-issue a new body part; re-issuing a new POK, on the other hand, is feasible. Another important difference is that, obviously, a person always has direct access to his biometric, whereas a POK can be left behind. For these reasons, POKs can be used in many more (cryptographic) applications than biometrics.

1.1.4 – Outsourcing of the error correction. The device that performs the biometric verification or the POK key reconstruction is often considered to be resource-constrained, e.g. a smartcard. The bottleneck of the HDS is the computationally expensive ECC decoding step (Syndrome decoding in Eq. 1.1). An elegant solution was proposed in [39], where it was shown how the decoding can be securely outsourced to a more powerful second party. The string $\mathbf{W} \oplus \mathtt{Syn}(\mathbf{Y}^{(1)})$ can be communicated to the other party, which then

performs the decoding and sends back the error pattern. An eavesdropper learns only the error pattern. This trick is known as 'Reverse Fuzzy Extractor'. The most difficult HDS task for the constrained device is now merely to compute a syndrome, which can be done very efficiently.

1.2 — Challenges

My PhD research was done in the ESPRESSO project (Efficient and Strong template PRotection by enabling Secure Sketch On-card). The aim of this project was to develop good HDS algorithms for fingerprint matching. Because of the similarity between biometrics and POKs this meant in practice that I was working on HDS algorithms in general. In fingerprint template protection and POKs there are numerous technical challenges that have to be addressed.

Low entropy of fingerprints.
In the HDS approach, a one-way hash function is computed of the error-corrected biometric. The input of the hash function should have high entropy, otherwise the brute force attack is possible to guess the 'secret'. In principle biometrics contain a lot of information. However, this information can be reproducibly acquired only if the measurements take place in a precisely controlled environment. When the capture of biometric data is automated or unsupervised, the reliably extractable information (mutual information) is much less, e.g. less than 40 bits for fingerprints [91] or even less than 20 bits [95]. Since this is not secure we need to (1) combine multiple fingers (2) extract as much information as possible per finger. This problem has to be addressed already at the preprocessing step, i.e. before applying the first-stage HDS. Furthermore the HDS itself must also not waste entropy.

High bit error rate.
Fingerprint images are inherently noisy. This is caused by poorly controlled finger alignment and pressure, differences in sensors, dirt etc. The noise generates challenges at every stage of template protection scheme. In particular the 1st stage HDS outputs a string that is short and has a high Bit Error Rate (BER). Thus the 2nd stage HDS needs a good ECC which is capable of handling the high BER while still having a good code rate, even for short codewords.

Appearance/disappearance of minutiae.
A minutia is a special point in a fingerprint, such as an ending or bifurcation of a ridge, or a small isolated ridge. The measurement noise can cause the number of detected minutiae to be different on every image capture. At the same time, an error-correction code requires input to have a fixed length. It is not straightforward to transform a list of minutiae into a fixed-length representation in such a way that the appearance/disappearance of minutiae has only a minor effect on the fixed-length output.

Data dependent noise.
The idea of the Reverse Fuzzy Extractor (outsourcing the error correction)

works perfectly if the noise leaks nothing about the original data. However in the real life problems the noise can be data dependent. This leads to potential security issues. For example for RO PUFs after observing one error pattern an attacker learns 2% of all information about the source and 21% after observing ten error patterns (Chapter 5). There is no way to upper bound the leakage if there is no a priori upper bound on the number of observations that the adversary can do.

PUF drift.

Experiments have shown that some PUFs are prone to drift. Thus, the PUFs become recognizable when the Reverse Fuzzy Extractor is used. This has an impact on privacy. Additionally the POK reconstruction will fail after certain amount of drift.

1.3 — Research questions

My research was focused on the following questions.

Research Question 1. Can we maximize the entropy extracted by a Zero Leakage Helper Data System quantizer for a given source distribution and noise level?

Research Question 2. Can we construct a high-performance HDS based on fingerprint minutiae? Here high performance means high accuracy of the matching decision as well as fast processing.

Research Question 3. Is it feasible to use the Reverse Fuzzy Extractor trick when the noise is data dependent and the POK has drift?

These questions were motivated by the objectives of the project in combination with the technical challenges discussed in Section 1.2.

1.4 — Contributions

To address **Research question 1** we have optimized the zero leakage schemes developed in [97], [19]. In [97] zero leakage property was introduced and discrete helper data was introduced having the zero leakage property. This scheme has equiprobable secret S and equiprobable helper data W (see Fig. 1.3). In [19] the number of quantization intervals of W is sent to infinity, providing continuous W. Here the subdivision of S-regions needs to remain equiprobable, but the scheme also works for non-uniform S. In [19] S-regions have not been optimized. We have performed the optimization of the quantization, i.e. the choice of S-intervals. We characterized the performance of a HDS as the number of secret bits that can be consistently reconstructed from the source (mutual information).

We worked under the following conditions: i) W is continuous and satisfies zero leakage property. ii) Not necessarily uniform S. Our results show that optimization of the quantization intervals brings marginal improvement when the signal to noise ratio is either very low or very high. At an intermediate level of noise the improvement in terms of mutual information is a few percent (up to 7%) but the bit error rate can be reduced by as much as 50%. This has practical impact on performance of real-life error-correcting codes. The result is generic and not limited to biometrics only. This work led to the publication [81] (Chapter 2).

To address **Research question 2** we have (a) introduced new fixed-length representations for fingerprint minutia sets; (b) built and tested a two-stage HDS with highly efficient error correction in the 2nd stage.

Inspired by Xu et al's spectral function approach [105], [106], we introduced a new fixed-length representation of fingerprint minutiae. The representation is based on coordinate differences between *pairs* of minutiae. By working with coordinate differences we immediately obtained a translation-invariant representation. In comparison to Xu et al., we achieved similar recognition performance before discretization, but with smaller template size, which makes our approach faster and more practical. Also we noticed that, in contrast to Xu et al., trying multiple rotations of the verification image does not have much impact on the recognition performance. Thus the rotation step can be skipped, which leads to a further speed-up. This work led to the publication [82] (Chapter 3).

We have built a two-stage helper data system from the above mentioned spectral function representations, combined with a zero leakage quantization scheme and the Code Offset Method. We have introduced an enrollment method that we call 'superfinger', in which we average the spectral functions from multiple enrollment images. We have used the results of [81] to choose the best number of quantization intervals. Due to the low signal-to-noise ratio, two quantization regions (binarization) is the optimum. We have tested the recognition performance for various choices of system parameters at every step of the data processing. We evaluated the recognition performance of the system by making ROC curves and extracting from them Equal Error Rates.[2] For our purposes the best choice of an ECC turned out to be Polar Codes, because of short codeword length and high bit error rate that can be corrected. Polar codes are low-complexity capacity-achieving codes with flexible rate. By combining three enrollment images and constructing a polar code specifically tuned to the individual bit error rate of each bit position, we achieve an EER

[2] A Receiver Operator Characteristic (ROC) curve plots the False Accept Rate (FAR) versus the False Reject Rate (FRR). Each value of the decision threshold gives a point on the curve. The Equal Error Rate (EER) is the point where FAR=FRR.

of around 1% for a high-quality fingerprint database, and around 6% for a low-quality database. This is not as good as the EER of state-of-the-art matching *without* privacy protection, but similar to other work on protected templates. We see that at the optimal configuration the performance degradation with respect to unprotected minutiae sets is caused mainly by the step that maps the minutiae set to spectral function. The step from unprotected spectral function to HDS-protected spectral function is almost 'for free' in the case of high quality fingerprints. This work led to the publication [83] (Chapter 4).
Our work has shown that it is feasible to construct a fast two-stage HDS for fingerprint matching. The matching performance is not as good as for unprotected templates. Multiple fingers are needed for security as well as for obtaining EERs well below 1%.

To address **Research question 3** we introduced two modifications to the Reverse Fuzzy Extractor scheme which together eliminate both leakage problems, leakage about POK keys due to data-dependent noise and identifiability of POKs due to drift.

- Adding asymmetric artificial noise that turns the asymmetric noise channel into a symmetric channel. Due to the resulting symmetry the leakage is entirely eliminated: the error pattern no longer reveals anything about the POK value. Of course, the introduction of artificial noise leads to a loss of channel capacity. This loss is around 30% in the worst case that we encountered. It is entirely feasible to cope with such losses.
- A separate buffer that stores recent error patterns and the estimated drift in the prover device. The POK device keeps track of the computed error pattern over time. If the error pattern exhibits behavior constant in time (drift), the device modifies the stored helper data to compensate the drift. Thus the future error pattern will not reveal the drift. This solves the privacy problem.

This work led to the publication [77] (Chapter 5).

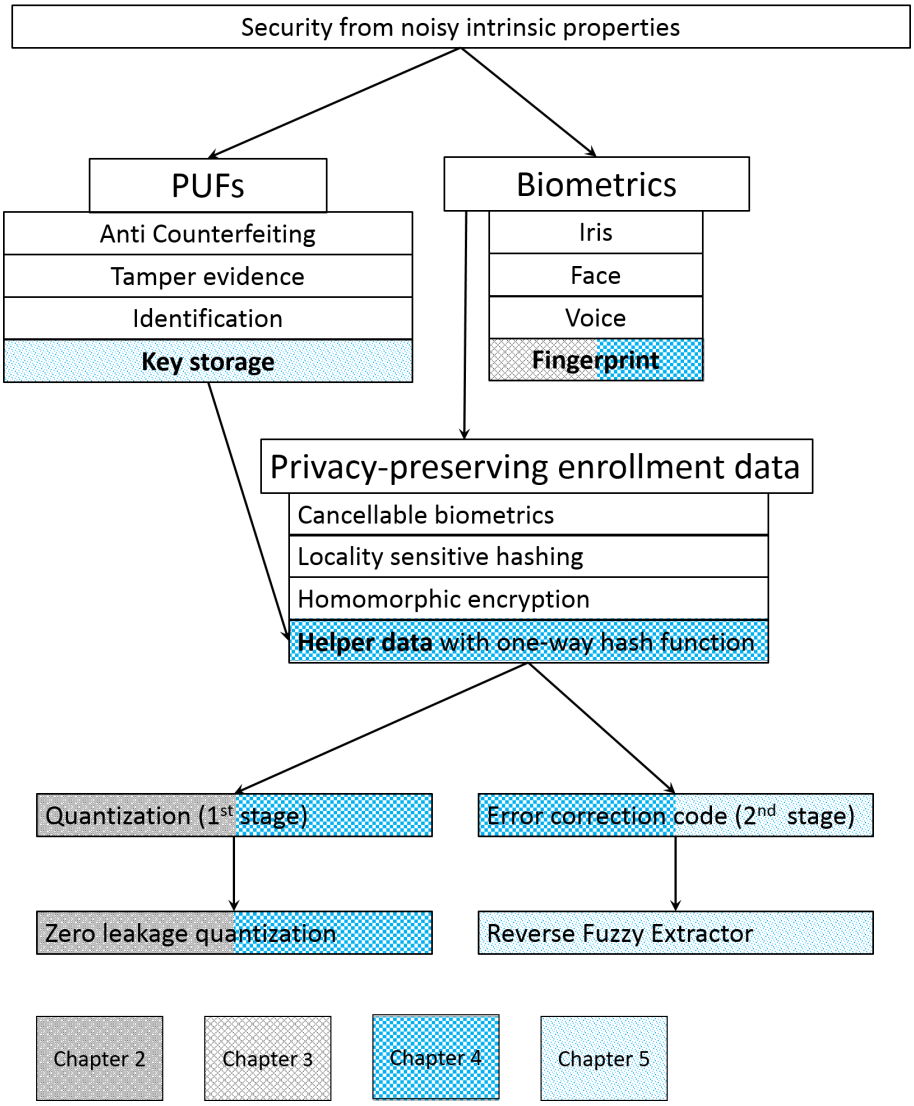


Figure 1.4: *Relationship between the topics discussed in this thesis. The shading indicates what is covered in each chapter.*

List of abbreviations and symbols

Notation	Meaning
A_α	quantization intervals at enrollment
f	probability density function
F	cumulative density function
G	hash function
$I(X;Y)$	mutual information between random variables X and Y
$L_x(q,\omega)$	spectral function
$L_{x\theta}(q,\omega)$	spectral function
$M_x(q,R)$	spectral function
$M_{x\theta}(q,R)$	spectral function
N	number of quantization intervals
p_α	$Pr[S=\alpha]$
R_{ab}	Euclidean distance between a-th and b-th minutia
S	secret
$\hat{S}$	reconstructed secret
V	zero-mean additive noise
W	one-dimensional discrete helper data
$\widetilde{W}$	quantile helper data, $\widetilde{W} \in [0,1)$
$\mathbf{W}$	helper data vector, $\mathbf{W} \in \{0,1\}^*$
$\mathbf{x_a} = (x_a, y_a)$	minutia location
$\mathbf{X}$	enrollment measurement
$\mathbf{Y}$	reconstruction measurement
Z	number of minutiae found in a fingerprint
β_a, β_b	invariant angles for a minutiae pair
δ	angle of image rotation

θ_a	minutia orientation
λ	attenuation parameter in the noise model
$\xi_{s,\widetilde{w}}$	mapping of $(s, \widetilde{w})$ to x-axis
$\rho(A, B)$	Pearson correlation between random variables A and B
σ_A	standard deviation of a random variable A
$\tau_{\alpha,\widetilde{w}}$	reconstruction boundary for α-th interval given helper data $\widetilde{w}$
ϕ_{ab}	angle between a-th and b-th minutiae
Ω_α	the left boundary of the quantization interval A_α
BAC	Binary asymmetric channel
BSC	Binary symmetric channel
BER	Bit error rate
COM	Code-offset method
ECC	Error correction code
EER	Equal error rate
FAR	False acceptance rate
FE	Fuzzy extractor
FRR	False rejection rate
HDS	Helper data system
POK	Physically Obfuscated Key
PUF	Physical unclonable function
ROC	Receiver operating characteristic
SS	Secure sketch
ZLFE	Zero leakage fuzzy extractor
ZLHDS	Zero leakage helper data system

Chapter 2

Optimized quantization in Zero Leakage Helper Data Systems

In this chapter we ask how to maximize the entropy extracted by a ZLHDS quantizer for a given source distribution and noise level (Research question 1). We optimize the scheme by deviating from equiprobable quantization boundaries. We derive analytical expressions for the mutual information between the original and the reconstructed secret given the helper data. The expressions depend on the number of quantization intervals, the signal-to-noise ratio, source and the noise distribution. It turns out to be a nontrivial optimization problem which does not have an analytical solution, due to the large amount of cross dependencies between variables that we are dealing with and intractable expressions. Every parameter choice needs a separate numerical optimization, which means large amount of numerics has to be done to perform a complete analysis of the approach. The results show that the optimized quantization is able to extract more 'useful' information than the equiprobable ZLHDS and lowers the bit error rate at reconstruction.

2.1 — Introduction

2.1.1 – Helper Data Systems. *Security with noisy data* is the art of reproducible extracting secret data from noisy measurements on a physical system. The two main applications are read-proof storage of cryptographic keys using Physical Unclonable Functions (PUFs) [9,33,56,75,89,90] and privacy-preserving storage of biometric data. Power-off storage of keys in *digital* memory can often be considered insecure. (For instance, fuses can be optically inspected with a microscope; flash memory may be removed and then read digitally.) PUFs provide an alternative way to store keys, namely in analog form, which allows the designer to exploit the inscrutability of analog physical behavior. Keys

stored in this way are referred to as Physically Obfuscated Keys (POKs) [29].

In both the biometrics and the PUF/POK case, one faces the problem that some form of error correction has to be performed, but under the constraint that the redundancy data (which is visible to attackers) does not endanger the secret. This problem has been addressed by the introduction of a special security primitive, the *Helper Data System* (HDS). A HDS in its most general form is shown in Fig. 1.1. The `Gen` procedure takes as input a measurement X. `Gen` outputs a secret S and (public) Helper Data W. The helper data is stored. In the reproduction phase, a fresh measurement Y is obtained. Typically Y is a noisy version of X, close to X (in terms of e.g. Euclidean distance or Hamming distance) but not necessarily identical. The `Rep` procedure takes Y and W as input. It outputs $\hat{\mathsf{S}}$, an estimate of S. If Y is not too noisy then $\hat{\mathsf{S}} = \mathsf{S}$.

Two special cases of the general HDS are the Secure Sketch (SS) and the Fuzzy Extractor (FE) [24]. The Secure Sketch has $\mathsf{S} = \mathsf{X}$ (and $\hat{\mathsf{S}} = \hat{\mathsf{X}}$, an estimator for X). If X is not uniformly distributed, then S is not uniform. The SS is suitable for privacy-preserving biometrics, where the stored biometric enrollment data is a cryptographic hash of X, just like hashed storage of passwords; high entropy of S (given W) is required, but not uniformity. The Fuzzy Extractor is required to have a (nearly) uniform S given W. The FE is typically used for extracting keys from PUFs and POKs. Note that there is a generic construction to obtain a FE from a SS: privacy amplification on X by applying a suitable information-theoretic hash function. This can be either a Universal Hash Function (UHF) [15, 48, 84] or, more sophisticatedly, a q-wise independent hash function. UHFs have the advantage of being simple to implement and providing information-theoretic security guarantees for all applications of the extracted key; however, they waste a lot of source entropy. Key derivation with q-wise independent hash functions can be done almost without any entropy loss [23] but gives information-theoretic guarantees only for 'unpredictability applications', which include signatures, Message Authentication Codes and keyed hashing.

In this paper we consider the general HDS case: $\mathsf{S} \neq \mathsf{X}$ and S is not necessarily uniform. The general HDS is of particular interest when X is a continuum variable: (i) The least significant digits of X are not interesting for key extraction and (ii) In view of the excellent performance of q-wise independent hashes [23] it is best to first extract from X a non-uniform high-entropy discrete secret and then compress it to make it more uniform.

2.1.2 – Zero Leakage quantization. In the biometrics case and in several PUF/POK scenarios the raw measurement data X is analog or nearly analog. A typical HDS then consists of two stages. The first stage is a HDS that maps the continuous X to a discrete space, i.e. it discretizes (quantizes) X. The second stage is a HDS acting on a discrete source, e.g. the Code Offset Method [7, 22, 24, 47, 99]. Both stages make use of helper data, and in both stages one has to worry about leakage.

In the fist stage it is possible to make a construction such that W leaks nothing about S. Intuitively speaking, W contains the 'least significant bits' of X, which are noisy, while S contains the 'most significant bits'. A HDS that achieves independence of S and W is called a Zero Leakage HDS (ZLHDS).
Verbitskiy et al. [96] introduced a Zero Leakage Fuzzy Extractor (ZLFE) for $X \in \mathbb{R}$.[1] They divided the space $\mathbb{R}$ into N intervals $A_0, \ldots, A_{N-1}$ that are equiprobable in the sense that $\Pr[X \in A_j] = 1/N$ for all j. At enrollment, if X lies in interval A_j then S is set to j. For the helper data they introduced a further division of each interval A_j into m equiprobable subintervals $(A_{jk})_{k=0}^{m-1}$. If the enrollment measurement X lies in interval A_{jk} then the index k is stored as helper data. The fact that all these subintervals are equiprobable leads to independence between the helper data and the secret.
De Groot et al. [19] took the limit $m \to \infty$ and showed that the resulting scheme is not just *a* ZLFE but *the* generic best performing ZLFE for $X \in \mathbb{R}$; other ZLFEs for $X \in \mathbb{R}$ can be derived from the generic scheme. Furthermore, de Groot et al. generalized the scheme of [96] from ZLFEs to general ZLHDSs by allowing intervals $A_0, \ldots, A_{N-1}$ that are not equiprobable. Several questions were left open regarding the `Rep` procedure in general ZLHDSs and the performance of ZLHDSs compared to ZLFEs.

2.1.3 – Contributions and outline. We investigate ZLHDSs for $X \in \mathbb{R}$.

- First we derive an optimal `Rep` procedure that minimizes the probability of reconstruction errors. We obtain analytic formulas for Gaussian noise and Lorentz-distributed noise.
- Using this `Rep` procedure we study the performance of ZLHDSs compared to ZLFEs. We define performance as the mutual information between S and $\hat{S}$ conditioned on the fact that the adversary knows W. This mutual information $I(S; \hat{S}|W)$ represents the maximum amount of secret key material that can be extracted from X using a ZLHDS. It turns out that the intricacies of the `Rep` procedure cause the mutual information to become a very complicated function of the choice of quantization intervals $A_0, \ldots, A_{N-1}$. We have to resort to numerics. Our numerical results for Gaussian source and Gaussian noise show that optimization of the quantization intervals yields an improvement with respect to the ZLFE in terms of mutual information as well as reconstruction error probability. In most cases the gain in $I(S; \hat{S}|W)$ is modest (a few percent), but the reduction of the error rate can be substantial. *We conclude that in practice it is better to use a ZLHDS than a ZLFE.*

In Section 4.3 we introduce the notation used in this paper and give a rather long summary of the results of de Groot et al. [19]. In Section 2.3 we derive the

[1] A high-dimensional measurement is usually split into one-dimensional components, e.g. using Principal Component Analysis or similar methods. A HDS is then applied to each component individually. The results are combined and then serve as input for the 2nd stage.

optimal Rep procedure and provide analytic expressions (as far as possible) for the mutual information and the error rate. Section 2.4 presents the numerical results for Gaussian X and Gaussian noise.

2.2 — Preliminaries

2.2.1 – Notation and terminology. We use capital letters to represent random variables, and lowercase letters for their realizations. The input and output variables of the HDS are as depicted in Fig. 1.1. Sets are denoted by calligraphic font. The set $\mathcal{S}$ is defined as $\mathcal{S} = \{0, \ldots, N-1\}$. For $\alpha \in \mathcal{S}$ we define $p_\alpha = \Pr[X \in A_\alpha]$. The expected value with respect to a random variable Z is denoted as $\mathbb{E}_z$. The mutual information (see e.g. [18]) between X and Y is denoted as $I(X;Y)$, and the mutual information conditioned on the third variable Z as $I(X;Y|Z)$. The probability density function (pdf) of the random variable $X \in \mathbb{R}$ in written as $f(x)$ and its cumulative distribution function (cdf) as $F(x)$.

2.2.2 – Zero Leakage definition. For technical reasons, de Groot et al. used the following definition of the Zero Leakage property.

Definition 2.1 (Zero Leakage)**.** *Let $W \in \mathcal{W}$. We call a HDS a Zero Leakage HDS if and only if*

$$\forall_{\mathcal{V} \subseteq \mathcal{W}} \ \Pr[S = s | W \in \mathcal{V}] = \Pr[S = s]. \tag{2.1}$$

The property (2.1) implies $I(W;S) = 0$.

2.2.3 – Noise model. We adopt the noise model from [19]. The X and Y are considered to be noisy versions of an underlying 'true' value. Without loss of generality X is taken to have zero mean. The standard deviations of $X, Y \in \mathbb{R}$ are denoted as σ_X and σ_Y respectively. The verification sample Y is related to the enrollment measurement as $Y = \lambda X + V$, where $\lambda \in [0,1]$ is the *attenuation parameter* and V is zero-mean additive noise, independent of X. We have $\sigma_Y^2 = \lambda^2 \sigma_X^2 + \sigma_V^2$. The correlation between X and Y is

$$\rho \stackrel{\text{def}}{=} \frac{\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]}{\sigma_X \sigma_Y} = \lambda \frac{\sigma_X}{\sigma_Y}, \tag{2.2}$$

with $\rho \in [-1,1]$. The relation between λ, ρ, σ_X, and σ_V is given by $\lambda^2 = \frac{\rho^2}{1-\rho^2} \frac{\sigma_V^2}{\sigma_X^2}$.
Two special cases are often considered:
Perfect enrollment. During enrollment there is no noise. The X equals the 'true' value. In this situation it holds that $\sigma_Y^2 = \sigma_X^2 + \sigma_V^2$ and $\lambda = 1$.
Identical conditions. The amount of noise is the same during enrollment and reconstruction. In this situation it holds that $\sigma_Y^2 = \sigma_X^2$ and $\lambda^2 = \rho^2 = 1 - \sigma_V^2/\sigma_X^2$.
The pdf of Y given $X = x$ is denoted as $\psi(y|x) = \nu(y - \lambda x)$. The noise is considered to be symmetric and fading, i.e. $\nu(-z) = \nu(z)$ and $\nu(z)$ is a decreasing function of $|z|$. The cdf corresponding to ν is denoted as V.

2.2.4 – The ZL scheme of [19]. The helper data is considered to be continuous, $W \in \mathcal{W} \subset \mathbb{R}$, and without loss of generality de Groot et al. set $\mathcal{W} = [0, 1)$. The left boundary of the quantization region A_α is denoted as Ω_α, $\alpha \in \mathcal{S}$. (See Fig. 2.1.) It holds that

$$\Omega_\alpha = F^{\text{inv}}\left(\sum_{i=0}^{\alpha-1} p_i\right), \tag{2.3}$$

where F^{inv} stands for the inverse function of F. Note that $\Omega_0 = -\infty$. The `Gen` procedure is written as $s = Q(x)$, $w = g(x)$, where the Q and g functions are given by

$$Q(x) = \max\{\alpha \in \mathcal{S} : \; x \geqslant \Omega_\alpha\}; \quad g(x) = \frac{F(x) - F(\Omega_{Q(x)})}{p_{Q(x)}} = \frac{F(x) - \sum_{i=0}^{Q(x)-1} p_i}{p_{Q(x)}} \tag{2.4}$$

The relation between x, s and w can be written in a more friendly form as

$$F(x) = F(\Omega_s) + w p_s = \sum_{i=0}^{s-1} p_i + w p_s. \tag{2.5}$$

The thus defined $\widetilde{w} \in [0, 1)$ is called *quantile* helper data since it measures which quantile of the probability mass p_s is located between $F(\Omega_s)$ and x. It was shown that the random variable W, given S, has a uniform pdf. Consequently the scheme is a ZLHDS.
The mapping of x to $(s, \widetilde{w})$ is a bijection. For the mapping of $(s, \widetilde{w})$ to x the following notation is used,[2]

$$\xi_{s,\widetilde{w}} \stackrel{\text{def}}{=} F^{\text{inv}}\Big(\sum_{i=0}^{s-1} p_i + \widetilde{w} p_s\Big). \tag{2.6}$$

In the case of the Fuzzy Extractor ($p_\alpha = 1/N$ for all $\alpha \in \mathcal{S}$) the optimal reconstruction procedure was found to be the following maximum-likelihood 'decoder',

$$\hat{s} = \mathtt{Rep}^{\text{FE}}(y, \widetilde{w}) = \arg\max_{\alpha \in \mathcal{S}} \psi(y | \xi_{\alpha\widetilde{w}}). \tag{2.7}$$

Eq. (2.7) can be conveniently implemented by defining decision boundaries $(\tau_{\alpha\widetilde{w}})_{\alpha=0}^{N}$. If $y \in [\tau_{\alpha\widetilde{w}}, \tau_{\alpha+1,\widetilde{w}})$, then $\hat{s} = \alpha$. In the case of symmetric fading noise the location of the decision boundaries dictated by (2.7) was found to be

$$\tau^{\text{FE}}_{\alpha\widetilde{w}} = \lambda \frac{\xi_{\alpha-1,\widetilde{w}} + \xi_{\alpha\widetilde{w}}}{2}. \tag{2.8}$$

Here one has to read $\xi_{-1,\widetilde{w}} = -\infty$ and $\xi_{N\widetilde{w}} = \infty$, resulting in $\tau_{0\widetilde{w}} = -\infty$, $\tau_{N\widetilde{w}} = \infty$. Fig. 2.2 shows how to intuitively understand (2.8). Each pdf

[2] We often omit the comma and write $\xi_{s\widetilde{w}}$.

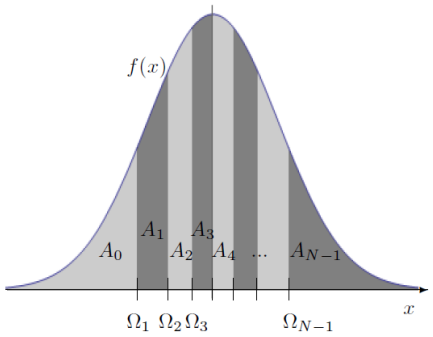


Figure 2.1: *Illustration of the quantization boundaries* Ω_α *and regions* A_α.

$\psi(y|\xi_{\alpha\tilde{w}})$ in (2.7) is centered around $y = \lambda\xi_{\alpha\tilde{w}}$ and drops off symmetrically. The crossing point where one α-value becomes more likely than another lies exactly halfway between the centers of two neighboring pdfs; such a crossing point is a decision boundary.

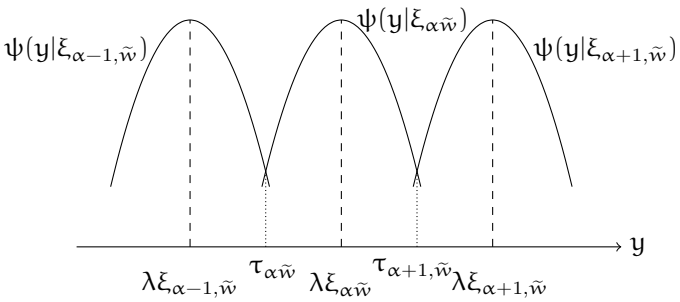


Figure 2.2: *Visual representation of the decision boundaries for the reconstruction phase.*

2.3 — Optimization of the general ZLHDS

In this section we extend the results of de Groot et al. [19]. We generalize equations (2.7) and (2.8). Then we derive analytic expressions for $I(S; \hat{S}|\widetilde{W})$ and the reconstruction error probability P_{err} in terms of the scheme's parameters. We also discuss the relation between P_{err} and the bit error rate.

2.3.1 – ZLHDS reconstruction. For the sake of completeness we explicitly show that $\widetilde{W}$ given $S = s$ is uniform. (This fact was implicit in [19] and was not

separately stated.)

Lemma 2.2. The probability density function of the helper data $\widetilde{W}$ given the secret S is uniform.

Proof. For the pdf of $\widetilde{W}$ given $S = \alpha$ we write $\rho(\widetilde{w}|\alpha)$. We start from $p_\alpha \rho(\widetilde{w}|\alpha) d\widetilde{w} = f(\xi_{\alpha\widetilde{w}}) d\xi_{\alpha\widetilde{w}}$. The validity of this equation is readily verified. Applying $\int_0^1$ to the left hand side yields p_α by definition; on the right hand side the equivalent operation is integration over $\xi_{\alpha\widetilde{w}}$ on the interval A_α, which also yields p_α. Now we can write $\rho(\widetilde{w}|\alpha) = \frac{f(\xi_{\alpha\widetilde{w}})}{p_\alpha d\widetilde{w}/d\xi_{\alpha\widetilde{w}}}$ $= \frac{f(\xi_{\alpha\widetilde{w}})}{dF(\xi_{\alpha\widetilde{w}})/d\xi_{\alpha\widetilde{w}}} = \frac{f(\xi_{\alpha\widetilde{w}})}{f(\xi_{\alpha\widetilde{w}})} = 1$. In the second equality we used (2.5) with $s = \alpha$ kept constant while $\widetilde{w}$ varies. □

Lemma 2.3. For the general HDS the optimal reconstruction procedure is given by

$$\hat{s} = \mathtt{Rep}(y, \widetilde{w}) = \arg\max_{\alpha \in \mathcal{S}} p_\alpha \psi(y|\xi_{\alpha\widetilde{w}}). \tag{2.9}$$

Proof. This is a slight modification of Lemma 4.1 in [19], with the same starting point.

$$\begin{aligned} \mathtt{Rep}(y, \widetilde{w}) &= \arg\max_{\alpha \in \mathcal{S}} \Pr[S = \alpha | Y = y, \widetilde{W} = \widetilde{w}] \\ &= \arg\max_{\alpha \in \mathcal{S}} \frac{\Pr[S = \alpha, Y = y, \widetilde{W} = \widetilde{w}]}{\Pr[Y = y, \widetilde{W} = \widetilde{w}]}. \end{aligned} \tag{2.10}$$

Since the denominator does not depend on α, it can be eliminated.

$$\begin{aligned} \mathtt{Rep}(y, \widetilde{w}) &= \arg\max_{\alpha \in \mathcal{S}} \Pr[S = \alpha, Y = y, \widetilde{W} = \widetilde{w}] \\ &= \arg\max_{\alpha \in \mathcal{S}} \Pr[Y = y | S = \alpha, \widetilde{W} = \widetilde{w}] \rho(w|\alpha) p_\alpha. \end{aligned}$$

Using Lemma 2.2 we get

$$\hat{s} = \mathtt{Rep}(y, \widetilde{w}) = \arg\max_{\alpha \in \mathcal{S}} p_\alpha \Pr[Y = y | S = \alpha, \widetilde{W} = \widetilde{w}]. \tag{2.11}$$

Since (α, w) uniquely defines $\xi_{\alpha\widetilde{w}}$, the probability $\Pr[Y = y | S = \alpha, \widetilde{W} = \widetilde{w}]$ equals $\Pr[Y = y | X = \xi_{\alpha\widetilde{w}}]$, for which the notation $\psi(y|\xi_{\alpha\widetilde{w}})$ is used. □

From (2.9) we can derive an optimal placement of the boundaries $\tau_{\alpha\widetilde{w}}$ for general noise and general HDS.

Lemma 2.4. For a ZLHDS the reconstruction boundary $\tau_{\alpha\tilde{w}}$ obtained using pdf intersections satisfies the following equation:

$$p_{\alpha-1}\psi(\tau_{\alpha\tilde{w}}|\xi_{\alpha-1,\tilde{w}}) = p_\alpha\psi(\tau_{\alpha\tilde{w}}|\xi_{\alpha\tilde{w}}). \tag{2.12}$$

Proof. From Lemma 2.3 we see that the decision boundary is the point y where the function $p_\alpha\psi(y|\xi_{\alpha\tilde{w}})$ intersects the function $p_{\alpha-1}\psi(y|\xi_{\alpha-1,\tilde{w}})$. □

In the FE case, $p_{\alpha-1} = p_\alpha$ and (2.12) reduces to $\psi(\tau_{\alpha\tilde{w}}|\xi_{\alpha-1,\tilde{w}}) = \psi(\tau_{\alpha\tilde{w}}|\xi_{\alpha\tilde{w}})$, which directly yields (2.8). In the general HDS case, however, the difference between the p_α parameters changes the heights of the pdfs $\psi(y|\cdots)$ in Fig. 2.2, which leads to a more complicated solution for the decision boundaries.

Theorem 2.5. *Let the noise be Gaussian with zero mean and variance σ_V^2. Then the intersection points as specified in (2.12) are given by*

$$\tau_{\alpha\tilde{w}} = \lambda\frac{\xi_{\alpha-1,\tilde{w}} + \xi_{\alpha\tilde{w}}}{2} + \frac{\sigma_V^2 \ln\frac{p_{\alpha-1}}{p_\alpha}}{\lambda(\xi_{\alpha\tilde{w}} - \xi_{\alpha-1,\tilde{w}})}. \tag{2.13}$$

Proof. The Gaussian noise is given by $\psi(y|x) = \frac{1}{\sqrt{2\pi}\sigma_V}e^{-\frac{(y-\lambda x)^2}{2\sigma_V^2}}$. Eq. (2.12) then becomes

$$\frac{p_{\alpha-1}}{\sqrt{2\pi}\sigma_V}e^{-\frac{(\tau_{\alpha w}-\lambda\xi_{\alpha-1,w})^2}{2\sigma_V^2}} = \frac{p_\alpha}{\sqrt{2\pi}\sigma_V}e^{-\frac{(\tau_{\alpha w}-\lambda\xi_{\alpha w})^2}{2\sigma_V^2}}. \tag{2.14}$$

Taking the logarithm on both sides of the equation yields a *linear* equation in $\tau_{\alpha\tilde{w}}$, with solution (2.13). □

Theorem 2.6. *Let the noise be Lorentz-distributed, $\psi(y|x) = \frac{1/\sigma_V}{1+\pi^2(y-\lambda x)^2/\sigma_V^2}$. Let $p_\alpha \neq p_{\alpha-1}$. If the following condition holds*

$$p_\alpha p_{\alpha-1}(\lambda\xi_{\alpha,\tilde{w}} - \lambda\xi_{\alpha-1,\tilde{w}})^2 \geqslant \sigma_V^2\frac{(p_\alpha - p_{\alpha-1})^2}{\pi^2}, \tag{2.15}$$

then the reconstruction boundary $\tau_{\alpha\tilde{w}}$ is given by

$$\begin{aligned}\tau_{\alpha\tilde{w}} &= \frac{p_{\alpha-1}\lambda\xi_{\alpha\tilde{w}} - p_\alpha\lambda\xi_{\alpha-1,\tilde{w}}}{p_{\alpha-1} - p_\alpha} \\ &- \frac{\sqrt{p_\alpha p_{\alpha-1}(\lambda\xi_{\alpha\tilde{w}} - \lambda\xi_{\alpha-1,\tilde{w}})^2 - \frac{\sigma_V^2}{\pi^2}(p_{\alpha-1} - p_\alpha)^2}}{p_{\alpha-1} - p_\alpha}.\end{aligned} \tag{2.16}$$

Proof. Substitution of the Lorentz distribution into (2.12) yields

$$\frac{p_\alpha}{1 + \pi^2\sigma_V^{-2}(\tau_{\alpha\tilde{w}} - \lambda\xi_{\alpha\tilde{w}})^2} = \frac{p_{\alpha-1}}{1 + \pi^2\sigma_V^{-2}(\tau_{\alpha\tilde{w}} - \lambda\xi_{\alpha-1,\tilde{w}})^2}. \tag{2.17}$$

Inversion of both sides of the equation gives a quadratic equation in $\tau_{\alpha\tilde{w}}$. (If $p_\alpha = p_{\alpha-1}$ then it reduces to a linear equation with (2.8) as the solution.) The quadratic equation has solutions only if the discriminant is non-negative, which is equivalent to the condition (2.15). Finally we have to choose the correct sign preceding the square root of the determinant. We choose the sign in such a way that $\lambda\xi_{\alpha-1,\tilde{w}} < \tau_{\alpha\tilde{w}} < \lambda\xi_{\alpha\tilde{w}}$. We verify as follows that (2.16) indeed satisfies these inequalities. On the one hand, (2.16) can be written as

$$\tau_{\alpha\tilde{w}} = \lambda\xi_{\alpha\tilde{w}} + \frac{p_\alpha\lambda(\xi_{\alpha\tilde{w}} - \xi_{\alpha-1,\tilde{w}}) - \sqrt{\cdots}}{p_{\alpha-1} - p_\alpha}. \tag{2.18}$$

Note that $\xi_{\alpha\tilde{w}} - \xi_{\alpha-1,\tilde{w}} > 0$. If $p_{\alpha-1} > p_\alpha$ then the $\sqrt{\cdots}$ 'wins' and the numerator of the fraction is negative, as it should be. If $p_{\alpha-1} < p_\alpha$ then the denominator is negative and the $\sqrt{\cdots}$ 'loses', making the numerator positive. On the other hand, (2.16) can also be written as

$$\tau_{\alpha\tilde{w}} = \lambda\xi_{\alpha-1,\tilde{w}} + \frac{p_{\alpha-1}\lambda(\xi_{\alpha\tilde{w}} - \xi_{\alpha-1,\tilde{w}}) - \sqrt{\cdots}}{p_{\alpha-1} - p_\alpha}. \tag{2.19}$$

If $p_{\alpha-1} > p_\alpha$ then the $\sqrt{\cdots}$ 'loses' and the fraction is positive. If $p_{\alpha-1} < p_\alpha$ then the $\sqrt{\cdots}$ 'wins' and the fraction is again positive. □

Remark. *If one adopts (2.13) as decision boundaries, an incorrect reconstruction procedure may result under some pathological circumstances. This can happen, for example, if for some α it happens that $p_\alpha \ll p_{\alpha-1}$ and $p_\alpha \ll p_{\alpha+1}$; then in Fig. 2.2 the middle curve is located beneath the intersection of its neighbors, and $\hat{s}$ cannot equal α even if $s = \alpha$. In practice we will never see this pathological case.*

2.3.2 – Optimization of the quantization intervals. As announced in Section 2.1.3, we want to maximize the amount of key material extracted from X by the ZLHDS. We have to take into account two effects: the noise, which limits how much of the entropy of X can be recovered in the reconstruction phase, and the fact that the adversary knows $\widetilde{W}$. The quantity of interest is the mutual information between S and $\hat{S}$ given $\widetilde{W}$: $I(S;\hat{S}|\widetilde{W})$. This represents the 'secrecy capacity' or quality of the channel from S to $\hat{S}$ created by the ZLHDS. If a perfect error correction mechanism is used as the second-stage HDS, i.e. one that achieves the Shannon bound, then $I(S;\hat{S}|\widetilde{W})$ is the achievable key length. We note that even though $H(S|\widetilde{W}) = H(S)$, we have $I(S;\hat{S}|\widetilde{W}) \neq I(S;\hat{S})$ because $\hat{S}$ is not independent of $\widetilde{W}$.

Lemma 2.7. For a zero leakage helper data system the mutual information can be expressed as

$$I(S;\hat{S}|\widetilde{W}) = H(S) - H(S|\hat{S},\widetilde{W}) = I(S;\hat{S},\widetilde{W}). \tag{2.20}$$

Proof. We write $I(S;\hat{S}|\widetilde{W}) = H(S|\widetilde{W}) - H(S|\hat{S},\widetilde{W})$. Due to the ZL property it holds that $H(S|\widetilde{W}) = H(S)$. □

The mutual information $I(S;\hat{S}|\widetilde{W})$ can be seen as a function of the system parameters $p_0,\dots,p_{N-1}$. These parameters completely fix the `Gen` and `Rep` procedures. (The λ, σ_X and σ_V are given by nature and cannot be chosen). Hence we want to determine how to set vector $(p_\alpha)_{\alpha\in\mathcal{S}}$ as a function of λ, σ_X, σ_V so as to maximize our target function. Unfortunately, $I(S;\hat{S}|W)$ depends on the p_α parameters in a very complicated way. The `Gen` is simple enough, but the `Rep` procedure has decision boundaries $\tau_{\alpha\tilde{w}}$ (2.12) that depend on $p_0,\dots,p_{N-1}$ not only directly but also via the $\xi_{\alpha\tilde{w}}$ points as specified in (2.6); this dependence is quite convoluted as the $\xi_{\alpha\tilde{w}}$ invoke the non-smooth step-wise function Q as well as the nonlinear F^{inv}. Analytic maximization of $I(S;\hat{S}|\widetilde{W})$ is intractable. It is clear, however, that a maximum must exist. Consider the ZLFE at fixed $N \geqslant 3$. Not all intervals A_α have equal width, which leads to unequal probabilities for jumping from one interval to another due to noise. Making the narrowest intervals slightly broader reduces the reconstruction error probability (with a positive effect on our target function) and the entropy of S (with a negative effect). It is intuitively clear that at large σ_V the effect of reconstruction errors weighs more heavily than the $H(S)$ effect; then we expect a nontrivial maximum at a p_α setting different from the FE's $p_\alpha = 1/N$. The numerics in Section 2.4 show that this is indeed the case.

For the efficiency of the numerical optimization we now look for a simple form in which to represent $I(S;\hat{S}|\widetilde{W})$. We introduce the following notation,

$$\begin{aligned}\Upsilon_{\hat{s}|s\tilde{w}} &\stackrel{\text{def}}{=} \Pr[\hat{S}=\hat{s}|S=s,\widetilde{W}=\tilde{w}] = \int_{\tau_{\hat{s}\tilde{w}}}^{\tau_{\hat{s}+1,\tilde{w}}} \psi(y|\xi_{s\tilde{w}})\,dy \\ &= V(\tau_{\hat{s}+1,\tilde{w}} - \lambda\xi_{s\tilde{w}}) - V(\tau_{\hat{s}\tilde{w}} - \lambda\xi_{s\tilde{w}}). \end{aligned} \tag{2.21}$$

We can express the mutual information entirely in terms of the p_α and $\Upsilon_{\hat{s}|s\tilde{w}}$ parameters.

Lemma 2.8. For the ZLHDS the mutual information can be written as

$$I(S;\hat{S}|\widetilde{W}) = \sum_{s=0}^{N-1}\sum_{\hat{s}=0}^{N-1}\int_0^1 d\tilde{w}\; p_s \Upsilon_{\hat{s}|s\tilde{w}} \log \frac{\Upsilon_{\hat{s}|s\tilde{w}}}{\sum_{\beta=0}^{N-1} p_\beta \Upsilon_{\hat{s}|\beta\tilde{w}}}. \tag{2.22}$$

Proof.

$$\begin{aligned} I(S;\hat{S}|\widetilde{W}) &= \mathbb{E}_{s\hat{s}\widetilde{w}} \log \frac{\Pr[S=s,\hat{S}=\hat{s}|\widetilde{W}=\widetilde{w}]}{\Pr[S=s|\widetilde{W}=\widetilde{w}]\Pr[\hat{S}=\hat{s}|\widetilde{W}=\widetilde{w}]} \\ &= \mathbb{E}_{\widetilde{w}} \sum_{s,\hat{s}=0}^{N-1} \Pr[S=s|\widetilde{W}=\widetilde{w}] \\ &\quad \Upsilon_{\hat{s}|s\widetilde{w}} \log \frac{\Pr[S=s,\hat{S}=\hat{s}|\widetilde{W}=\widetilde{w}]}{\Pr[S=s|\widetilde{W}=\widetilde{w}]\Pr[\hat{S}=\hat{s}|\widetilde{W}=\widetilde{w}]}. \end{aligned} \tag{2.23}$$

In the last line we used the chain rule $\Pr[S=s,\hat{S}=\hat{s},\widetilde{W}=\widetilde{w}] = \mathbb{E}_w \Pr[S=s|\widetilde{W}=\widetilde{w}]\Upsilon_{\hat{s}|s\widetilde{w}}$. Next we use $\mathbb{E}_{\widetilde{w}}(\cdots) = \int_0^1 d\widetilde{w}(\cdots)$ as implied by Lemma 2.2, and $\Pr[S=s|\widetilde{W}=\widetilde{w}] = p_s$ by the ZL property. Finally we apply these rules, and $\Pr[\hat{S}=\hat{s}|\widetilde{W}=\widetilde{w}] = \sum_s p_s \Upsilon_{\hat{s}|s\widetilde{w}}$, inside the logarithm. □

2.3.3 – Reconstruction errors. While we are mainly interested in the mutual information, we also care about the practical implementation aspects of the second-stage HDS. The second-stage HDS typically employs an Error-Correcting Code (ECC). If the output of the first-stage HDS has a high bit error rate, this causes problems for the ECC. In our numerics we keep track of the error rate. We write $P_{err} = \Pr[\hat{S} \neq Q(X)]$ for the overall probability that $\hat{S}$ is not equal to S. This is an averaged quantity, i.e. averaged over X. For fixed x we have

$$\Pr[\hat{S}=Q(X)|X=x] = \Upsilon_{Q(x)|Q(x),g(x)}. \tag{2.24}$$

Averaging over x gives

$$1-P_{err} = \mathbb{E}_x \Pr[\hat{S}=Q(X)|X=x] = \mathbb{E}_x \Upsilon_{Q(x)|Q(x),g(x)} = \sum_{s\in\mathcal{S}} p_s \int_0^1 d\widetilde{w}\, \Upsilon_{s|s\widetilde{w}}. \tag{2.25}$$

In the last step we used that x uniquely maps to $(s,\widetilde{w}) = (Q(x), g(x))$. Eq. (2.25) together with (2.21) is the most convenient way to analytically express the reconstruction error probability.

We consider the case where s is encoded as a Gray code. This is a well known technique to reduce the number of bit flips when a reconstruction error occurs. Table 2.1 lists the Gray code that we use. (Other, equivalent, encodings are possible.) We will look at $N \in \{3,4,5,6\}$. The length of the Gray code is $\lceil \log N \rceil$ bits.

Table 2.1: *Three-bit Gray code used for* $N = 5$ *and* $N = 6$. *The highlighted cell shows the two-bit Gray code that we use for* $N = 3$ *and* $N = 4$.

s	1st bit	2nd bit	3rd bit
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1

The Bit Error Rate (BER) is given by

$$\mathrm{BER} = \frac{\mathbb{E}[\#\text{ bit errors}]}{\lceil \log N \rceil} = \frac{1}{\lceil \log N \rceil} \sum_{t=0}^{\lceil \log N \rceil} t \Pr[\#\text{ bit errors} = t]. \tag{2.26}$$

We introduce the following notation,

$$\Delta_{\hat{s}|s} \stackrel{\text{def}}{=} \Pr[\hat{S} = \hat{s} | S = s] = \mathbb{E}_{\tilde{w}} \Upsilon_{\hat{s}|s\tilde{w}}. \tag{2.27}$$

All the probabilities in (2.26) can be calculated in terms of the $\Delta_{\hat{s}|s}$ probabilities. The details are given in the Appendix.

2.4 — Numerical results

We present numerical results for the optimization described in Section 2.3, for $N \in \{3, 4, 5, 6\}$. We consider a Gaussian source X and Gaussian noise. (This is already a rather accurate model for Coating PUFs [89]). Without loss of generality we set $\sigma_X = 1$. Only the ratio σ_V/σ_X matters. We consider the two cases defined in Section 2.2.3: *perfect enrollment* and *identical conditions*. We implemented (2.22) in Wolfram Mathematica 10.2 as a symbolic function. We used the built-in function `FindMaximum` to obtain optimum values for $p_0, \ldots, p_{N-1}$. In order to reduce the dimension of the search space we imposed the symmetry $p_{N-1-\alpha} = p_\alpha$ by hand.
Fig. 2.3 shows $I(S; \hat{S}|W)$ versus P_{err} for various σ_V.

- When σ_V is small, the optimum setting of the HDS is close to the FE setting $p_\alpha = 1/N$, and it is clearly visible that increasing N has a very large benefit for the mutual information.
- For somewhat larger σ_V, there is a clear difference between the optimized HDS and the FE. For example, in the $\lambda = 1$ graph at $\sigma_V = 0.25$ we see that at $N = 6$ the transition from FE to HDS brings a modest improvement

of the mutual information and a reduction of P_{err} from $\approx 23\%$ to $\approx 10\%$. The reduced P_{err} means that the ECC in the second stage is much easier to implement for the HDS than for the FE.

- At $\sigma_V > 0.5$ the noise is so bad that the HDS and the FE perform almost equally badly (though the HDS is always slightly better). Increasing N improves the mutual information only slightly, and at the cost of a large increase in P_{err}.

Fig. 2.4 shows the same data, but with the BER on the horizontal axis. The 'zigzag' at the transition from $N = 4$ to $N = 5$ occurs because the Gray code jumps from a 2-bit representation of s to a 3-bit representation, with little noise in the first of the three bits.

Fig. 2.5 shows the BER as a function of σ_V/σ_X. The curves for $N = 4$ and $N = 5$ cross each other; this causes the 'zigzag' in Fig. 2.4. The graphs of P_{err} as a function of σ_V/σ_X (Fig. 2.6) are much smoother. For completeness Fig. 2.7 plots the BER versus P_{err}. The relation is clearly nonlinear.

Fig. 2.8 shows the optimal values of $p_0, \ldots, p_{N-1}$ for the perfect enrollment case ($\lambda = 1$). At $\sigma_V = 0$ it holds that $p_\alpha = 1/N$ for all α, which is the FE configuration. When σ_V increases, the outer regions A_0, A_{N-1} shrink while the central region(s) become broader. Then at some point this trend reverses. At very large σ_V the p_α values stabilize, but not in the FE configuration.

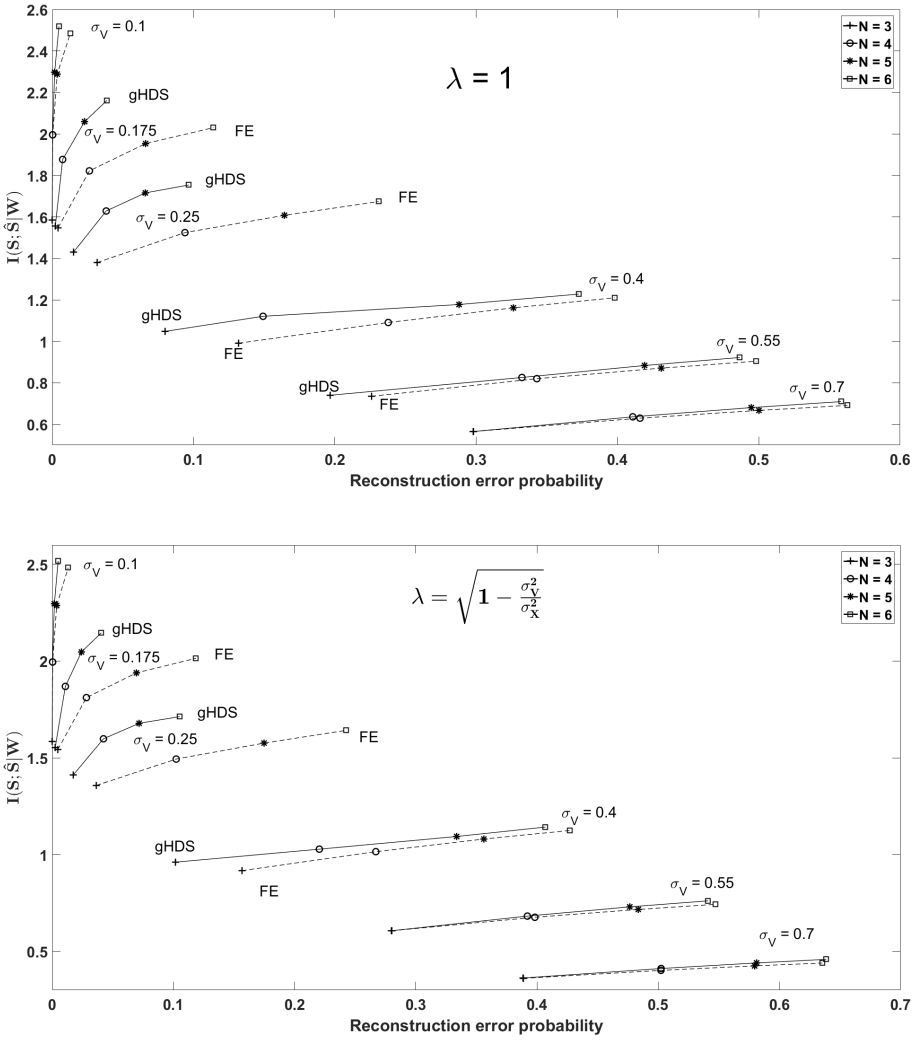


Figure 2.3: *Mutual information versus* P_{err} *for perfect enrollment (upper figure) and identical conditions (lower figure). At fixed* σ_V*, data points for the general HDS are connected with a solid line, while a dashed line corresponds to the FE.*

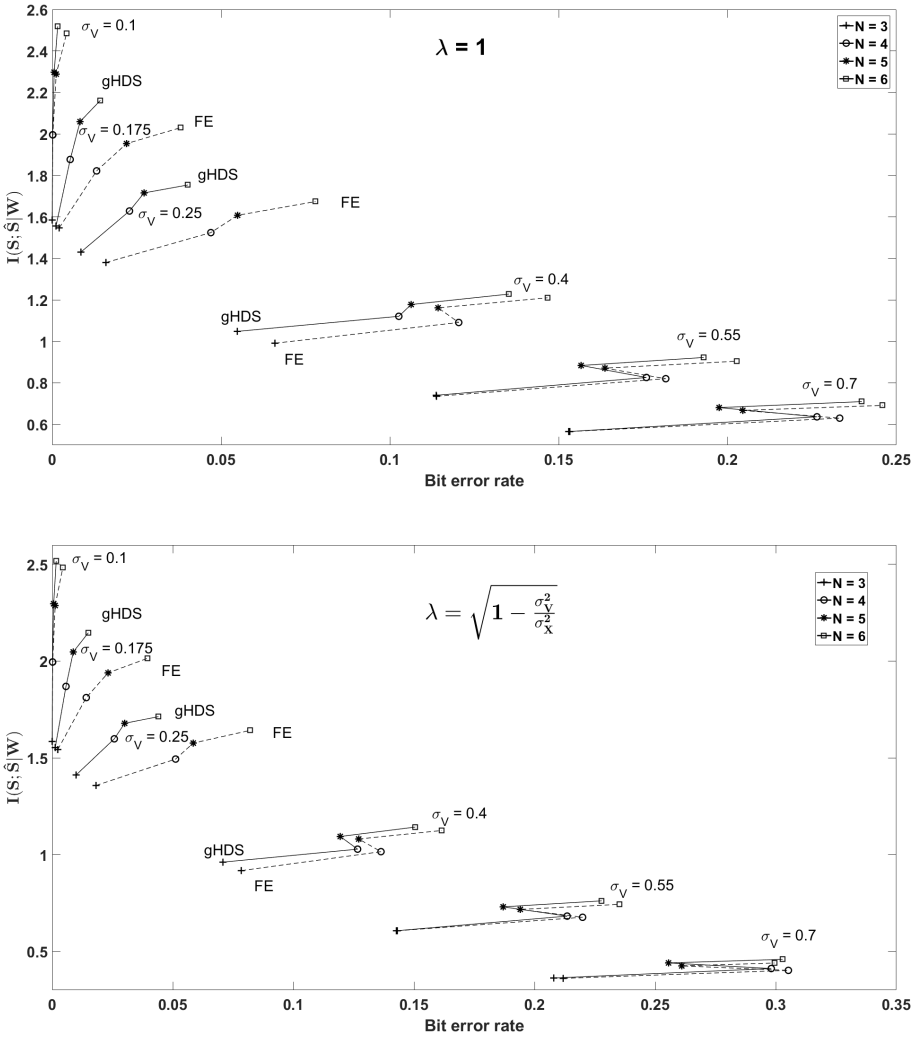


Figure 2.4: *Mutual information versus BER for perfect enrollment (upper figure) and identical conditions (lower figure). At fixed* σ_V*, data points for the general HDS are connected with a solid line, while a dashed line corresponds to the FE.*

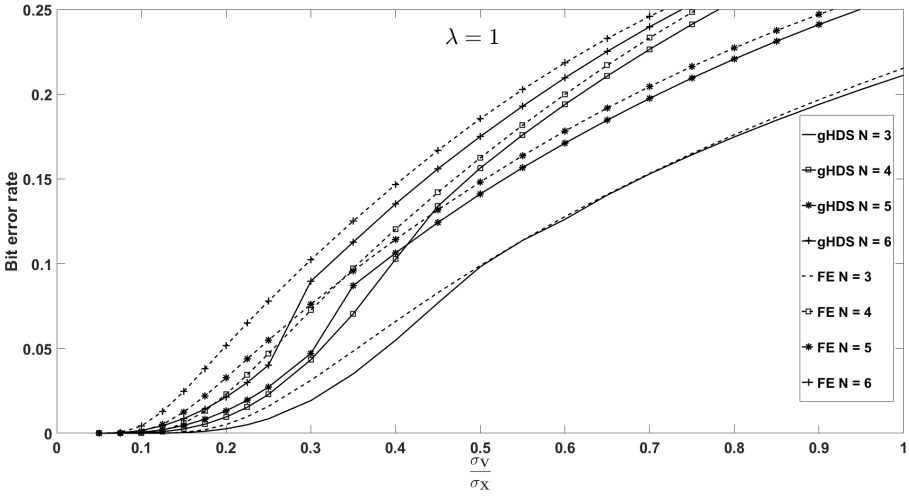


Figure 2.5: *Bit Error Rate as a function of the noise parameter* σ_V/σ_X. *Perfect enrollment*

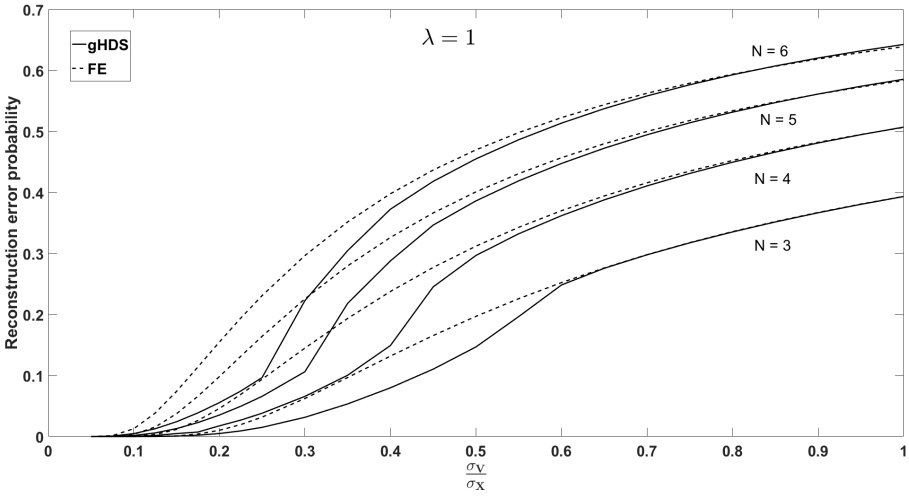


Figure 2.6: P_{err} *as a function of the noise parameter* σ_V/σ_x. *Perfect enrollment.*

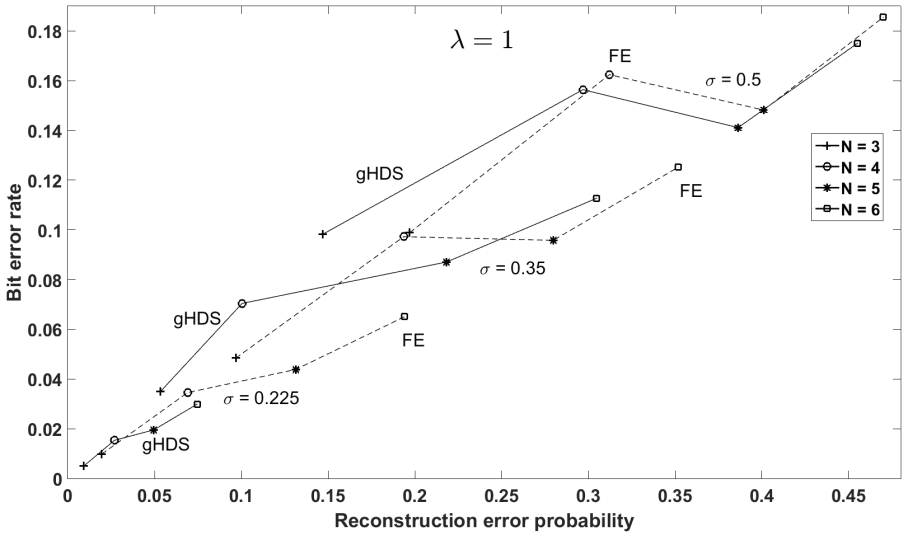


Figure 2.7: *BER versus reconstruction error probability* $\mathsf{P}_{\mathrm{err}}$. *Perfect enrollment. At given* σ_{V}, *data points for the HDS are connected with a solid line, while a dashed line corresponds to the FE.*

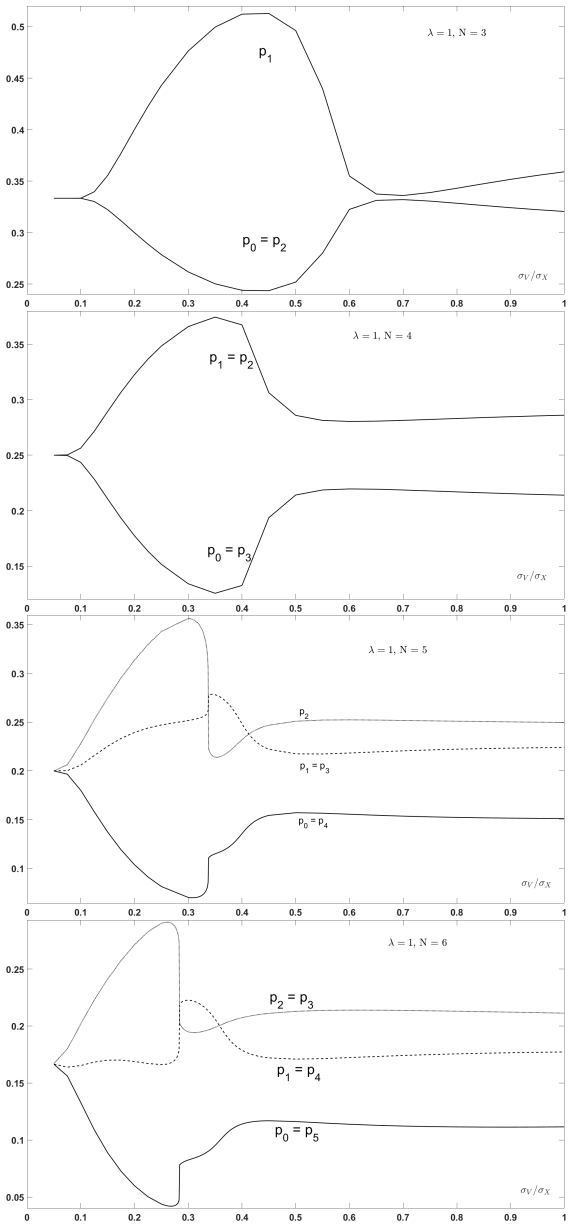


Figure 2.8: *The* p_α *values as a function of the noise parameter* $\sigma_\mathsf{V}/\sigma_\mathsf{X}$*, for* $\lambda = 1$*,* $\mathsf{N} = 3, 4, 5, 6$.

2.5 — Summary

We have extended the results of de Groot et al. [19] in the case of non-equiprobable quantization intervals. Lemma 2.4 gives the recipe for finding the optimal decision boundaries used in `Rep`. The result for Gaussian and Lorentzian noise is given in Theorems 2.5 and 2.6.

We have studied the mutual information $I(S;\hat{S}|\widetilde{W})$, which is an upper bound on the amount of secret key material that can be robustly extracted from X. The mutual information is most conveniently expressed in terms of the p_s and $\Upsilon_{\hat{s}|s\tilde{w}}$ parameters (2.22). The dependence of the $\Upsilon_{\hat{s}|s\tilde{w}}$ on $p_0, \ldots, p_{N-1}$ is so complicated that optimization of $I(S;\hat{S}|\widetilde{W})$ cannot be done analytically. The figures in Section 2.4 show the results of numerical optimization in a simple model where the source and the noise are Gaussian. Such a model is reasonably accurate for Coating PUFs. For every combination $(N, \sigma_V/\sigma_X)$ the optimized ZLHDS clearly performs better than the ZLFE in terms of both mutual information and reconstruction error rate. The reduction in P_{err} is substantial. This makes the design of a second-stage HDS much more practical, since it makes is easier to implement an ECC that can cope with the bit errors introduced by reconstruction errors.

As future work we will apply the numerical optimization to different source distributions, matching e.g. biometric data.

Appendix: Bit error rates

We list expressions for the BER (2.26) in terms of the $\Delta_{\hat{s}|s}$ probabilities (2.27), when the Gray code of Table 2.1 is used. We assume a symmetric source pdf f and symmetric noise. As a result the optimal p_α values have the symmetry $p_{N-1-\alpha} = p_\alpha$, and there is a large number of symmetries between the $\Delta_{\ldots}$ values, $\Delta_{N-1-\hat{s}|N-1-s} = \Delta_{\hat{s}|s}$.

N	N·BER
3	$2p_0(\Delta_{1\|0} + 2\Delta_{2\|0}) + 2p_1\Delta_{2\|1}$
4	$2p_0(\Delta_{1\|0} + \Delta_{3\|0} + 2\Delta_{2\|0}) + 2p_1(\Delta_{0\|1} + \Delta_{2\|1} + 2\Delta_{3\|1})$
5	$2p_0(\Delta_{1\|0} + \Delta_{3\|0} + 2\Delta_{2\|0} + 2\Delta_{4\|0}) + 2p_1(\Delta_{0\|1} + \Delta_{2\|1} + 2\Delta_{3\|1} + 3\Delta_{4\|1})$ $+2p_2(\Delta_{1\|2} + 2\Delta_{0\|2})$
6	$2p_0(\Delta_{1\|0} + \Delta_{3\|0} + 2\Delta_{2\|0} + 2\Delta_{4\|0} + 3\Delta_{5\|0})$ $+2p_1(\Delta_{0\|1} + \Delta_{2\|1} + 2\Delta_{3\|1} + 2\Delta_{5\|1} + 3\Delta_{4\|1})$ $+2p_2(\Delta_{1\|2} + \Delta_{3\|2} + 2\Delta_{0\|2} + 2\Delta_{4\|2})$

The p-index in this table runs only to $\lceil N/2 \rceil - 1$ because of the $\alpha \leftrightarrow N-1-\alpha$ symmetry; this also gives rise to the factor 2 in front of each p_α. Inside the parentheses, the numerical factor in front of each Δ indicates the number of bit flips that occur due to that specific transition.

2

Chapter 3

Minutia-pair spectral representations for fingerprint template protection

In this chapter we want to address the challenge of appearance/disappearance of minutiae. As mentioned in Chapter 1 the number of detectable minutiae is often different on every image. This is an important issue since we want to provide template protection of fingerprints by using one-way hash functions. This demands the use of an error-correction code, which consequently requires a fixed-length representation. Generally speaking if one compares two images of a finger, not only the number of minutiae is different, the images are usually misaligned in terms of different rotation angle, scaling ratio, translation, etc. This should be addressed already at the stage of fixed-length representation (before quantization), since it is very hard to address the misalignment at the later stages.

We introduce a spectral function based on minutiae pairs. By using minutiae-pair approach translation invariance comes automatically. We do not have to discard the phase information of the spectral function as Xu et al. to obtain translation invariance. The initial idea was to take the absolute value of the pair-wise spectral function in order to obtain complete translation, rotation and scaling invariance. However this approach turned out to work badly in practice even for good quality fingerprints. Since for most fingerprint databases the scaling factor is close to one, we decided to neglect scaling and substitute the radial function by a Gaussian kernel. The final representation is not invariant under rotation and scaling; however, in practice modest rotation of the verification image hardly affects the performance.

3.1 — Introduction

3.1.1 – Privacy-preserving storage of biometric data. Biometrics-based authentication has become popular because of its great convenience. Biometrics cannot be forgotten or accidentally left at home. While biometric data is not strictly speaking secret (we are after all leaving a trail of fingerprints, DNA etc. behind us), it is important to protect biometric data for various reasons, the most important of which is privacy. Unprotected storage of biometric data would reveal medical conditions and would allow for cross-matching entries in different databases. Furthermore, large-scale availability of biometric data would make it easier for malevolent parties to leave misleading traces at at crime scene. (E.g. artificial fingerprints [61], synthesized DNA [28].)
One of the easiest ways to properly protect a biometric database against breaches and insider attacks is to store biometrics in *hashed* form, just like passwords, but with the addition of an error-correction step to get rid of the measurement noise. To prevent critical leakage from the error correction redundancy data, one uses a *Helper Data System* (HDS) [19, 54, 81], for instance a *Fuzzy Extractor* or a *Secure Sketch* [14, 22, 47].
A HDS typically makes use of an error-correcting code and hence needs a *fixed-length representation* of the biometric. Such a representation is not straightforward when the measurement noise can cause features of the biometric to appear or disappear, due to e.g. occlusion of iris areas or fuzziness of fingerprint minutiae. A very useful fixed-length representation called *spectral minutiae* was introduced by Xu et al. [102–105]. A Fourier-like spectral function is built up on a fixed discrete grid, in such a way that each detected fingerprint minutia adds a contribution to the function. Comparison of spectral functions is robust against changes in the number of available biometric features.

3.1.2 – Contributions and outline. We have the following results regarding spectral representations of fingerprint minutiae.

- We introduce spectral functions based on pairs of minutiae. By working with coordinate *differences* we immediately obtain a translation-invariant representation. Whereas Xu et al.'s spectral functions have to discard phase information in order to achieve translation invariance, our method retains phase information.
- We test our pair-based spectral minutiae matching technique on two fingerprint databases. The achieved Equal Error Rate is comparable to Xu et al.
- Our fingerprint matching is faster even though we have to sum over minutia pairs instead of individual minutiae. The speedup is due to the fact that we need fewer grid points on which to compute the spectral function.
- A further speedup can be obtained by skipping one laborious step in the verification procedure: rotating the fingerprint so as to obtain optimal

alignment with the enrolled fingerprint. Skipping this step leads only to a minimal penalty in terms of False Acceptance Rate and False Rejection Rate.

In Section 4.3 we briefly review Helper Data Systems and spectral minutiae functions. In Section 3.3 we discuss the drawbacks of Xu et al.'s spectral minutiae technique. We introduce our minutia pair approach in Section 3.4, and we study its fingerprint matching performance in Section 4.6. Section 3.6 discusses the computational efficiency of the verification procedure.

3.2 — Preliminaries

3.2.1 – Notation and terminology. We denote the number of minutiae found in a fingerprint by Z. The coordinates of the j'th minutia are x_j, y_j and its orientation is θ_j. Let f be a function of two real-valued arguments. The two-dimensional Fourier transform $\tilde{f} = \mathcal{F}f$ is defined as $\tilde{f}(k_x, k_y) = \int_{-\infty}^{\infty} f(x,y) e^{-ik_x x - ik_y y} dx dy$. The inverse relation $f = \mathcal{F}^{-1}\tilde{f}$ is given by
$f(x,y) = (\frac{1}{2\pi})^2 \int_{-\infty}^{\infty} \tilde{f}(k_x, k_y) e^{ik_x x + ik_y y} dk_x dk_y$.
The complex conjugate of $z \in \mathbb{C}$ is written as z^*. The hermitian conjugate $M^\dagger$ of a matrix M is given by $(M^\dagger)_{ij} = M^*_{ji}$. The inner product of two complex vectors u, v is $\langle u, v \rangle = u^\dagger v$. The Pearson correlation coefficient of two length-n vectors is defined as $\rho(u,v) = \frac{1}{n} \langle \frac{u - u_{av}}{\sigma_u}, \frac{v - v_{av}}{\sigma_v} \rangle$, where $u_{av} = \frac{1}{n}\sum_i u_i$ and $\sigma_u^2 = \frac{1}{n}\sum_i |u_i - u_{av}|^2$.
We will use the abbreviations FR = False Reject, FRR = False Reject Rate, FA = False Accept, FAR = False Accept Rate, EER = Equal Error Rate, ROC = Receiver Operating Characteristic.

3.2.2 – Helper Data Systems. A Helper Data System (HDS) for a (possibly non-discrete) source consists of two functions, `Gen` and `Rep`. Given an enrollment measurement X of the source, `Gen` produces redundancy data $W \in \{0,1\}^*$ called *helper data* and a secret string S. The helper data is stored. The storage is considered insecure, i.e. attackers learn W. At some later time, a verification measurement is performed, yielding outcome $X' \approx X$ which is a noisy version of X. The `Rep` function takes as input X' and W. It outputs an estimator $\hat{S}$ which should equal S if the noise was not excessive. In a general HDS, there is no constraint on the distribution of S. A desirable property is that S has high entropy given W.
A HDS is the perfect primitive for privacy protection of biometric databases against inside attackers and intruders, who typically obtain access not only to stored data but also to decryption keys. The HDS creates a noiseless secret and thus makes it possible to protect biometric secrets in the same way as passwords: by hashing. For every enrolled user, the database contains W and a hash $\chi(S)$. In the verification phase, the hash of the reconstructed $\hat{S}$ is compared against the stored $\chi(S)$. Ideally, W contains just enough information to allow for the

error correction, and does not leak any privacy-sensitive information about the raw biometric X. Furthermore, if χ is a properly chosen one-way function and S has enough entropy given W, the hash value $\chi(S)$ does not reveal S.
HDSs for discrete sources [7, 10, 14, 22, 47] and continuum sources [19, 54, 81, 96] are a well studied topic. Typically a HDS uses an error correcting code, which requires that the biometric measurement is turned into a discrete fixed-length representation.

3.2.3 – Spectral representation of minutiae. Subsequent measurements of the same finger may not always result in the same set of observed minutiae. This is problematic if one needs a fixed-length representation of a fingerprint, e.g. when a HDS is used. The technique of *spectral minutiae* was introduced by Xu et al. [102, 103, 105] as a way to obtain a fixed-length representation. The set of enrolled minutiae is turned into a function $f_\sigma(x, y)$ on the xy-plane by summing narrow Gaussian peaks (with width σ) centered on the minutia locations; then a translation-invariant expression g_σ is obtained by taking the absolute value of the Fourier transform,

$$g_\sigma(k_x, k_y) = |\tilde{f}_\sigma(k_x, k_y)| = e^{-\frac{\sigma^2}{2}(k_x^2+k_y^2)} \left| \sum_{j=1}^{Z} e^{-ik_x x_j - ik_y y_j} \right|. \tag{3.1}$$

In order to get an expression with simple behavior under rotation and scaling, they sampled g_σ on a log-polar grid. Let $k_x(\alpha, \beta) = e^\alpha \cos\beta$ and $k_y(\alpha, \beta) = e^\alpha \sin\beta$ where α, β are sampled with equal spacing. A matrix G^σ is constructed as $G^\sigma_{\alpha\beta} = g_\sigma(k_x(\alpha, \beta), k_y(\alpha, \beta))$. Under the combination of scaling and rotation, $\binom{x_j}{y_j} \mapsto \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \binom{\lambda x_j}{\lambda y_j}$ for all j, the G^σ transforms as $G^\sigma_{\alpha\beta} \mapsto G^{\sigma/\lambda}_{\alpha+\ln\lambda, \beta+\varphi}$. For small σ it holds that $\sigma/\lambda \approx \sigma$ and hence the transform is almost equal to a shift on the $\alpha\beta$-grid.[1] Xu et al. investigated fingerprint matching in the spectral minutiae domain by looking at the Pearson correlation between a freshly obtained G^σ and the enrolled G^σ. Their procedure included a search to find values λ, φ that maximize the correlation. It turned out that in practice one can fix $\lambda = 1$ and that the φ-search can be restricted to the interval from $-10°$ to $+10°$, in steps of $2°$.
In order to extract more information from a fingerprint Xu et al introduced a variant of the g_σ function which contains information about the minutia orientations θ_j. They inserted a factor $(k_x \cos\theta_j + k_y \sin\theta_j)$ or $e^{i\theta_j}$ into the summation in g_σ (3.1). Unsurprisingly, using information from both the ordinary G^σ representation and the orientation-containing variant yielded better results (in terms of e.g. ROC curves and EER) than using only a single representation.

[1] The effect on σ was not explicitly mentioned in the work of Xu et al.

Xu et al also investigated a minutiae representation that is fully invariant under translation, rotation and scaling. Let $H^\sigma = \mathcal{F}G^\sigma$ be the discrete Fourier transform of $G^\sigma_{\alpha\beta}$ with respect to α and β; then scalings and rotations have the effect of merely producing a phase factor multiplying H^σ; the absolute value $|H^\sigma|$ is fully invariant. However, it turned out that fingerprint matching in the $|H^\sigma|$-domain does not perform well.

3.3 — Motivation

The spectral minutiae technique as developed by Xu et al [102, 103, 105] has a number of unsatisfactory aspects.

1. Translation invariance is obtained by taking the absolute value of a Fourier transform. This step discards a lot of information.
2. Xu et al conclude that the scaling factor λ does not have to be taken into account, since it is always close to 1. But in their best fingerprint matching implementation they still apply logarithmic sampling in the radial k-direction, $\sqrt{k_x^2 + k_y^2} = e^\alpha$. Such sampling does not match the radial information density in the fingerprint and hence makes it necessary to take many many samples than in the case of linear sampling.
3. In combination with a HDS, the φ-search is time consuming. This is caused not by the repeated re-computation of the score, but by the fact that in a full HDS every φ-attempt needs an evaluation of the `Rep` function and the computation of a hash.

We address the first issue by introducing a spectral representation that is based on coordinate differences $\mathbf{x}_a - \mathbf{x}_b$ only. The advantage is immediate translation invariance without information loss, enabling us to work with fewer samples. The drawback is that each summation over Z minutiae is replaced by a summation over $\binom{Z}{2}$ pairs. The overall effect on the computation time during reconstruction is a trade-off between these two. In Section 3.6 we show that the trade-off works in our advantage.

We address the second issue by performing a Fourier transform *only in the angular direction*. In the radial direction our sampling occurs in the spatial domain and is linear.

The third issue could be addressed by developing a method to quickly determine the global orientation of a captured fingerprint image. (Knowledge of the global orientation, even if inaccurate, reduces the search space. Furthermore, storing the global orientation during enrollment as helper data does not leak sensitive information.) However, with our pair-based spectral representation it turns out that executing the φ-search yields only a very modest performance improvement; the search may as well be omitted. In Section 3.5.3 we show the difference in performance.

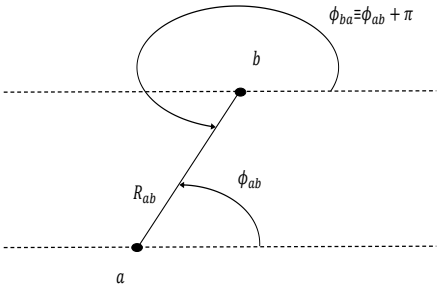


Figure 3.1: *Distance* R_{ab} *and angle* φ_{ab} *for a minutia pair.*

3.4 — The minutia-pair approach

3.4.1 – Definitions and properties. Let $R_{ab} = |\mathbf{x}_a - \mathbf{x}_b|$ and let $\tan\varphi_{ab} = (y_a - y_b)/(x_a - x_b)$ for minutiae $a, b \in \{1, \ldots, Z\}$. See Fig. 3.1. We define two translation-invariant spectral functions as follows

$$L_{\mathbf{x}}(q,\omega) \stackrel{\text{def}}{=} \sum_{\substack{a,b\in\{1,\ldots,Z\}\\ a\neq b}} e^{iq\varphi_{ab}} e^{i\omega \ln R_{ab}} \tag{3.2}$$

$$L_{\mathbf{x}\theta}(q,\omega) \stackrel{\text{def}}{=} \sum_{\substack{a,b\in\{1,\ldots,Z\}\\ a\neq b}} e^{iq\varphi_{ab}} e^{i\omega \ln R_{ab}} e^{i(\theta_a - \theta_b)}. \tag{3.3}$$

Here the subscript $\mathbf{x}$ denotes the set of minutia locations, and likewise θ stands for the set of minutia orientations. We call the functions $L_{\mathbf{x}}, L_{\mathbf{x}\theta}$ 'spectral' because (3.2) is the Fourier transform (with respect to the radial coordinate $\ln R$ and the angle φ) of a sum of delta functions centered on the values $\mathbf{x}_a - \mathbf{x}_b$ in the plane.

Let $\Phi = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$ be a rotation matrix. Our spectral functions (3.2),(3.3) have simple behaviour under the combined scaling and rotation $\mathbf{x}_j \mapsto \lambda\Phi\mathbf{x}_j$, $\theta_j \mapsto \theta_j + \varphi$,

$$L_{\lambda\Phi\mathbf{x}}(q,\omega) = e^{iq\varphi} e^{i\omega\ln\lambda} L_{\mathbf{x}}(q,\omega) \tag{3.4}$$

$$L_{\lambda\Phi\mathbf{x},\theta+\varphi}(q,\omega) = e^{iq\varphi} e^{i\omega\ln\lambda} L_{\mathbf{x}\theta}(q,\omega). \tag{3.5}$$

Note that the absolute values $|L_{\mathbf{x}}(q,\omega)|$, $|L_{\mathbf{x}\theta}(q,\omega)|$ are invariant under translation, scaling and rotation. Without giving details we mention that, unfortunately, fingerprint matching based on $|L_{\mathbf{x}}|$, $|L_{\mathbf{x}\theta}|$ without the phase information performs badly.

Similar to Xu et al we need to sample ω at equally spaced steps in order to exploit the phase behaviour (3.4),(3.5) under scaling. However, if we choose to

ignore scaling entirely (see point 2 in Section 3.3), then there is no reason to Fourier transform the radial direction, and we introduce an alternative spectral function,

$$M_x(q,R) \stackrel{\text{def}}{=} \sum_{\substack{a,b\in\{1,\dots,Z\}\\ a\neq b}} e^{iq\varphi_{ab}} \exp\left[-\frac{(R-R_{ab})^2}{2\sigma^2}\right] \tag{3.6}$$

$$M_{x\theta}(q,R) \stackrel{\text{def}}{=} \sum_{\substack{a,b\in\{1,\dots,Z\}\\ a\neq b}} e^{iq\varphi_{ab}} \exp\left[-\frac{(R-R_{ab})^2}{2\sigma^2}\right] e^{i(\theta_a-\theta_b)}. \tag{3.7}$$

In the radial direction, the functions M_x and $M_{x\theta}$ consist of a sum of $\binom{Z}{2}$ Gaussian peaks centered on the values R_{ab}. The width $\sigma > 0$ reduces the scheme's sensitivity to small perturbations in the minutia properties.
Under a rotation ($x_j \mapsto \Phi x_j$, $\theta_j \mapsto \theta_j + \varphi$) we have $M_x(q,R) \mapsto e^{iq\varphi}M_x(q,R)$ and $M_{x\theta}(q,R) \mapsto e^{iq\varphi}M_{x\theta}(q,R)$. We want all our spectral functions to be single-valued.[2] Hence q always has to be integer.

Lemma 3.1. *For odd* q *it holds that* $L_x(q,\omega) = 0$ *for all* ω*, and* $M_x(q,R) = 0$ *for all* R*.*

Proof. In (3.2) every pair of indices a, b gives two terms in the summation. Using $R_{ba} = R_{ab}$ and $\varphi_{ba} \equiv \varphi_{ab}+\pi \mod 2\pi$ (see Fig. 3.1), we write $e^{iq\varphi_{ab}}e^{i\omega \ln R_{ab}} + e^{iq\varphi_{ba}}e^{i\omega \ln R_{ba}} = e^{iq\varphi_{ab}}e^{i\omega \ln R_{ab}}[1 + e^{iq\pi}] = e^{iq\varphi_{ab}}e^{i\omega \ln R_{ab}}[1 + (-1)^q]$. This vanishes when q is odd. The proof for M_x is analogous. □

3.4.2 – Choosing the grid points. We have to choose a discrete (q,w)-grid of points on which to evaluate L_x and $L_{x\theta}$. On the one hand, the grid should contain many points so that the spectral functions contain sufficient information about the fingerprint. On the other hand, having too many grid points results in an inefficient scheme. Lemma 3.1 tells us that we do not have to compute L_x for odd q. Furthermore, we know that, at a given q, the spectral functions detect angular periodic features of size $\approx 2\pi/q$ radians. This leads to a natural cutoff at large q where the length scale becomes smaller than the feature size in a typical fingerprint, and noise starts to dominate. Similarly, there is a natural maximum for ω, namely where $2\pi/\omega$ matches $\min_{ab:a\neq b} \ln R_{ab}$. Finally we note that $L_x(-q,-\omega) = L_x^*(q,\omega)$ and $L_{x\theta}(-q,-w) = (-1)^q L_{x\theta}^*(q,\omega)$. This means that the grid point $(-q,-\omega)$ contains exactly the same information as (q,ω) and hence can be omitted. The considerations listed above are the only theoretical guidelines for choosing the grid; the best choice must be found by trial and error.

[2] Invariant under rotations φ that are an integer multiple of 2π.

The considerations for $M_{\mathbf{x}}, M_{\mathbf{x}\theta}$ are similar. The grid is a (q, R)-grid. The maximum q should be roughly the same as for the L-functions. The natural cutoffs for R are given by $\min_{ab:a\neq b} R_{ab}$ and $\max_{ab} R_{ab}$. It holds that $M_{\mathbf{x}}(-q, R) = M^*_{\mathbf{x}}(q, R)$ and $M_{\mathbf{x}\theta}(-q, R) = (-1)^q M^*_{\mathbf{x}\theta}(q, R)$. Hence it suffices to look at positive q only.

3.4.3 – Introducing weights. In the computation of a spectral function at enrollment, it is possible to introduce a weight factor for each of the (a, b)-pairs in the summation. It is advantageous to set a low weight for minutia pairs which are unlikely to be recovered later. A low recovery likelihood may occur e.g. when a minutia has low quality. Another reason can be a very large value of R_{ab}, in which case the recovery is sensitive to noise at the edge of the image, or a very small R_{ab} which may cause later minutia misidentification in case of noise. In our experiments we have not used weights other than 0 or 1.

3.4.4 – Choosing the score function. Let F denote one of the four spectral functions $L_{\mathbf{x}}, L_{\mathbf{x}\theta}$, $M_{\mathbf{x}}$, $M_{\mathbf{x}\theta}$ obtained at enrollment, and F' the noisy version of F obtained later, in the verification phase. We need a metric or 'score' function which quantifies how close F' is to F. As F and F' are complex-valued, there are quite some options. We have experimented with correlation functions for the radial and phase part of the complex numbers, as well as the real and imaginary part. Furthermore we have tried distance in the complex plane, with and without normalization of the function F as a whole. In our experiments it turns out that a complex correlation-like quantity is best able to discriminate between genuine fingerprint matches and impostors. We define our score S as

$$S(F, F') = |\rho(F, F')| \tag{3.8}$$

where ρ stands for the correlation as defined in Section 3.2.1, and the matrices F, F' are treated as vectors.

3.4.5 – Fusion of scores. The spectral functions $L_{\mathbf{x}}$ and $L_{\mathbf{x}\theta}$ together contain more information about the fingerprint than each one separately. The information is partially overlapping. We construct a 'fused' score by adding the two scores (3.8) in the same way as [105]: $S(L_{\mathbf{x}}, L'_{\mathbf{x}}) + S(L_{\mathbf{x}\theta}, L'_{\mathbf{x}\theta})$. Analogously, for the M-functions we work with the fused score $S(M_{\mathbf{x}}, M'_{\mathbf{x}}) + S(M_{\mathbf{x}\theta}, M'_{\mathbf{x}\theta})$.

3.5 — Experimental results

We have applied our minutia-pair approach to the VeriFinger database and the MCYT database [65]. The Verifinger database contains fingerprints from six individual persons, ten fingers per individual, eight images per finger. The size of each image is 326×357 pixels. The MCYT database contains fingerprints from 100 individuals, 10 fingers per individual, 12 images per finger (256×400 pixels). The fingerprints are generally of higher quality than in the Verifinger database.

We extracted minutia coordinates and orientations from the images by using the VeriFinger software [3].

3.5.1 – Optimal parameter choices. Good results were obtained with the following parameter settings. For the L-functions, $|q| \in \{1, \ldots, 24\}$ and $\omega \in [0.2, 37.7]$ with 32 equally spaced values. For the M-functions, $q \in \{1, \ldots, 16\}$; $R \in [16, 130]$ with 20 equally spaced points (MCYT database); $R \in [16, 160]$ with 25 equally spaced points (VeriFinger database). For the L_x and M_x functions we take only even q, as explained in Lemma 3.1. We set $\sigma = 2.3$ pixels.

A minutia extracted by VeriFinger is labeled with a quality $Q \in [0, 100]$. We took only minutiae with $Q \geqslant 45$. Furthermore we used an additional selection rule that turns out to improve overall results a bit: a minutia pair is discarded from the $\sum_{ab}$ summation in (3.2,3.3,3.6,3.7) if $2R_{ab}$ exceeds the horizontal size of the image.

In Fig. 3.2 we show an example of the M_x and $M_{x\theta}$ spectral function. Entirely different fingers obviously produce very different results. The two leftmost columns correspond to the same finger. Noisy images of the same finger do not produce results that, to the human eye, are clearly correlated. However, it turns out (Section 3.5.2) that the similarities are enough to distinguish between the enrolled user from an impostor.

3.5.2 – ROC curves and Equal Error Rates. We work in a *verification* setting, i.e. a stated identity has to be verified. We determine the False Rejection Rate (FRR) by comparing, for each finger in the database, all the pairs of images. We determine the False Acceptance Rate (FAR) by looking at each pair of different fingers, where one image is drawn at random for each finger (independently per pair).[3] We draw Receiver Operating Characteristic (ROC) curves as FAR plotted against FRR. Each point in the ROC curve corresponds to one threshold setting. The Equal Error Rate (EER) is the error rate in the point where FRR equals FAR. We build ROC curves based on only fingeprints from the available databases and not analyzing possible attacks.

Table 3.1 lists the EER values that we obtained. The ROC curves are shown in Fig. 3.3. We see that the M-functions consistently outperform the L-functions, and that the $L_{x\theta}, M_{x\theta}$ spectral functions outperform the location based functions. Furthermore we see that fusion of M_x and $M_{x\theta}$ yields only a modest improvement over $M_{x\theta}$. We conclude that, in our pair-based approach, the best option is to work either with $M_{x\theta}$ or the fusion of M_x and $M_{x\theta}$.

We benchmark our system against results reported by Xu et al. [105], which are based on ten individuals in the MCYT database who have high-quality fingerprint images. The ROC curves are shown in Fig. 3.4, and Table 3.2 contains

[3] This includes pairs of unlike fingers, e.g. thumb vs index finger. The statistics do not change much when only pairs of like fingers are compared.

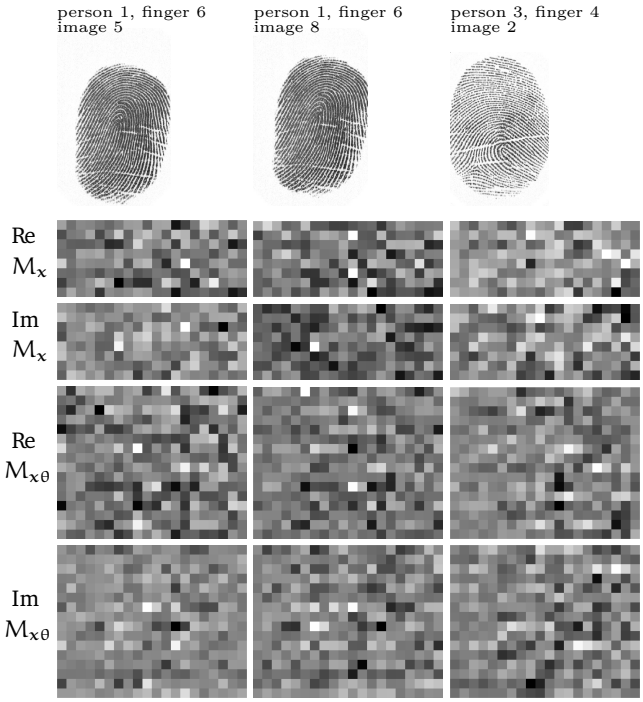


Figure 3.2: *Example of the spectral functions* M_x *and* $M_{x\theta}$*. MCYT database. The vertical axis is the* q*-axis, with* q *increasing upward. In each image, black represents the most negative value on the grid, and white the most positive.*

the EER comparison.[4] We conclude that our pair-based spectral function $M_{x\theta}$ has a discrimination performance comparable to Xu et al.'s spectral function.

3.5.3 – Rotation of the verification image. The results of Section 3.5.2 were obtained without Xu et al.'s procedure of trying out several image rotations so as to optimise the matching score. Now we discuss what happens when we do try a number of different rotation angles φ.

First we checked for the MCYT and the VeriFinger database how a rotation $\varphi \in (-10°, +10°)$ affects the M_x and $M_{x\theta}$-based score in case of a genuine image pair. At some optimal angle φ_0 the score is maximal. For all genuine pairs we determined φ_0, for M_x and $M_{x\theta}$. The histograms of φ_0 are shown in Fig. 3.5. We see that typically $|\varphi_0| < 6°$.

In Fig. 3.6 we present ROC curves that show the impact of trying multiple rotation angles φ in a limited range; we set the range based on Fig. 3.5. In the

[4] Unfortunately, [105] does not mention which ten individuals were selected.

Table 3.1: *Equal Error Rates obtained with the parameter settings given in Section 3.5.1. The notation 'F' stands for either L or M. No rotation of the verification image.*

Database	Function F	F_x	$F_{x\theta}$	Fusion
MCYT	L	5.3%	3.5%	3.0%
	M	4.0%	2.5%	2.2%
Verifinger	L	11%	4.9%	5.7%
	M	8.0%	3.3%	3.2%

Table 3.2: *Equal Error Rates for a subset of ten individuals in the MCYT database who have high-quality fingerprints. No rotation of the verification image. The last row is from Table VI in [105]. The L and M function were computed for individuals 16,24,26,32,34,35,46,53,80,94.*

Function F	F_x	$F_{x\theta}$	Fusion
L	1.1%	0.73%	0.31%
M	0.65%	0.35%	0.15%
Xu et al	0.47%	0.42%	0.22%

case of the MCYT database we see a consistent though small improvement. For the VeriFinger database the change is not always favourable; the ROC curves intersect. For both databases, the effect on the EER is minimal.

Increasing the range of φ does not improve the matching of genuine pairs; it does however increase the FAR. Hence the ROC curves become worse when we increase the range of φ.

These results allow for a very interesting trade-off: instead of opting for a minimal improvement of matching accuracy, we can skip the φ-search and thus significantly reduce the computation time. Note that Xu et al.'s method has a φ-search with 11 different values of φ.

3.6 — Computational efficiency

In this analysis we do not use the potential speedup that can be gained by skipping the φ-search.

Speed is important predominantly in the verification phase. From a freshly captured image the spectral function has to be computed on a number of grid points which we denote as N_{gr}. The spectral function has to be computed not once but several times, because N_φ different image orientations have to be

tried. Fortunately this does not multiply the total effort[5] by a factor N_φ, as the spectral function has a simple transform under rotation. (This holds for Xu et al as well as our L and M functions.)

Let Z be the number of minutiae. Let us denote the cost of computing one summation term of the spectral function in one grid point as T_s, and the cost of applying a rotation transform in one grid point as T_{rot}. The cost of computing the score can be written as $c \cdot N_{gr}$ where c is some small constant. The superscript 'G' will refer to Xu et al's spectral function; the superscript 'M' to our function M. The total cost for the verification phase (not counting the secure sketch) is

$$\text{Xu et al:} \quad N_{gr}^G Z T_s^G + (N_\varphi - 1) N_{gr}^G T_{rot}^G + N_\varphi c N_{gr}^G$$

$$\text{pair-based:} \; N_{gr}^M \binom{Z}{2} T_s^M + (N_\varphi - 1) N_{gr}^M T_{rot}^M + N_\varphi c N_{gr}^M .$$

We have $T_s^G \approx T_s^M$, $T_{rot}^G \approx T_{rot}^M$, $T_{rot} < T_s$. The main difference between the two approaches lies in the first term: $N_{gr}^G Z$ versus $N_{gr}^M \binom{Z}{2}$, i.e. N_{gr}^G versus $\frac{1}{2} N_{gr}^M (Z-1)$. Xu et al report a 128×256 grid, yielding $N_{gr}^G = 32768$. In contrast, our $M_{x\theta}$-function is evaluated on a grid of size $N_{gr}^M \leqslant 16 \cdot 25 = 400$. Given that typically $Z \approx 35$, we have $\frac{1}{2} N_{gr}^M (Z-1) \approx 6800$. Hence our verification is faster than [102, 103, 105].

Note that [104] introduces a reduced template size by applying Principal Component Analysis or a Discrete Fourier Transform to select informative features. This selection reduces the template size by roughly a factor 10. However, these methods still require computation of the spectral function on many grid points.

3.7 — Discussion

Achieving translation invariance by looking at *minutia pairs* seems to be advantageous compared to taking the absolute value of a Fourier transform. The minutia-pair approach is able to extract information from a fingerprint using fewer grid points. We conjecture that this is due to the fact that our spectral functions retain phase information instead of discarding it. Of the four functions that we studied, the $M_{x\theta}$ performs best. Fusion of the matching scores from M_x and $M_{x\theta}$ leads to an EER comparable to Xu et al.

Due to the reduction of the number of grid points our method is faster than the verification described by Xu et al., in spite of the increased number of summation terms. As an unexpected bonus, it turns out that we can omit the search for an optimal rotation angle; this gives an additional speed improvement.

[5] Here we look only at the computation of the spectral function and the score; not at the cost of N_φ Secure Sketch reconstruction attempts.

As topics for future work we mention (i) further speedup by discarding grid points that have a bad signal-to-noise ratio; (ii) applying Principal Component Analysis and similar techniques to improve the EER; (iii) constructing a HDS based on $\mathsf{M}_{\mathbf{x}}$ and $\mathsf{M}_{\mathbf{x}\theta}$.

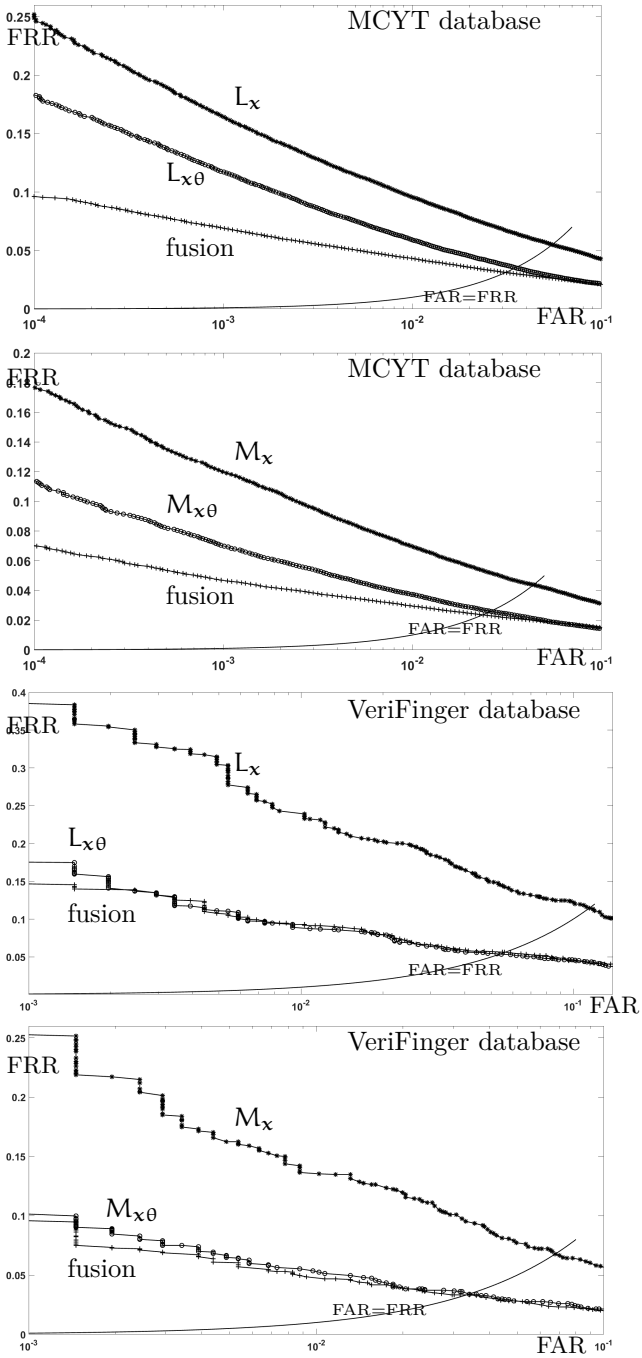


Figure 3.3: *ROC curves for our pair-based spectral functions applied to two databases. No rotation of the verification image.*

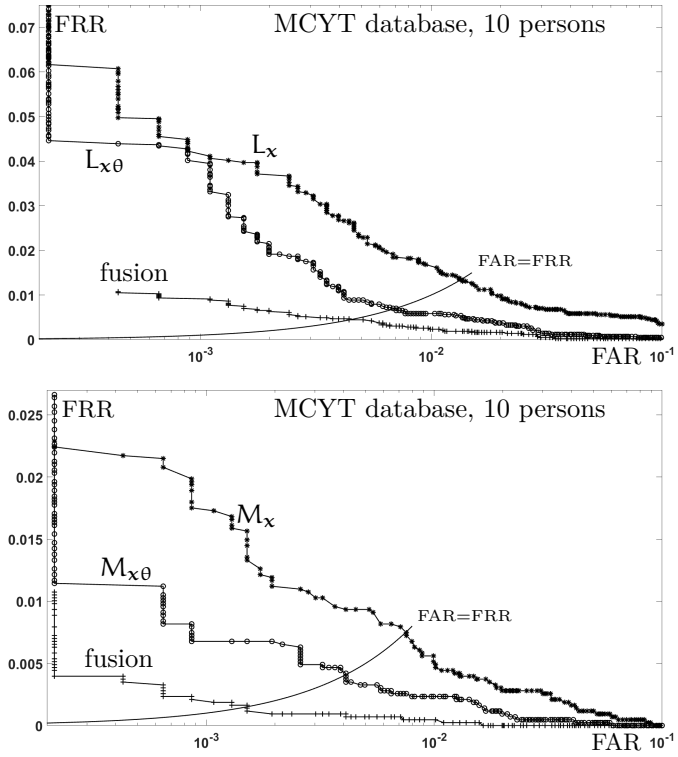


Figure 3.4: *ROC curves for the ten-person subset of the MCYT database. No rotation of the verification image.*

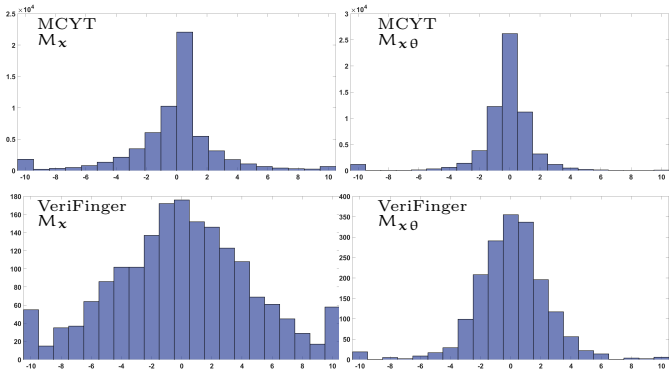


Figure 3.5: *Histograms of the optimal rotation angle* φ_0 *(degrees).*

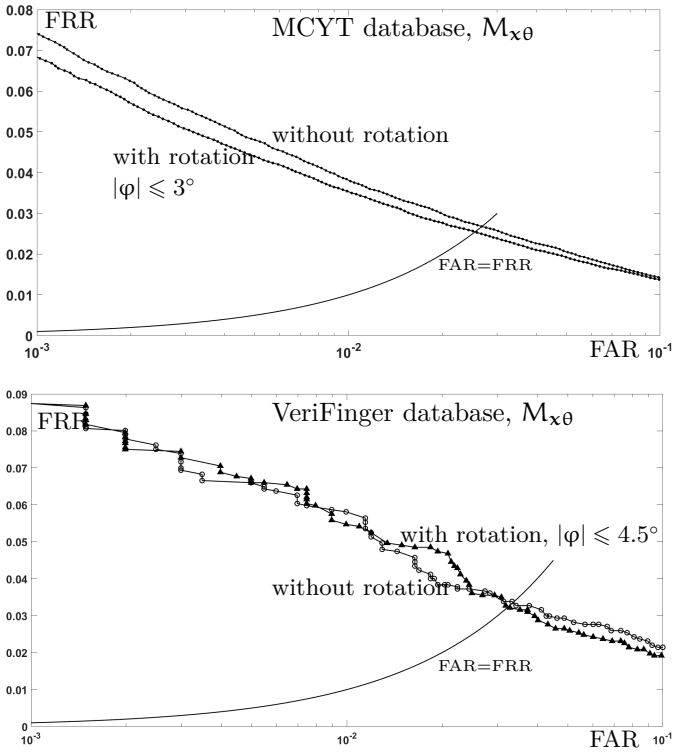


Figure 3.6: *ROC curves with and without rotation of the verification image.*

Chapter 4

Fingerprint template protection using minutia-pair spectral representations

In Chapter 2 and Chapter 3 we developed a technique to transform fingerprint images to a fixed-length representation and then quantize (to binary), as the foundation to construct a privacy-preserving scheme for fingerprints. Next we ask can we construct a high-performance (in terms of matching accuracy and speed) helper data system based on fingerprint minutiae? (Research Question 2). First we extend the minutia-pair spectral approach by including the second invariant angle that is present in a minutiae pair. At every step of the scheme we do an analysis of the performance of the two-stage HDS for various choices of spectral function parameters and number of images used at enrollment. We notice that, for high-quality fingerprints, the transition from an unprotected spectral function to fully protected enrollment data can be done with almost no performance penalty.

4.1 — Introduction

4.1.1 – Biometric template protection. Biometric authentication has become popular because of its convenience. Biometrics cannot be forgotten or left at home. Although biometric data is not exactly secret (we are leaving a trail of fingerprints, DNA etc.), it is important to protect biometric data for privacy reasons. Unprotected storage of biometric data could reveal medical conditions and would allow cross-matching of entries in different databases. Large-scale availability of unprotected biometric data would make it easier for malevolent parties to leave misleading traces at crime scenes (e.g. artificial

fingerprints [61], synthesized DNA [28].) One of the easiest ways to properly protect a biometric database against breaches and insider attacks (scenarios where the attacker has access to decryption keys) is to store biometrics in hashed form, just like passwords. An error-correction step has to be added to get rid of the measurement noise. To prevent critical leakage from the error correction redundancy data, one uses a Helper Data System (HDS) [19, 54, 81], for instance a Fuzzy Extractor or Secure Sketch [13, 22, 47].

We consider the HDS approach to be the preferred method for privacy-preserving storage of biometric enrolment data, because of its strong privacy guarantees combined with low computational requirements. (Approaches based on homomorphic encryption have excellent privacy but are computationally expensive; approaches based on random projections are cheap but their security is difficult to ascertain.) The best known and simplest HDS scheme is the code-offset method (COM). The COM utilizes a linear binary error-correction code and thus requires a fixed-length representation of the biometric measurement. Such a representation is not straightforward when the measurement noise can cause features of the biometric to appear/disappear. For instance, some minutiae may not be detected in every image captured from the same finger.

Bringer et al. [11] proposed a fixed-length representation based on minutia vicinities. However, it is vulnerable to minutia misdetections. Topcu et al. [87] introduced a system containing a machine learning step; it is difficult to convert to a privacy-preserving scheme. Jin et al. [43] proposed minutiae processing using kernel methods to arrive at a fixed-length representation. However, they do not explain how to deal, privacy-wise, with the personalised reliable components and personalised training data which are required for the reconstruction step. Tuyls et al. [88] used Gabor filters to derive a fixed-length representation, and applied a HDS. However, their scheme does not have translation invariance, and the helper data is personalized.

A fixed-length representation called *spectral minutiae* was introduced by Xu et al. [102–105]. For every detected minutia of sufficient quality, the method evaluates a Fourier-like spectral function on a fixed-size two-dimensional grid; the contributions from the different minutiae are added up. Disappearance of minutiae or appearance of new ones does not affect the size of the grid. Topcu et al. [86] applied biohashing to spectral minutiae as a form of template protection. Shao and Veldhuis [79] applied a HDS to spectral minutiae.

One of the drawbacks of Xu et al.'s construction is that phase information is discarded in order to obtain translation invariance. Nandakumar [63] proposed a variant which does not discard the phase information. However, it reveals personalised reliability data, which makes it difficult to use in a privacy-preserving scheme.

A minutia-*pair* based variant of Xu et al.'s technique was introduced in [82]. It has a more compact grid and reduced computation times. Minutia pairs (and even triplets) were used in [26, 44], but in the context of a different attacker

model which allows encryption keys to exist that are not accessible to the adversary.

4.1.2 – Contributions and outline.

- We extend the pair-based spectral minutiae method [82] by introducing a new spectral function that captures different information from the minutia orientations. A minutia pair contains two invariant angles, namely the two orientations relative to the connecting line. In [82] only one of these was exploited.
- Then we use the spectral functions as the basis for a two-stage template protection system consisting of two helper data systems, along the lines of [19]. The first stage discretises the analog spectral representation using a zero-leakage HDS [19, 81]. This first HDS reduces quantization noise, and the helper data reveals no information about the quantized data. Discretisation of the spectral functions typically yields only one bit per grid point. We concatenate the discrete data from all the individual grid points into one long bitstring. In the second stage we apply the Code Offset Method. Our code of choice is a Polar code, because Polar codes are low-complexity capacity-achieving codes with flexible rate.
- We present False Accept Rate (FAR) vs. False Reject Rate (FRR) tradeoffs at various stages of the data processing. We introduce the 'superfinger' enrollment method, in which we average the spectral functions from multiple enrollment images. By combining three images in this way, and constructing a Polar code specifically tuned to the individual bit error rate of each bit position, we achieve an Equal Error Rate (EER) around 1% for a high-quality fingerprint database, and around 6% for a low-quality database. Our HDS achieves these numbers while matching extracted strings that are short, 25 bits or less. The entropy of such a string is less than the string length because of mutual dependencies between the bits. This in contrast to the much larger numbers mentioned in other works.
- Our results show that, once we have switched to the spectral representation, privacy protection causes little performance degradation in terms of FAR, FRR. However, the transition from a list of minutiae to the spectral representation reduces performance.

In a sense we have a 'negative' result. Our EER is worse than for matching schemes without privacy protection, or schemes that use homomorphic encryption. (On the other hand, by combining multiple fingers the EER can be lowered to an acceptable level.) The main contribution of this paper is, however, that we push the minutia-pair spectral function approach to its limits while at the same time giving the ZLHDS technique a baptism of fire in a real-life biometrics problem. We find that (i) we cannot really recommend spectral functions as a good fixed-length representation, although there is no better alternative; (ii) the ZLHDS performs splendidly; (iii) we can confirm that Polar codes are well

suited for use in a HDS, even under tougher circumstances than in previous work [16].
In Section 4.3 we introduce notation and briefly review helper data systems, the spectral minutiae representation, and polar codes. In Section 4.4 we introduce the new spectral function. In Section 4.5 we explain our experimental approach and motivate certain design choices such as the number of discretisation intervals and the use of a Gaussian approximation. We introduce two methods for averaging enrollment images. Section 4.6 contains our results, mostly in the form of ROC curves. In Sections 4.7 and 4.8 we discuss the results and identify topics for future work.

4.2 — Methods

The aim of this study is to develop improved techniques for privacy-preserving storage of biometric data. We use fingerprint data from publicly available databases and analyze the performance of our template protection scheme using standard nonproprietary techniques. We compare primarily against the existing spectral minutiae technique of Xu et al.

4.3 — Preliminaries

4.3.1 – Notation and terminology. We use capitals to represent random variables, and lowercase for their realizations. Sets are denoted by calligraphic font. The set $\mathcal{S}$ is defined as $\mathcal{S} = \{0, \ldots, N-1\}$. The mutual information (see e.g. [18]) between X and Y is $I(X;Y)$. The probability density function (pdf) of the random variable $X \in \mathbb{R}$ in written as $f(x)$ and its cumulative distribution function (cdf) as $F(x)$. We denote the number of minutiae found in a fingerprint by Z. The coordinates of the j'th minutia are $\mathbf{x}_j = (x_j, y_j)$ and its orientation is θ_j. We write $\mathbf{x} = (\mathbf{x}_j)_{j=1}^{Z}$ and $\boldsymbol{\theta} = (\theta_j)_{j=1}^{Z}$ We will use the abbreviations FRR = False Reject Rate, FAR = False Accept Rate, EER = Equal Error Rate, ROC = Receiver Operating Characteristic. Bitwise xor of binary strings is denoted as $\oplus$.

4.3.2 – Helper Data Systems. A HDS is a cryptographic primitive that allows one to reproducibly extract a secret from a noisy measurement. A HDS consist of two algorithms: `Gen` (generation) and `Rep` (reconstruction), see Fig. 1.1. The `Gen` algorithm takes a measurement X as input and generates the secret S and a helper data W. The `Rep` algorithm has as input a noisy measurement Y and the helper data; it outputs an estimator $\hat{S}$. If Y is sufficiently close to X then $\hat{S} = S$. The helper data should not reveal much about S. Ideally it holds that $I(W;S) = 0$. This is known as *Zero Leakage* helper data.

4.3.3 – Two-stage HDS template protection scheme. Fig. 4.1 shows the two-stage HDS architecture as described e.g. in [19]. The enrollment measurement x is transformed to the spectral representation $(x_i)_{i=1}^{M}$ on M grid points. The

first-stage enrollment procedure `Gen1` is applied to each x_i individually, yielding short (mostly one-bit) secrets s_i and zero-leakage helper data w_i. The $s_1 \dots s_M$ are concatenated into a string k. Residual noise in k is dealt with by the second-stage HDS (Code Offset Method), whose `Gen2` produces a secret c and helper data r. A hash $h(c||z)$ is computed, where z is salt. The hash and the salt are stored.
In the verification phase, the noisy y is processed as shown in the bottom half of Fig. 4.1. The reconstructed secret $\hat{c}$ is hashed with the salt z; the resulting hash is compared to the stored hash.

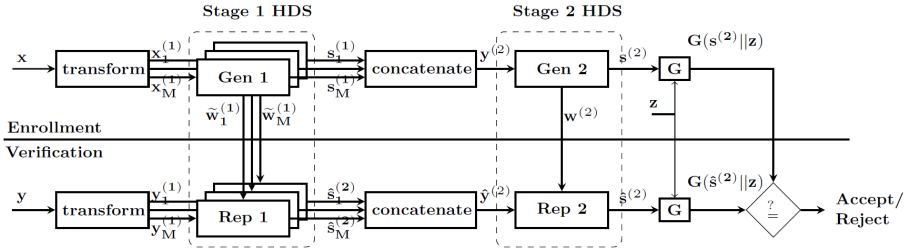


Figure 4.1: *Two-stage Helper Data System. Figure adapted from [19].*

4.3.4 – Minutia-pair spectral representation. Minutiae are features in a fingerprint, e.g. ridge endings and bifurcations. We briefly describe the minutia-pair spectral representation introduced in [82]. For minutia indices $a, b \in \{1, \dots, Z\}$ the distance and angle are given by $R_{ab} = |\mathbf{x}_a - \mathbf{x}_b|$ and $\tan \phi_{ab} = \frac{y_a - y_b}{x_a - x_b}$. The spectral function $\mathcal{M}_S$ is defined as

$$\mathcal{M}_{x\theta}(q, R) = \sum_{\substack{a,b \in \{1,\dots,Z\} \\ a<b}} e^{iq\phi_{ab}} e^{-\frac{(R-R_{ab})^2}{2\sigma^2}} e^{i(\theta_b - \theta_a)}, \tag{4.1}$$

where σ is a width parameter. The spectral function is evaluated on a discrete (q, R) grid. A pair (q, R) is referred to as a grid point. The variable q is integer and can be interpreted as the Fourier conjugate of an angular variable, i.e. a harmonic. The function $\mathcal{M}_S$ is invariant under translations of $\mathbf{x}$. When a rotation of the whole fingerprint image is applied over an angle δ, the spectral function transforms in a simple way,

$$\mathcal{M}_{x\theta}(q, R) \to e^{iq\delta} \mathcal{M}_{x\theta}(q, R). \tag{4.2}$$

4.3.5 – Zero Leakage Helper Data Systems. We briefly review the ZLHDS developed in [19, 81] for quantization of an enrollment measurement $X \in \mathbb{R}$. The

density function of X is f, and the cumulative distribution function is F. The verification measurement is Y. The X and Y are considered to be noisy versions of an underlying 'true' value. They have zero mean and variance σ_X^2, σ_Y^2, respectively. The correlation between X and Y can be characterized by writing $Y = \lambda X + V$, where $\lambda \in [0,1]$ is the attenuation parameter and V is zero-mean noise independent of X, with variance σ_V^2. It holds that $\sigma_Y^2 = \lambda^2\sigma_X^2 + \sigma_V^2$. We consider the *identical conditions* case: the amount of noise is the same during enrollment and reconstruction. In this situation we have $\sigma_X^2 = \sigma_Y^2$ and $\lambda^2 = 1 - \frac{\sigma_V^2}{\sigma_X^2}$.

The real axis $\mathbb{R}$ is divided into N intervals $\mathcal{A}_\alpha = (\Omega_\alpha, \Omega_{\alpha+1})$, with $\alpha \in \mathcal{S}$, $\mathcal{S} = \{0, \ldots, N-1\}$. Let $p_\alpha = \Pr[X \in \mathcal{A}_\alpha]$. The quantization boundaries are given by $\Omega_\alpha = F^{\text{inv}}(\sum_{j=0}^{\alpha-1} p_j)$. The `Gen` algorithm produces the secret s as $s = \max\{\alpha \in \mathcal{S} : x \geqslant \Omega_\alpha\}$ and the helper data $\widetilde{w} \in [0,1)$ as $\widetilde{w} = [F(x) - \sum_{j=0}^{s-1} p_j]/p_s$. The inverse relation, for computing x as a function of s and $\widetilde{w}$, is given by $\xi_{s,\widetilde{w}} = F^{\text{inv}}(\sum_{j=0}^{s-1} p_j + \widetilde{w}p_s)$.

The `Rep` algorithm computes the estimator $\hat{s}$ as the value in $\mathcal{S}$ for which it holds that $y \in (\tau_{\hat{s},\widetilde{w}}, \tau_{\hat{s}+1,\widetilde{w}})$, where the parameters τ are decision boundaries. In the case of Gaussian noise these boundaries are given by

$$\tau_{\alpha,\widetilde{w}} = \lambda\frac{\xi_{\alpha-1,\widetilde{w}} + \xi_{\alpha,\widetilde{w}}}{2} + \frac{\sigma_V^2 \ln \frac{p_{\alpha-1}}{p_\alpha}}{\lambda(\xi_{\alpha,\widetilde{w}} - \xi_{\alpha-1,\widetilde{w}})}. \tag{4.3}$$

Here it is understood that $\xi_{-1,\widetilde{w}} = -\infty$ and $\xi_{N,\widetilde{w}} = \infty$, resulting in $\tau_{0,\widetilde{w}} = -\infty$, $\tau_{N,\widetilde{w}} = \infty$.

The above scheme ensures that $I(\widetilde{W}; S) = 0$ and that the reconstruction errors are minimized.

4.3.6 – The Code Offset Method (COM). We briefly describe how the COM is used as a Secure Sketch. Let C be a linear binary error correcting code with message space $\{0,1\}^m$ and codewords in $\{0,1\}^n$. It has an encoding `Enc`: $\{0,1\}^m \to \{0,1\}^n$, a syndrome function `Syn`: $\{0,1\}^n \to \{0,1\}^{n-m}$ and a syndrome decoder `SynDec`: $\{0,1\}^{n-m} \to \{0,1\}^n$. In Fig. 4.1 the `Gen2` computes the helper data $\mathbf{W}$ as $\mathbf{W} = \texttt{Syn}\,\mathbf{Y}^{(2)}$. The $\mathbf{S}$ in Fig. 4.1 is equal to $\mathbf{Y}^{(2)}$. The `Rep2` computes the reconstruction $\hat{\mathbf{S}} = \hat{\mathbf{Y}}^{(2)} \oplus \texttt{SynDec}(\mathbf{W} \oplus \texttt{Syn}\,\hat{\mathbf{Y}}^{(2)})$.

4.3.7 – Polar codes. Polar codes, proposed by Arıkan [5], are a class of linear block codes that get close to the Shannon limit even at small code length. They are based on the repeated application of the *polarization* operation $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ on two bits of channel input. Applying this operation creates two virtual channels, one of which is better than the original channel and one worse. For n channel inputs, repeating this procedure in the end yields m near-perfect virtual channels, with m/n close to capacity, and $n - m$ near-useless channels. The

m-bit message is sent over the good channels, while the bad ones are 'frozen', i.e used to send a fixed string known a priori by the recipient.

Polar codes have a number of advantages, such as flexible code rate and excellently performing soft-decision decoders. The most popular decoder is the Successive Cancellation Decoder (SCD), which sequentially estimates message bits $(S_i)_{i=1}^m$ according to the frozen bits and the previously estimated bits $\hat{S}_{i-1}$. Polar codes have been recently adopted for the next generation wireless standard (5G), especially for control channels, which have short block length ($\leqslant 1024$). Because of these advantages we have chosen Polar codes for implementing the error correction step in our HDS scheme (see Section 4.6).

4.4 — A new spectral function

Consider Fig. 4.2 (modified from [101]). The invariant angle β_a is defined as the angle from the orientation of minutia a to the connecting line ab, taken in the positive direction. (The β_b is defined analogously). Modulo 2π it holds that $\theta_a + \beta_a = \phi_{ab}$ and $\theta_b + \beta_b = \phi_{ab} + \pi$. The spectral function (4.1) uses only the invariant angle $\beta_a - \beta_b + \pi = \theta_b - \theta_a$. The second invariant angle, which can be written e.g. as $\pi - \beta_a - \beta_b = \theta_a + \theta_b - 2\phi_{ab}$, is not used in [82]. We therefore now introduce a new spectral function, denoted as $\mathcal{M}_{x\beta}$, which incorporates the invariant angle $\pi - \beta_a - \beta_b$.

$$\mathcal{M}_{x\beta}(q, R) = \sum_{\substack{a,b \in \{1,\dots,Z\} \\ a<b}} e^{i\phi_{ab}(q-2)} e^{-\frac{(R-R_{ab})^2}{2\sigma^2}} e^{i(\theta_b+\theta_a)}. \tag{4.4}$$

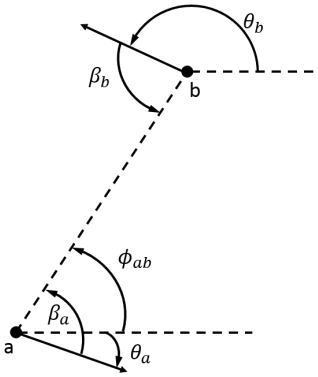


Figure 4.2: *The relevant angles in a minutia pair. The β_a and β_b are rotation invariant. The ϕ_{ab} is sensitive to image rotation.*

Under image rotation over an angle δ this function transforms as

$$\mathcal{M}_{x\beta}(q, R) \to e^{iq\delta}\mathcal{M}_{x\beta}(q, R). \tag{4.5}$$

We will use $\mathcal{M}_{x\theta}$, $\mathcal{M}_{x\beta}$ and their fusion.

4.5 — Experimental approach

4.5.1 – Databases. We use the MCYT, FVC2000(DB2), and FVC2002(DB2) database. From this point on we omit the 'DB2' designation for brevity. The MCYT database [65] contains good-quality images from 100 individuals: 10 fingers per individual and 12 images per finger. FVC2000 and FVC2002 contain low-quality images (only index and middle fingers [60]). Each FVC database contains 100 fingers, 8 images per finger. In FVC2002, images number 3, 4, 5, and 6 have an exceptionally large angular displacement, so they are omitted from the experiments.

We extract the minutia position and orientation (x_j, y_j, θ_j) by using VeriFinger software [3]. For MCYT we evaluate the spectral functions on the same grid as [82], namely $R \in \{16, 22, 28, \ldots, 130\}$ and $q \in \{1, 2, \ldots, 16\}$ and we maintain $\sigma = 2.3$ pixels. For the FVC databases we use the same grid, and $\sigma = 3.2$ pixels turns out to be a good choice. The average number of minutiae that can be reliably found is $Z = 35$.

4.5.2 – No image rotation. As mentioned in [82], during the reconstruction procedure one can try different rotations of the verification image, but it results only in a minor improvement of the EER. For this reason we do not apply image rotation.

4.5.3 – Quantization methods. Before quantization all spectral functions are normalized to zero mean and unit variance, where the variance is taken of the real and imaginary part together. We quantize the real and imaginary part of the spectral functions separately. We study two methods: 'hard thresholding' (without helper data) and the Zero Leakage quantization of Section 4.3.2. The hard thresholding gives a bit value '1' if $\mathrm{Re}\, M > 0$ and '0' otherwise. We will show results for this method mainly to demonstrate the advantages of Zero Leakage quantization.

4.5.4 – Gaussian probability distributions. When using the ZLHDS formulas we will assume that the spectral functions are Gaussian-distributed. Figs. 4.3 and 4.4 illustrate that this assumption is not far away from the truth.[1]

4.5.5 – Zero leakage quantization.

[1] Note that we often see correlations between the real and imaginary part. This has no influence on the ZLHDS.

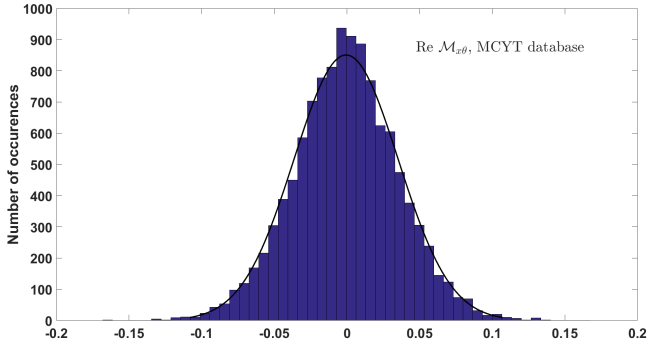


Figure 4.3: *Histogram of* $\mathrm{Re}\,\mathcal{M}_{x\theta}$*, and a fitted Gaussian.*

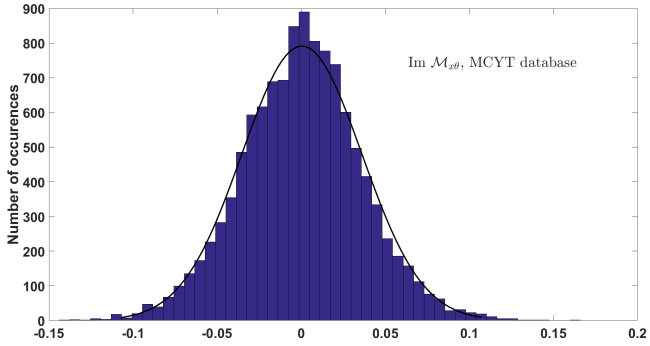


Figure 4.4: *Histogram of* $\mathrm{Im}\,\mathcal{M}_{x\theta}$*, and a fitted Gaussian.*

Signal to noise ratio; setting N. In the ZLHDS of Section 4.3.5, the optimal choice of the parameter N (number of quantization intervals) depends on the signal to noise ratio. Fig. 4.5 shows a comparison between $N = 2$ and $N = 3$. At low noise it is obvious that $N = 3$ extracts more information from the source than $N = 2$. At σ_V/σ_X larger than approximately 0.3, there is a regime where $N = 3$ can extract more in theory, but is hindered in practice by the high bit error rate. At $\sigma_V/\sigma_X > 0.55$ the $N = 2$ 'wins' in all respects.

For our data set, we define a $\sigma_X^2(q, R)$ for every grid point (q, R) as the variance of $\mathcal{M}(q, R)$ over all images in the database. The noise $\sigma_V^2(q, R)$ is the variance over all available images of the same finger, averaged over all fingers.

Figs. 4.6 and 4.7 show the noise-to-signal ratio σ_V/σ_X. Note the large amount of noise; even the best grid points have $\sigma_V/\sigma_X > 0.45$. Fig. 4.5 tells us that setting $N = 2$ is the best option, and this is the choice we make. At $N = 2$ we extract two bits per grid point from each spectral function (one from $\mathrm{Re}\,\mathcal{M}$, one

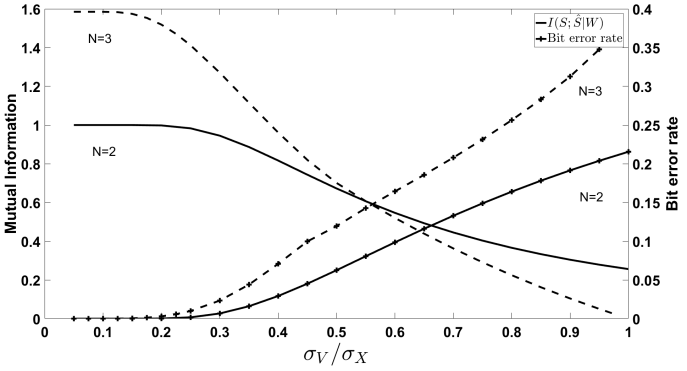


Figure 4.5: *Comparison of ZLHDS with* $\mathsf{N} = 2$ *versus* $\mathsf{N} = 3$. *Lines without markers: Mutual information between the enrolled key* S *and the reconstructed key* $\hat{\mathsf{S}}$ *given helper data* W*, as a function of* $\sigma_{\mathsf{V}}/\sigma_{\mathsf{X}}$. *Markers: bit error rate as a function of* $\sigma_{\mathsf{V}}/\sigma_{\mathsf{X}}$. *The curves follow equations (22) and (26) from [82].*

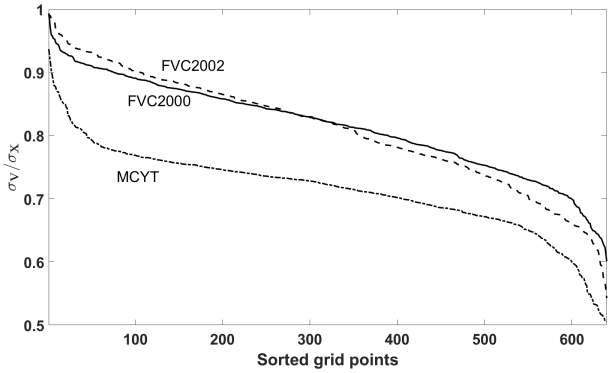


Figure 4.6: *Sorted noise-to-signal ratio of* $\mathcal{M}_{\mathsf{x}\theta}$ *for different databases.*

from $\mathsf{Im}\,\mathcal{M}$). Hence our bit string string k (see Fig. 4.1) derived from $\mathcal{M}_{\mathsf{x}\theta}$ has length 640. When we apply fusion of $\mathcal{M}_{\mathsf{x}\theta}$ and $\mathcal{M}_{\mathsf{x}\beta}$ this becomes 1280.
For $\mathsf{N} = 2$ the formulas in Section 4.3.5 simplify to $\mathcal{A}_0 = (-\infty, 0)$, $\mathcal{A}_1 = [0, \infty)$, $\mathsf{p}_0 = \mathsf{p}_1 = \frac{1}{2}$, $\xi_{0,\widetilde{w}} = \mathsf{F}^{\mathsf{inv}}(\frac{\widetilde{w}}{2})$, $\xi_{1,\widetilde{w}} = \mathsf{F}^{\mathsf{inv}}(\frac{1}{2} + \frac{\widetilde{w}}{2})$, $\tau_{1,\widetilde{w}} = \frac{\lambda}{2}(\xi_{0,\widetilde{w}} + \xi_{1,\widetilde{w}})$. Since we work with Gaussian distributions, F is the Gaussian cdf ('probability function').

Enrollment and reconstruction. We have experimented with three different enrollment methods:

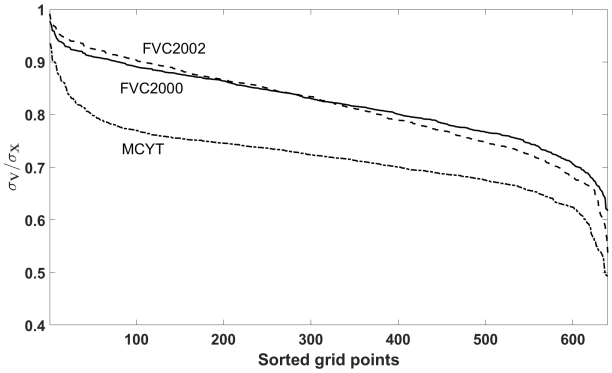


Figure 4.7: *Sorted noise-to-signal ratio of* $\mathcal{M}_{x\beta}$ *for different databases.*

E1. A single image is used.
E2: We take the first[2] t images of a finger and calculate the average spectral function. We call this the 'superfinger' method. In the ZLHDS calculations the signal-to-noise ratio of the average spectral function is used.
E3: For each of t images we calculate an enrollment string k. We apply bit-wise majority voting on these strings. (This requires odd t.) The reconstruction boundaries are calculated based on the superfinger method, i.e. as in E2.
Reconstruction:
We study fingerprint authentication with genuine pairs and impostor pairs. For pedagogical reasons we will present results at four stages of the signal processing: (1) spectral function domain, before quantization; (2) binarized domain, without applying the first-stage HDS; (3) binarized with first-stage ZLHDS; (4) with first-stage ZLHDS and discarding the highest-noise grid points.
In the spectral function domain the fingerprint matching is done via a correlation score [82]. In the binarized domain we look at the Hamming weight between the enrolled k and the reconstructed $\hat{k}$. For all cases we will show ROC curves in order to visualise the FAR-FRR tradeoff as a function of the decision threshold. Let the number of images per finger be denoted as M, and the number of fingers in a database as L .

E1: For the spectral domain and the quantization without HDS we compare all genuine pairs, i.e. $\binom{M}{2}$ image pairs per finger, resulting in $L\binom{M}{2}$ data points. For ZLHDS the number is twice as large, since there is an asymmetry between enrollment and reconstruction. For the FVC databases we generate all possible

[2]We take the *first* t images to show that the approach works. We are not trying to optimize the choice of images.

impostor combinations (all images of all impostor fingers), resulting in $\mathcal{O}(M^2L^2)$ data points.

For the MCYT database, which is larger, we take only *one random image* per impostor finger, resulting in $\mathcal{O}(ML^2)$ data points.

E2+E3: For genuine pairs we compare the superfinger to the remaining $M - t$ images. Thus we have $(M - t)L$ data points. Impostor pairs are generated as for E1.

Note: The VeriFinger software was not able to extract information for every image.

4.6 — Experimental results

4.6.1 – FAR/FRR rates before error correction. For each the data processing steps/options before application of the Code Offset method, we investigate the False Accept rates and False Reject rates. We identify a number of trends.

- Figs. 4.8 and 4.9 show ROC curves. All the non-analog curves were made under the implicit assumption that for each decision threshold (number of bit flips) an error-correcting code can be constructed that enforces that threshold, i.e. decoding succeeds only if the number of bit flips is below the threshold. Unsurprisingly, we see in the figures that quantization causes a performance penalty. Furthermore the penalty is clearly less severe when the ZLHDS is used. Finally, it is advantageous to discard some grid points that have bad signal-to-noise ratio σ_X/σ_V. For the curves labeled 'ZLHDS+reliable components' only the least noisy[3] 512 bits of k were kept (1024 in the case of fusion). Our choice for the number 512 is not entirely arbitrary: it fits error-correcting codes. Note in Fig. 4.9 that ZLHDS with reliable component selection performs better than analog spectral functions *without* reliable component selection. (But not better than analog with selection.) For completeness we mention that Verifinger's privacy-less matching based on minutiae (without spectral functions) has an EER of 0.58% for FVC2000 [1] and 0.52% for the FVC2002 database [2]. Clearly the transition to spectral functions causes a performance loss.
- The E2 and E3 enrollment methods perform better than E1. Furthermore, performance increases with t. A typical example is shown in Fig. 4.10.
- The spectral functions $\mathcal{M}_{x\theta}$ and $\mathcal{M}_{x\beta}$ individually have roughly the same performance. Fusion yields a noticeable improvement. An example is shown in Fig. 4.11. (We implemented fusion in the analog domain as addition of the two similarity scores.)

[3] This is defined as a global property of the whole database. Our selection of reliable components does *not* reveal anything about an individual and hence preserves privacy. Note that [63] does reveal personalised reliable components and obtains better FA and FN error rates.

- Tables 4.1 to 4.5 show Equal Error Rates and Bit Error Rates. We see that enrollment methods E2 and E3 have similar performance, with E2 yielding a somewhat lower genuine-pair BER than E3.
- In Table 4.1 it may look strange that the EER in the rightmost column is sometimes lower than in the 'analog' column. We think this happens because there is no reliable component selection in the 'analog' procedure.
- Ideally the impostor BER is 50%. In the tables we see that the impostor BER can get lower than 50% when the ZLHDS is used and the enrollment method is E2. On the other hand, it is always around 50% in the 'No HDS' case. This seems to contradict the Zero Leakage property of the helper data system. The ZLHDS is supposed not to leak, i.e. the helper data should not help impostors. However, the zero-leakage property is guaranteed to hold only if the variables are independent. In real-life data there are correlations between grid points and correlations between the real and imaginary part of a spectral function.

Table 4.1: *Equal Error Rates and Bit Error Rates. MCYT database. Enrollment methods E1 and E2. Numbers displayed as a percentage are EERs. Numbers without a % sign are BERs: the left number is for genuine pairs, right for impostors.*

#images (t)		Analog	No HDS	ZLHDS	ZLHDS+r.c.
1	$\mathcal{M}_{x\theta}$	2.6%	3.7% 0.33 0.50	3.4% 0.30 0.49	3.2% 0.29 0.49
	$\mathcal{M}_{x\beta}$	2.4%	3.7% 0.33 0.50	3.4% 0.31 0.50	3.2% 0.29 0.49
	Fusion	2.1%	2.9% 0.33 0.50	2.6% 0.30 0.49	2.3% 0.29 0.49
2	$\mathcal{M}_{x\theta}$	2.1%	3.2% 0.33 0.50	2.3% 0.28 0.46	2.1% 0.27 0.46
	$\mathcal{M}_{x\beta}$	1.7%	3.01% 0.33 0.50	2.4% 0.28 0.47	2.2% 0.27 0.47
	Fusion	1.6%	2.3% 0.33 0.50	1.7% 0.28 0.46	1.4% 0.27 0.47
3	$\mathcal{M}_{x\theta}$	1.4%	2.2% 0.31 0.50	1.3% 0.24 0.45	1.1% 0.23 0.46
	$\mathcal{M}_{x\beta}$	1.1%	2.0% 0.31 0.50	1.2% 0.25 0.46	1.1% 0.23 0.46
	Fusion	1.1%	1.5% 0.31 0.50	0.9% 0.24 0.46	0.7% 0.23 0.46
4	$\mathcal{M}_{x\theta}$	1.2%	1.7% 0.29 0.50	1.0% 0.22 0.45	0.9% 0.21 0.45
	$\mathcal{M}_{x\beta}$	1.0%	1.6% 0.30 0.50	0.9% 0.22 0.45	0.8% 0.21 0.45
	Fusion	0.9%	1.1% 0.30 0.50	0.6% 0.22 0.45	0.5% 0.21 0.45

Table 4.2: *EERs and BERs for the FVC2000 database. Enrollment methods E1 and E2.*

#images (t)		Analog	No HDS	ZLHDS	ZLHDS+r.c.
1	$\mathcal{M}_{x\theta}$	6.0%	9.4% 0.39 0.50	9.0% 0.37 0.50	8.0% 0.36 0.50
	$\mathcal{M}_{x\beta}$	6.1%	10.4% 0.39 0.50	9.5% 0.38 0.50	8.1% 0.37 0.50
	Fusion	4.8%	7.3% 0.39 0.50	6.5% 0.38 0.50	5.5% 0.36 0.50
2	$\mathcal{M}_{x\theta}$	4.5%	7.2% 0.37 0.50	5.7% 0.33 0.47	5.0% 0.32 0.47
	$\mathcal{M}_{x\beta}$	4.8%	7.9% 0.38 0.50	6.9% 0.34 0.47	5.6% 0.32 0.47
	Fusion	3.9%	5.1% 0.37 0.50	5.0% 0.33 0.47	4.1% 0.32 0.47
3	$\mathcal{M}_{x\theta}$	3.0%	5.6% 0.36 0.50	5.3% 0.31 0.46	4.4% 0.29 0.46
	$\mathcal{M}_{x\beta}$	3.2%	7.2% 0.37 0.50	5.3% 0.32 0.46	4.9% 0.30 0.46
	Fusion	2.2%	4.5% 0.37 0.50	4.0% 0.32 0.46	3.3% 0.30 0.46
4	$\mathcal{M}_{x\theta}$	2.1%	5.5% 0.37 0.50	5.5% 0.31 0.45	4.8% 0.29 0.45
	$\mathcal{M}_{x\beta}$	2.2%	7.1% 0.37 0.50	6.5% 0.32 0.46	5.0% 0.30 0.46
	Fusion	1.3%	4.3% 0.37 0.50	4.3% 0.31 0.45	3.3% 0.30 0.45

Table 4.3: *EERs and BERs for the FVC2002 database. Enrollment methods E1 and E2.*

#images (t)		Analog	No HDS	ZLHDS	ZLHDS+r.c.
1	$\mathcal{M}_{x\theta}$	5.8%	12.1% 0.38 0.50	10.8% 0.37 0.50	8.8% 0.36 0.50
	$\mathcal{M}_{x\beta}$	6.4%	10.9% 0.39 0.50	10.9% 0.38 0.50	9.2% 0.36 0.50
	Fusion	5.5%	9.4% 0.39 0.50	9.3% 0.38 0.50	7.0% 0.36 0.50
2	$\mathcal{M}_{x\theta}$	5.4%	10.9% 0.39 0.50	9.8% 0.35 0.48	7.3% 0.33 0.48
	$\mathcal{M}_{x\beta}$	5.5%	10.7% 0.39 0.50	8.4% 0.36 0.48	7.4% 0.34 0.48
	Fusion	4.4%	9.8% 0.39 0.50	7.3% 0.36 0.48	5.9% 0.34 0.48

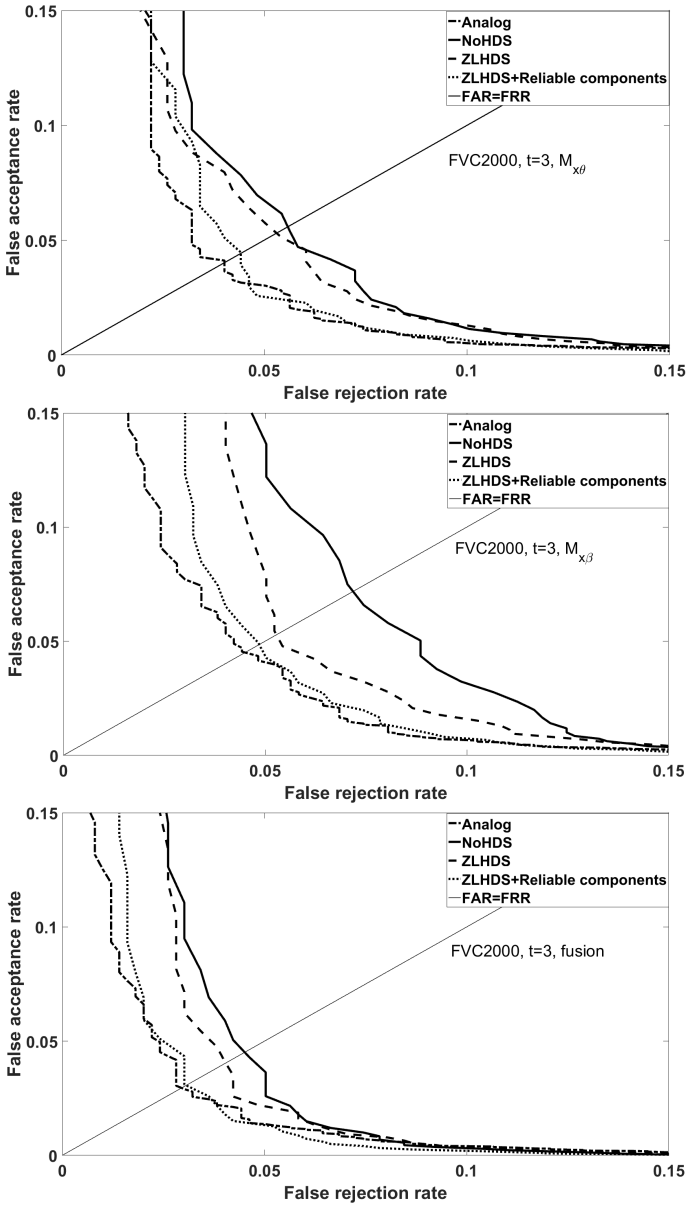


Figure 4.8: *Performance result for several processing methods. FVC2000. Enrollment method E2 with* $\mathrm{t} = 3$.

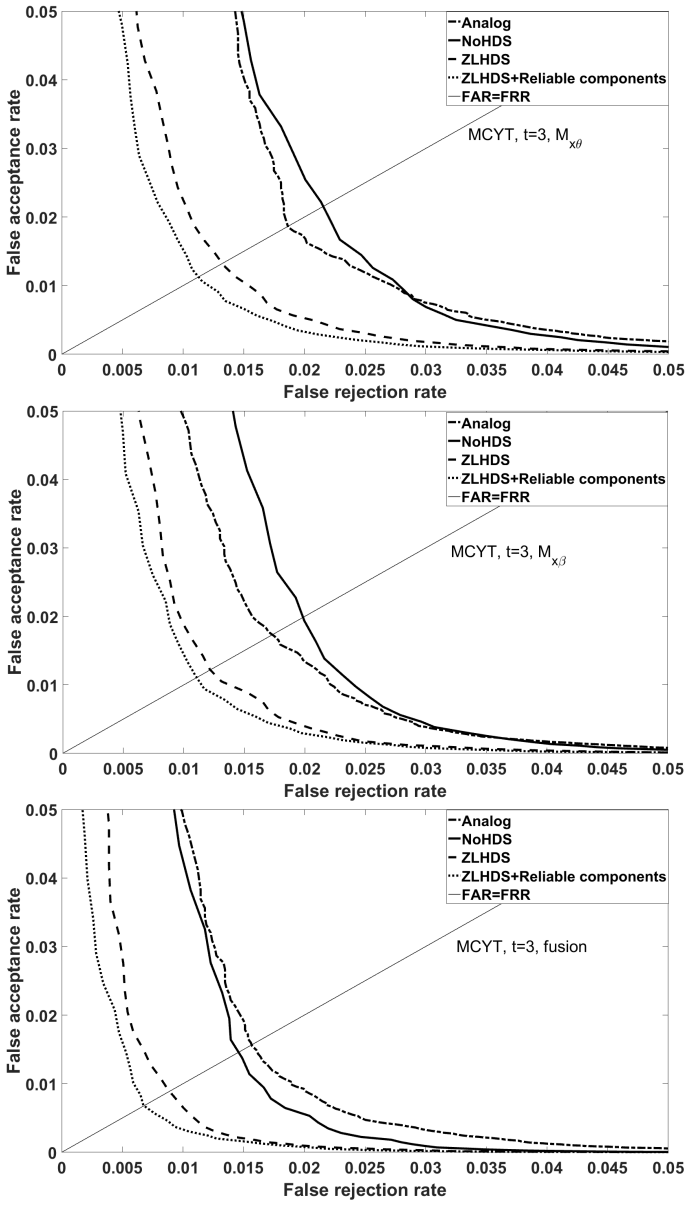


Figure 4.9: *Performance result for several processing methods. MCYT. Enrollment method E2 with* $\mathrm{t} = 3$.

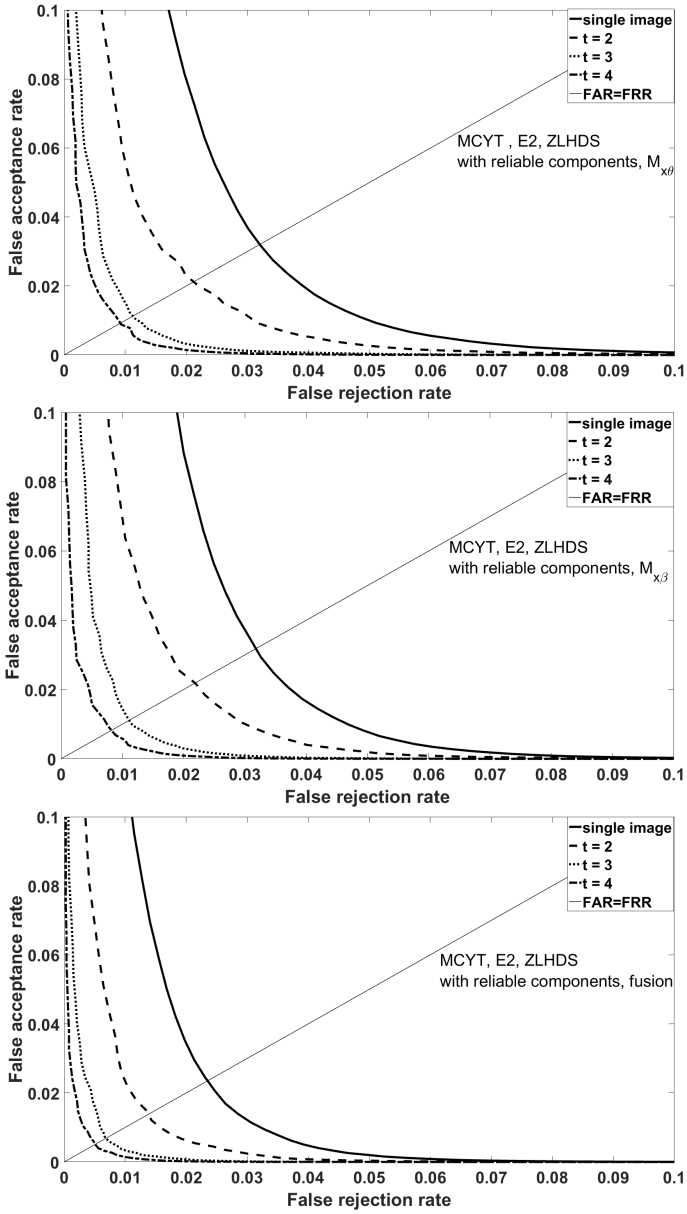


Figure 4.10: *Performance effect of the number of images used for enrollment.*

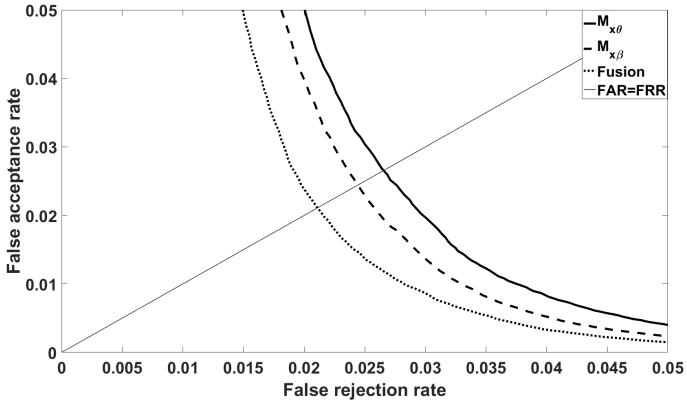


Figure 4.11: *Performance of* $\mathrm{M}_{x\theta}$ *and* $\mathrm{M}_{x\beta}$ *individually, and of their fusion. MCYT database; enrollment method E1; analog domain.*

Table 4.4: *EERs and BERs for the FVC2000 database. Enrollment method E3.*

#images (t)		Analog	No HDS	ZLHDS	ZLHDS+r.c.
3	$\mathcal{M}_{x\theta}$	3.0%	5.8% 0.37 0.50	5.2% 0.36 0.50	4.2% 0.34 0.50
	$\mathcal{M}_{x\beta}$	3.2%	8.1% 0.37 0.50	6.1% 0.36 0.50	5.4% 0.35 0.50
	Fusion	2.2%	5.3% 0.37 0.50	4.0% 0.36 0.50	3.1% 0.34 0.50

Table 4.5: *EERs and BERs for the MCYT database. Enrollment method E3.*

#images (t)		Analog	No HDS	ZLHDS	ZLHDS+r.c.
3	$\mathcal{M}_{x\theta}$	1.4%	2.4% 0.31 0.50	1.6% 0.29 0.49	1.4% 0.28 0.49
	$\mathcal{M}_{x\beta}$	1.1%	2.2% 0.32 0.50	1.5% 0.30 0.50	1.4% 0.28 0.50
	Fusion	1.1%	1.6% 0.32 0.50	1.0% 0.30 0.49	0.9% 0.28 0.50

4.6.2 – Error correction: Polar codes. The error rates in the genuine reconstructed $\hat{\mathsf{k}}$ are high, at least 0.21. In order to apply the Code Offset Method with a decent message size it is necessary to use a code that has a high rate even at small codeword length.

Consider the case of fusion of $\mathcal{M}_{\mathsf{x}\theta}$ and $\mathcal{M}_{\mathsf{x}\beta}$. The codeword length is 1280 bits (1024 if reliable component selection is performed). Suppose we need to distinguish between 2^{20} users. Then the message length needs to be at least 20 bits, in spite of the high bit error rate. Furthermore, the security of the template protection is determined by the entropy of the data that is input into the hash function (see Fig. 4.1); it would be preferable to have at least 64 bits of entropy.

We constructed a number of Polar codes tuned to the signal-to-noise ratios of the individual grid points. The codes are designed to find a set of reliable channels, which are then assigned to the information bits. Each code yields a certain FAR (impostor string accidentally decoding correctly) and FRR (genuine reconstruction string failing to decode correctly), and hence can be represented as a point in an ROC plot. This is shown in Fig. 4.12. For the MCYT database we have constructed a Polar code with message length 25 at an EER around 1.2% (compared to 0.7% before error correction). For the FVC2000 database we have constructed a Polar code with message length 15 at ≈6% EER (compared to 3.3% EER before error correction). Note that the error correction is an indispensable part of the privacy protection and inevitably leads to a performance penalty. However, we see that the penalty is not that bad, especially for high-quality fingerprints.

We briefly comment on the entropy contained in the extracted 'message' strings. In Section 4.8 we present a method to compute the upper bound on the entropy of a random vector, in the case where the probability distribution obeys a number of symmetries. This provides an upper bound to the mutual information between the k-bit strings extracted at enrollment and verification. We use this method to get a rough estimate for the actual systems at hand. For the MCYT database and message length 25, the message bit means vary between 0.49 and 0.51, and the off-diagonal elements of the covariance matrix vary between −0.02 and 0.02. Applying the method of Section 4.8 with constant off-diagonal covariance 0.02 yields an upper bound of 24.3 bits of entropy. For FVC2000 with message length 15 bits, the bit means vary between 0.44 and 0.58, and the off-diagonal elements of the covariance matrix have magnitudes below 0.04. Applying the method of Section 4.8 with constant off-diagonal covariance 0.04 yields an upper bound of 14.1 bits of entropy. The actual entropies may be a lot lower than the estimates that we give here. Because of these low entropies, the data extracted from multiple fingers needs to be combined in order to achieve a reasonable security level of the hash. We do not see this as a drawback of our HDS; given that the EER for one finger is

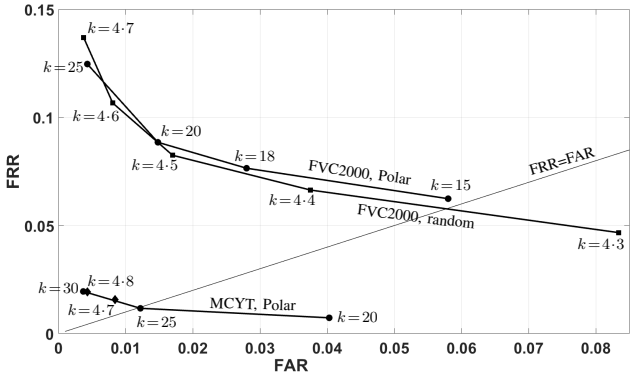


Figure 4.12: *FRR versus FAR achieved by Polar codes and random codebooks (average over three random codebooks). The ♦ markers denote random codebook, and happen to coincide with the line connecting the Polar markers.*

around 1%, which is impractical in real-life applications, *it is necessary anyhow to combine multiple fingers.* For comparison by using the results of [95] we obtain 6.5 bits of secrecy from ROC curves. However the model in [95] requires a good approximation of an ROC curve, which is very hard to obtain.

4.6.3 – Error correction: random codebooks. There is a large discrepancy between the message length of the Polar code ($k \leqslant 25$) and the known information content of a fingerprint. According to Ratha et al [71] the reproducible entropy of a fingerprint image with $Z = 35$ robustly detectable minutiae should be more than 120 bits. Furthermore, the potential message size that can be carried in a 1024-bit string with a BER of 23% is $1024[1 - h(0.23)] = 227$ bits. (And 122 bits at 30% BER.)

We experimented with random codebooks to see if we could extract more entropy from the data than with polar codes. At low code rates, a code based on random codewords can be practical to implement. Let the message size be ℓ, and the codeword size m. A random table needs to be stored of size $2^\ell \cdot m$ bits, and the process of decoding consists of computing 2^ℓ Hamming distances. We split the 1024 reliable bits into 4 groups of $m = 256$ bits, for which we generated random codebooks, for various values of ℓ. The total message size is $k = 4\ell$ and the total codeword size is $n = 4m$. The results are shown in Fig. 4.12. In short: random codebooks give hardly any improvement over Polar codes.

4.7 — Summary

We have built a HDS from a spectral function representation of fingerprint data, combined with a Zero Leakage quantization scheme. It turns out that the performance degradation w.r.t. unprotected templates is caused mainly by the step that maps a list of minutiae to spectral functions. The step from unprotected spectral functions to HDS-protected spectral functions is almost 'for free' in the case of high-quality fingerprints.
The best results were obtained with the 'superfinger' enrollment method (E2, taking the average over multiple enrollment images in the spectral function domain), and with fusion of the $\mathcal{M}_{x\theta}$, $\mathcal{M}_{x\beta}$ functions. The superfinger method performs slightly better than the E3 method and also has the advantage that it is not restricted to an odd number of enrollment captures.
For the high-quality MCYT database, our HDS achieves an EER around 1% and extracts an noise-robust 25-bit string that contains less than 24.3 bits of entropy. In practice multiple fingers need to be used in order to obtain an acceptable EER. This automatically increases the entropy of the hashed data. The entropy can be further increased by employing tricks like the Spammed Code Offset Method [100].

4.8 — Discussion

Any form of privacy protection (excepting perhaps homomorphic crypto) causes fingerprint matching degradation. Building a good template protection system is therefore an exercise in 'damage control': protect privacy while limiting the performance loss. We have pushed the two-dimensional spectral function approach to its limits, but even after the omission (in [82]) of the second invariant angle is corrected we still see that the transition from a minutia list to spectral functions destroys a lot of information. It remains a topic for future work to determine whether a higher-dimensional spectral function can retain more information while still yielding a practical template size. Given the experiences in [19] and the current paper, we expect that the ZLHDS privacy protection technique will do a good job there too, i.e. cause only little performance degradation, as long as the biometric data is of reasonable quality. We see that Polar codes perform extremely well at the high BER caused by noisy biometrics. Polar codes have been used in a HDS before [16], but under somewhat different circumstances, namely a simple a priori known noise distribution. The results of Section 4.6.2 demonstrate the efficiency of Polar codes also in the case where the noise distribution is unknown and has to be estimated from the training data.
We briefly comment on the computational effort of our scheme in the verification phase. The number of (complex-valued) summation terms in the computation of a spectral function is $\binom{Z}{2}N_{\text{grid}} \approx \binom{35}{2} \cdot 20 \cdot 16 = 1.9 \cdot 10^5$. The reconstruction step of the first-stage ZLHDS has negligible cost compared to that. Successive

Cancellation Decoding of Polar codes is lightweight, with complexity $\mathcal{O}(n \log n)$, $n = 1024$. On a modern processor, computing a hash takes less than 100 clock cycles per input byte. Clearly the bottleneck is the computation of the spectral functions. We have observed that reducing the number of grid points from the current $20 \cdot 16$ causes severe degradation of the matching performance, while increasing the number of points does not yield much improvement.
As topics for future work we mention (i) testing the HDS on more databases; (ii) further optimization of parameter choices such as the number of reliable components, and the number of minutiae used in the computation of the spectral functions; (iii) further tweaking of the Polar codes; (iv) other (spectral?) representations that cause less performance degradation while still allowing for a HDS to be constructed.

4.9 — Entropy upper bound

Let $X \in \{0,1\}^n$ be a random variable with probability mass function p_x. Using the Lagrange multiplier technique it is readily ascertained that the maximum-entropy distribution for X, for given first and second moment, must be of the Gaussian form $p_x \propto \exp[-a^T x - x^T M x]$, where x is interpreted as a column vector, a is a vector and M is a matrix. In general the a and M are very complicated functions of the first moments $m_i \stackrel{\text{def}}{=} \mathbb{E}x_i$ and the covariances $c_{ij} \stackrel{\text{def}}{=} \mathbb{E}x_i x_j - m_i m_j$ $(i \neq j)$. The computations become more tractable if we impose permutation invariance on X as well as $0 \leftrightarrow 1$ symbol symmetry. Then we have $p_x = N_\beta^{-1} \exp[\beta(|x| - \frac{n}{2})^2]$, where β is a parameter and the normalization constant N_β is defined as $N_\beta = \sum_{w=0}^{n} \binom{n}{w} \exp[\beta(w - \frac{n}{2})^2]$. Furthermore the imposed symmetries yield $m_i = \frac{1}{2}$ for all i, and constant covariance $c_{ij} = c$ for $i \neq j$. The relation between β and c is given by the 2nd moment constraint $N_\beta^{-1} \sum_{w=0}^{n} (w - \frac{n}{2})^2 \binom{n}{w} \exp[\beta(w - \frac{n}{2})^2] = \frac{n}{4} + (n^2 - n)c$. This equation has to be solved numerically for β. Then, with the numerical value of β, we can evaluate the entropy (in nats) as $\mathbb{E} \ln \frac{1}{p_x} = \ln N_\beta - \beta[\frac{n}{4} + (n^2 - n)c]$.

Chapter 5

Eliminating Leakage in Reverse Fuzzy Extractors

In the previous chapters we used HDSs for fingerprint template protection. In this chapter we use HDS for Physically Obfuscated Keys. The PUF takes the challenge and generates a noisy response and further the device uses the helper data for error correction. As mentioned in Chapter 1 error correction on the resource constrained device is problematic. This problem is solved by outsourcing the error correction (a trick often called 'reverse fuzzy extractor') to some external party. This approach has a drawback since each outsourcing of error correction reveals an error pattern. The PUF response may be leaked if the noise is data dependent. Another drawback is that the PUF may age which changes the response over time, and consequently the PUF becomes recognizable by its error pattern. This can lead to a privacy problem. Experimental data confirms the existence of asymmetric (data-dependent) noise and drift in several types of PUFs, e.g. RO PUFs. In this chapter we introduce two modifications to the Reverse FE scheme which together eliminate both leakage problems: (i) additional noise that turns asymmetric into symmetric noise. This solves the security problem; for PUFs with large noise asymmetry our approach leads to a reduction in channel capacity. The loss in channel capacity is approximately 30% which is acceptable for a practical key storage implementation. (ii) drift compensation by storing the estimated drift and recent error patterns in the prover device. This solves the privacy problem. Keeping track of only two error patterns already is enough to obtain an accurate drift estimator, thus demonstrating the efficiency of the proposed protocol.

This chapter is based on the paper

A. Schaller, T. Stanko, B. Škorić, and S. Katzenbeisser. Eliminating leakage in reverse fuzzy extractors. In *IEEE Transactions on Information Forensics and*

Security. 13, 4, pages 954-964, 2018.
I contributed to Sections III, IVc and V of the paper which are covered in Sections 5.3, 5.4.3 and 5.5 below.

5.1 — Introduction

In the past decade Physically Unclonable Functions (PUFs) have attracted increasing attention. With their desirable property of unclonability they were proposed as a promising security building block that can be applied to various identification and authentication applications. Several protocols featuring PUFs have been devised in the past, such as key storage, authentication and remote attestation schemes [74, 78, 93]. In this paper we focus on key storage, which is sometimes referred to as 'Physically Obfuscated Key'. In particular, we will consider the use of a PUF-based key storage in the context of privacy-preserving protocols that are designed to hide the identity of the users from eavesdroppers, such as low-cost anonymous access tokens.

PUFs are physical systems and thus their measurements always contain a certain amount of noise. However, cryptographic primitives like hashes and ciphers do not tolerate any noise. Thus, the noise in a PUF measurement must be removed before the measurement can be used as input to a cryptographic primitive. This introduces a complication: redundancy data (for the error correction) needs to be stored somewhere as part of the PUF enrollment data. The usual attacker model states that this redundancy data is public and thus can be accessed by the adversary. Hence, error correction needs to be designed such that the redundant data hardly leaks information about the PUF key. An error correction scheme that satisfies this requirement is variously known as Helper Data Scheme (HDS), Secure Sketch (SS) or Fuzzy Extractor (FE). FEs have the additional property that they generate a (nearly) uniform key. A FE can be trivially derived from a SS. One of the most popular HDSs is the Code Offset Method that employs a linear Error-Correcting Code (ECC). Particularly compact implementations are possible if *syndrome decoding* is used.

In many PUF applications the device containing the PUF is assumed to be resource-constrained. In the key reconstruction phase the device needs to perform an ECC decoding step, which may be infeasible given the constraints. An elegant solution was proposed in [39], where it was shown how the ECC decoding can be securely outsourced to a more powerful second party. The scheme was dubbed 'Reverse Fuzzy Extractor'. The most difficult HDS task for the device is now merely to compute a syndrome, which can be done very efficiently. On the downside, in each protocol run the Reverse FE reveals to eavesdroppers which error pattern is present in the PUF measurement, as compared to the enrollment measurement. In [39] it was argued that the PUF key is secure as long as the measurement noise is independent of the PUF value itself.

In this paper we (a) examine what happens when this assumption does not hold, i.e., we study the security implications of data-dependent noise; and (b) argue that the statement "the PUF key is secure as long as the measurement noise is data-independent", while true, does not cover all security aspects of the protocol.

Point (a) is important because data-dependent noise was shown to exist in PUFs such as FlipFlop PUFs, Latch PUFs and Buskeeper PUFs [94]. We use the Binary Asymmetric Channel (BAC) as our noise model. We quantify the leakage in this model and study possible countermeasures. It turns out that applying an extra Z-channel [32] after the BAC is a very effective solution.

Regarding point (b), we note that PUFs exhibit a slow 'drift' in the values of their response bits over time, which is due to device ageing. This drift is characteristic to individual PUF instances. A passive network attacker can try to identify PUF instances by analyzing the revealed error pattern, since the drift is directly reflected in the error pattern. This creates a *privacy* risk in case the PUF is used in privacy-preserving protocols, especially those that rely on the Reverse Fuzzy Extractor [6, 31], as the drift allows an attacker to link protocol executions from the same PUF. We show from experimental data that several PUF types indeed exhibit a drift. We propose an adaptation of the Reverse FE protocol that eliminates the drift issue.

5.1.1 – New Contributions. This paper is an extension to our publication [76], where we presented an evaluation of the systematic drift of Physically Unclonable Functions due to aging and further analyzed leakage involved. This version extends our previous work with the following contributions:

- We adopt the Binary Asymmetric Channel (BAC) as a noise model and provide detailed numbers on the potential leakage caused by the asymmetry of the noise.
- We propose an approach to eliminate the leakage. We apply artificial asymmetric noise. This results in two concatenated BACs which together form a symmetric channel. Due to the symmetry the leakage is entirely eliminated.
- The introduction of artificial noise leads to a loss of channel capacity. We estimate this loss.
- Finally, we propose a modified Reverse Fuzzy Extractor Protocol, which is resistant against leakage even if involved PUF instances exhibit drift.

The rest of the paper is organized as follows. In Section 5.2 we define notations, give a brief overview on PUFs and Fuzzy Extractors, introducing the Reverse Fuzzy Extractor in particular. In Section 5.3 we discuss the problem of data-dependent noise and describe our solution. In Section 5.4 we look at experimental data on drift and analyze the leakage caused by drift. We introduce an improved version of the Reverse Fuzzy Extractor in Section 5.5.

5.2 — Preliminaries

5.2.1 – Notation and terminology. The notation 'log' stands for the base-2 logarithm. Random variables are written in capital letters and their values in lowercase. The binary entropy function is written as

$$h(p) \stackrel{\text{def}}{=} -p \log p - (1-p)\log(1-p). \tag{5.1}$$

The Shannon entropy of a random variable X is denoted as $H(X)$, and mutual information as $I(X;Y)$.
The Binary Asymmetric Channel (BAC) is a memory-less channel. An transmitted bit X is received as a noisy bit X'. The channel is fully characterized by two parameters: $\alpha \stackrel{\text{def}}{=} \Pr[X'=1|X=0]$ and $\beta \stackrel{\text{def}}{=} \Pr[X'=0|X=1]$. Without loss of generality we consider only $\alpha, \beta \in [0, \frac{1}{2}]$. The case $\alpha = \beta$ is called the Binary Symmetric Channel (BSC). The case $\alpha = 0$ or $\beta = 0$ is known as a Z-channel. We will occasionally write $\alpha = \mu - \delta$, $\beta = \mu + \delta$, with $\mu \in [0, \frac{1}{2}]$ and $|\delta| \leqslant \min(\mu, \frac{1}{2} - \mu)$.

5.2.2 – Physically Unclonable Functions. A Physically Unclonable Function (PUF) is a complex physical structure that generates a response to a physical stimulus. The response depends on the challenge as well as on the micro- or nanoscale physical structure of the PUF itself. One typically assumes that the PUF can not be cloned, not even by the manufacturer of the device. Furthermore, the challenge-response behavior of the physical system is assumed to be complex enough such that the response to a given challenge can not be predicted.
Several different PUF constructions exist; for an overview we refer to [59]. Among them are memory-based PUFs, such as SRAM PUFs, which exploit biases in memory cells. At the power-up phase these cells initialize to either '0' or '1'. Most cells show a significant tendency to initialize to one of both values. The entirety of the the start-up values creates a start-up pattern, which is taken as PUF response. PUFs can also be based on random timing characteristics of circuits, among them Ring Oscillator PUFs and Arbiter PUFs.
Due to physical characteristics of the device, measurements of a PUF response are subject to noise; thus, subsequent measurements will be slightly different. In order to use them in cryptographic protocols, noisy responses must be stabilized. This is done by employing a Fuzzy Extractor [25], which extracts the stable part of the PUF response and transforms it to a uniformly distributed value.

5.2.3 – Fuzzy Extractors. The authors of [25] introduced Fuzzy Extractors as a means to deal with the noise. Commonly, Fuzzy Extractors work in two phases, a generation phase `Gen()` performed upon enrollment and a reconstruction phase `Rec()` performed after each measurement. During `Gen()`, a secret key K and a public Helper Data W are derived from a noisy PUF reference (enrollment) measurement X. The algorithm `Rec()` transforms a noisy PUF measurement X' back into the key K, thereby using the Helper Data W. This works as long

as X and X' are close enough (e.g., are two PUF measurements of the same challenge). Usually the reconstruction is achieved using an error correcting code.

5.2.4 – The Reverse Fuzzy Extractor. We briefly review the Reverse FE protocol [39].[1] We omit all details that are not critical for the key reconstruction itself (such as signal processing of the raw PUF data, or additional protection of the helper data, hashes of the key, quantities derived from the key, usage of the key). The Reverse Fuzzy Extractor is a two-party protocol which involves a prover, in possession of a (resource- constrained) PUF-enabled device, who wants to authenticate towards a computational powerful verifier. The description below is identical to the 'Syndrome-Only' Code Offset Method [7, 25] with the sole difference that syndrome decoding is outsourced to the verifier. A sequence diagram of the protocol is given in Figure 5.1.

System setup:

The parties agree on a linear error correcting code $\mathcal{C}$,with message length $\mathtt{m}$ and codeword length $\mathtt{n}$. The encoding algorithm of $\mathcal{C}$ is $\mathtt{Enc} : \{0,1\}^{\mathtt{m}} \to \{0,1\}^{\mathtt{n}}$, and the algorithm for computing the syndrome is denoted as $\mathtt{Syn} : \{0,1\}^{\mathtt{n}} \to \{0,1\}^{\mathtt{n}-\mathtt{m}}$. The code is chosen such that an efficient syndrome decoder $\mathtt{SynDec} : \{0,1\}^{\mathtt{n}-\mathtt{m}} \to \{0,1\}^{\mathtt{n}}$ exists. The parties also agree on a key derivation function $\mathtt{KeyDeriv} : \{0,1\}^{\mathtt{n}} \to \{0,1\}^{\mathtt{m}}$.

Enrollment:

A PUF enrollment measurement $\mathsf{X} \in \{0,1\}^{\mathtt{n}}$ is obtained. The helper data $W = \mathtt{Syn}(\mathsf{X})$ is computed. The prover stores W, while the verifier stores $\mathsf{K} = \mathtt{KeyDeriv}(\mathsf{X})$.

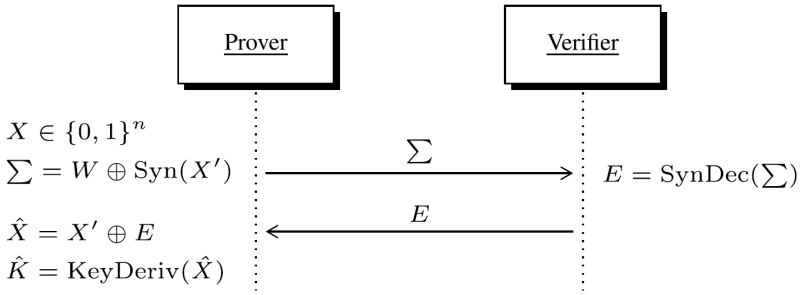


Figure 5.1: *Sequence diagram of the Reverse Fuzzy Extractor authentication protocol.*

[1] We will actually work with a more general primitive: a Secure Sketch. It is always possible to construct a Fuzzy Extractor from a Secure Sketch.

Reconstruction: The prover performs a fresh measurement $X' \in \{0,1\}^n$. He computes $\Sigma = W \oplus \mathtt{Syn}(X')$ and sends Σ to the verifier. The verifier computes the error pattern $E = \mathtt{SynDec}(\Sigma)$ and sends E to the prover. The prover computes the estimators $\hat{X} = X' \oplus E$ and $\hat{K} = \mathtt{KeyDeriv}(\hat{X})$.

Note that this protocol is extremely lightweight, as the prover only has to perform one $\mathtt{Syn}$ and one $\mathtt{KeyDeriv}$ operation. Note further that $\Sigma = \mathtt{Syn}(X \oplus X')$, due to the linearity of the code $\mathcal{C}$. Hence, if there is not too much noise, E is the error pattern that maps X' back to X and $\hat{X} = X$ during reconstruction.

Note, that one should not confuse asymmetric noise with biased PUF sources. It is known that a large bias in the distribution of the bit values of X causes problems if the code offset method is applied directly [40, 51], and various solutions have been suggested [21, 58, 80]. In this paper, however, we are concerned not about the distribution of the source X but about the asymmetry of the noise.

5.3 — Data-dependent noise

5.3.1 – Quantifying the problem. If the PUF noise is not independent of the measurement X, then some information about X is leaked to eavesdroppers via the error pattern E, during the reconstruction (Section 5.2.4). For instance, imagine that for a single bit of the PUF response a $0 \to 1$ transition is much more likely than a $1 \to 0$ transition. Then the error locations in E point to locations where a '0' in X is much more likely than a '1'. This is a *security* risk. It becomes even more serious if the adversary observes multiple transcripts from the same prover, carrying different information about X, and is able to link those transcripts together.

We adopt the Binary Asymmetric Channel (see Section 5.2.1) as our noise model and quantify the amount of leakage in this model. We further assume, for simplicity, that the bias is constant over the device, i.e., we consider a global bias.

Lemma 5.1. *Let $X \in \{0,1\}^n$ be the enrollment measurement, with i.i.d. bits $X_i \sim (1-p, p)$, i.e., all bits have the same bias $\Pr[X_i = 1] = p$. Let $X' \in \{0,1\}^n$ be the reconstruction measurement. Let the noise behave as a BAC. Let $E = X \oplus X'$ be the error pattern during reconstruction. Then the mutual information between the error pattern and X is given by*

$$I(X;E) = n\left[h\Big((1-p)\alpha + p\beta\Big) - (1-p)h(\alpha) - ph(\beta)\right] \tag{5.2}$$

and the entropy of X given E is

$$H(X|E) = nh(p) - I(X;E). \tag{5.3}$$

The proof is given in Appendix 5.7.1. Note that setting $\alpha = \beta$ in Lemma 5.1 gives $I(X; E) = 0$, i.e. if the noise is data-independent then an attacker learns nothing about X by observing E.
More generally, an attacker could observe multiple error vectors $E^{(1)}, \ldots, E^{(k)} \in \{0,1\}^n$ from the same device. We define T_i as the number of observations that yield an error in location i, i.e., $T_i = |\{k : E_i^{(k)} = 1\}| = \sum_{j=1}^{k} E_i^{(j)}$. We define $T = (T_i)_{i=1}^{n}$. The generalization of (5.2) then becomes

$$I(X; E^{(1)} \cdots E^{(k)}) = I(X; T) = H(T) - H(T|X) = nH(T_i) - nH(T_i|X_i),$$

where in the last line the index i is arbitrary. For the evaluation of $H(T_i|X_i)$ and $H(T_i)$ we need the corresponding probability distributions. For given X_i, T_i is binomial-distributed, thus $Pr[T_i = t|X_i = 0] = \binom{k}{t}\alpha^t(1-\alpha)^{k-t}$ and $Pr[T_i = t|X_i = 1] = \binom{k}{t}\beta^t(1-\beta)^{k-t}$. This yields $Pr[T_i = t] = (1-p)Pr[T_i = t|X_i = 0] + pPr[T_i = t|X_i = 1]$.
Figure 5.2 shows the leakage $nI(X_i; T_i)$ relative to the total information $nH(X_i) = nh(p)$ that can potentially be leaked, i.e., the leaked fraction. This is shown for various parameter settings of μ and k, where p has been tuned so as to maximize the attacker's uncertainty about X_i. By numerical methods we found the value of p (indicated as p*) that maximizes $H(X_i|T_i)$, where we used the above given probability distributions to compute $H(X_i|T_i)$ as $H(X_i) + H(T_i|X_i) - H(T_i)$. This is shown for various parameter settings of μ and k, where p has been tuned so as to maximize $H(X_i|T_i)$, the attacker's uncertainty about X_i. While k is the number of observed error instances observed by the attacker, μ is the average of α and β of the BAC, i.e., the average of the bit flip probabilities (see Section 5.2.1). The leakage is considerable. For example, in the $\mu = 0.05$ graph we see that already at $\delta = 0.025$ ten observations reveal almost 10% of the entropy of X. In order to connect Figure 5.2 to real-life PUFs, we evaluated PUF measurements of different PUF types regarding the extent of asymmetric noise and quantified the resulting leakage. For this purpose we leveraged the UNIQUE dataset [49], which contains measurements of different PUF types, including SRAM, latch, D-Flip-Flop (DFF), Arbiter and Ring Oscillator (RO) PUFs. Note that [49] conducted the standard analysis usually done for PUFs, which are agnostic of noise characteristics and does not asses whether the noise is symmetric or asymmetric. It uses the overall bit error rate $Pr[X = 0] * \alpha + Pr[X = 1] * \beta$. Table 5.1 lists values for μ and δ, which were computed by considering pairs of enrollment and reconstruction PUF measurements and applying the BAC model accordingly. In particular, for a given PUF type, we randomly selected an enrollment measurement at 20 °C and compared it with all reconstruction measurements taken at −40 °C, 20 °C and 80 °C. Values for α and β were computed by counting bit flips in the PUF measurement (see Section 5.2.1). Once values for (α, β) pairs were determined for each combination of PUF type and reconstruction temperature, corresponding μ and δ values were derived.

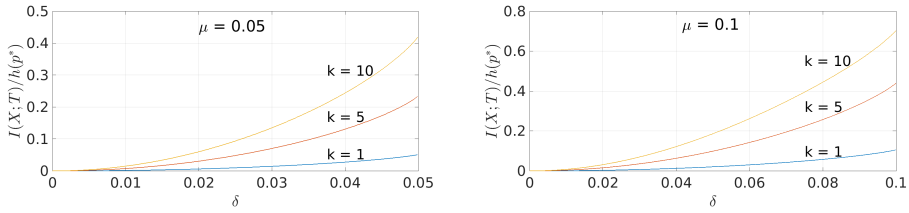


Figure 5.2: *The attacker's knowledge about* $\mathsf{X_i}$ *after observing* k *error instances, for various BAC parameters and various* k. *In each plotted point* p^* *is individually tuned to maximize* $\mathsf{H}(\mathsf{X_i}|\mathsf{T_i})$ *as a function of* μ, δ *and* k.

For Ring Oscillator (RO) PUFs, D-flip-flop (DFF) PUFs (high temperature) and Latch PUFs (low temperature), large values of $|\delta|$ up to 0.08, 0.17 and 0.2 respectively can occur (at $\mu \approx 0.2$), while SRAM PUFs have very little asymmetry with $|\delta| = 0.07, \mu = 0.002$. These are the 'raw' values before reliable component selection has been applied, or other processing, e.g. repetition codes, that reduces μ and δ. It is clear from Figure 5.2 that even after noise reduction residual asymmetries lead to significant leakage.

Table 5.1: *Values for* μ *and* δ, *given for various PUF types at different operational temperatures.*

Parameter	SRAM			LATCH		
	−40 °C	20 °C	80 °C	−40 °C	20 °C	80 °C
μ	0.0752	0.0548	0.0718	0.2435	0.0423	0.0914
$\|\delta\|$	0.0008	0.0002	0.0019	0.1953	0.0219	0.0175
	DFF			RO		
	−40 °C	20 °C	80 °C	−40 °C	20 °C	80 °C
μ	0.1312	0.0445	0.2076	0.2265	0.2093	0.2209
$\|\delta\|$	0.0035	0.0198	0.1650	0.0802	0.0737	0.0776

A naive attempt to deal with the leakage problem would be to tune the `KeyDeriv` function so that it compresses X more strongly, taking into account the expected leakage; however, there is no clear upper bound on the leakage, as the adversary can eavesdrop on additional protocol rounds.
Keeping in mind that even a few percent of key leakage can endanger the cryptographic primitives, we conclude that, no matter how `KeyDeriv` and the distribution of X are tuned, the Reverse FE has a serious leakage problem when

the noise is data-dependent.

5.3.2 – Eliminating the leakage. In order to eliminate the leakage, we propose a simple solution in the BAC case: to apply, in the reconstruction phase, an additional Z-channel that compensates the asymmetry in the measurement channel $\mathsf{X} \to \mathsf{X}'$. The parameters required for the Z-channel can be pre-computed based on calibration measurements which are done at system setup or at enrollment. The adapted reconstruction procedure is as follows.
Reconstruction:
1) The prover performs a fresh measurement $\mathsf{X}' \in \{0,1\}^n$. He applies additional Z-channel noise to X', yielding X''. He computes $\Sigma = W \oplus \mathtt{Syn}(\mathsf{X}'')$ and sends Σ to the verifier.
2) The verifier computes the error pattern $\mathsf{E} = \mathtt{SynDec}(\Sigma)$ and sends E to the prover.
3) The prover computes the estimators $\hat{\mathsf{X}} = \mathsf{X}'' \oplus \mathsf{E}$ and $\hat{\mathsf{K}} = \mathtt{KeyDeriv}(\hat{\mathsf{X}})$.

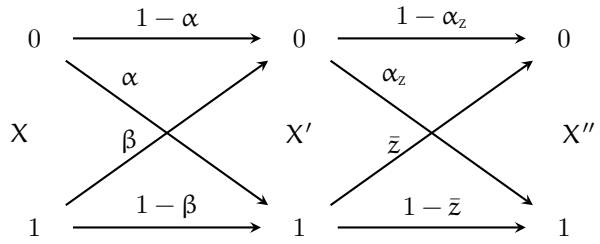


Figure 5.3: *Concatenation of two Binary Asymmetric Channels.*

We define the notation $\alpha_z = \Pr[\mathsf{X}''_i = 1 | \mathsf{X}'_i = 0]$ and $\bar{z} = \Pr[\mathsf{X}''_i = 0 | \mathsf{X}'_i = 1]$ for the Z-channel bit flip probabilities (see Figure 5.3). Note that at least one of the parameters $\alpha_z, \bar{z}$ is zero. The nonzero parameter is tuned such that the combined channel, consisting of the BAC with appended Z-channel, is a BSC. We will denote the bit error rate of this BSC as ε. The parameter tuning is given by the following theorem.

Theorem 5.2. *Let $\mathsf{X} \to \mathsf{X}'$ be a given BAC with parameters α, β (or μ, δ). Let $\mathsf{X}' \to \mathsf{X}''$ be a second BAC with parameters $\alpha_z, \bar{z}$ such that the combined channel $\mathsf{X} \to \mathsf{X}''$ is a BSC with bit error rate ε. Then ε is minimized by the following parameter choice:*

$$\alpha \geqslant \beta \ (\delta \leqslant 0): \qquad \alpha_z = 0,\ \bar{z} = \frac{\alpha - \beta}{1 + \alpha - \beta} = \frac{2|\delta|}{1 + 2|\delta|}$$

$$\beta \geqslant \alpha \ (\delta \geqslant 0): \qquad \bar{z} = 0,\ \alpha_z = \frac{\beta - \alpha}{1 + \beta - \alpha} = \frac{2\delta}{1 + 2\delta}.$$

Both cases yield $\varepsilon = \frac{\mu+|\delta|}{1+2|\delta|}$.

The proof is given in Appendix 5.7.2. Note that Theorem 5.2 does not assume that a Z-channel is the solution but starts more generically from a second BAC. Of course the legitimate parties need to estimate the noise parameters α, β of the BAC in order to be able to set $\alpha_z, \bar{z}$ as specified in Theorem 5.2. The noise parameters have to be established either (i) as part of a system setup phase before the enrollments or (ii) during operation, by using a subset of the PUF cells as non-secret test cells for calibration purposes.
Our solution entirely eliminates leakage from the communicated error pattern, but this comes at a cost: the additional noise degrades the channel, i.e., it reduces the amount of useful information about X that can be recovered after error correction. We now quantify how much 'worse' the channel $\mathsf{X} \to \mathsf{X}''$ is than the original channel $\mathsf{X} \to \mathsf{X}'$. First we show that the final noise parameter ε cannot be larger than the highest BAC parameter.

Corollary 5.3. *The bit-error probability* ε *specified in Theorem 5.2 satisfies*

$$\varepsilon \in \left[\frac{\alpha+\beta}{2}, \max\{\alpha,\beta\}\right] = [\mu, \mu+|\delta|]. \tag{5.4}$$

Proof. Let $\beta \geqslant \alpha (\delta \geqslant 0)$ w.l.o.g. From Theorem 5.2 we have $\varepsilon = \frac{\mu+\delta}{1+2\delta}$. Obviously, $\varepsilon \leqslant \mu+\delta$. Furthermore, $\varepsilon = \frac{\mu+\delta}{1+2\delta} = \frac{\mu(1+2\delta)+\delta-2\delta\mu}{1+2\delta} = \mu + \frac{\delta(1-2\mu)}{1+2\delta} \geqslant \mu$.In the last step we used $0 \leqslant \mu \leqslant \frac{1}{2}$. Thus, we obtain $\varepsilon \in [\mu, \mu+\delta] = \left[\frac{\alpha+\beta}{2}, \beta\right]$. The derivation for $\delta < 0$ follows exactly the same lines. □

Next we characterize the loss of channel quality by looking at the *channel capacity*. The channel capacity places a lower bound on how much source entropy (from X) is required to derive a noise-robust key of a certain size. A capacity equal to 1 corresponds to absence of noise, in which case all entropy from X is directly usable. In general, the capacity C is the *fraction* of all the entropy in X that may survive error correction in case of an ideal error-correcting code.
The BSC $\mathsf{X} \to \mathsf{X}''$ has capacity $\mathsf{C}_{\mathsf{BSC}} = 1 - \mathrm{h}(\varepsilon)$, with $\varepsilon = \frac{\mu+|\delta|}{1+2|\delta|}$ as specified in Theorem 5.2. The capacity of the BAC is given by (see e.g. [62]),

$$\begin{aligned} \mathsf{C}_{\mathsf{BAC}} &= \frac{\mu-|\delta|}{1-2\mu}\mathrm{h}(\mu+|\delta|) - \frac{1-\mu-|\delta|}{1-2\mu}\mathrm{h}(\mu-|\delta|) \\ &\quad + \log\left(1 + 2^{-\frac{\mathrm{h}(\mu+|\delta|)-\mathrm{h}(\mu-|\delta|)}{1-2\mu}}\right) \qquad (5.5) \\ &= 1 - \mathrm{h}(\mu) + (\frac{\delta}{\mu})^2 \frac{\mu}{2\ln 2} + \mathcal{O}(\frac{\delta^2}{\mu^2}[\mu \ln \mu]^2). \qquad (5.6) \end{aligned}$$

Theorem 5.4. *The capacity loss due to introducing the Z-channel can be approximated as*

$$\approx C_{BAC} - C_{BSC} = |\delta|(1-2\mu)\log\frac{1-\mu}{\mu} + \mathcal{O}(\delta^2) = |\delta|\log\frac{1}{\mu} + \mathcal{O}(\delta\mu\log\frac{1}{\mu}). \quad (5.7)$$

Proof. Follows from Taylor-expanding the expressions for C_{BAC} and C_{BSC}. □

In Figure 5.4 we plot the capacity loss $C_{BAC} - C_{BSC}$ relative to the original capacity C_{BAC}. The 'raw' noise levels in PUFs (i.e., without reliable cell selection) for different PUF types. As shown in Table 5.1, D-flip-flop PUFs can have high noise levels up to $\mu = 0.2, \delta = 0.1$ [94]. In this case and according to Figure 5.4, the Z-channel insertion would then lead to almost 40% capacity loss. In contrast, SRAM PUFs exhibit comparably little asymmetry with $\mu = 0.07, \delta = 0.02$, which results in less then 5% capacity loss. Depending on the context this may be acceptable. If not, the noise μ, δ can be reduced, as in Figure 5.4 we see that the capacity loss is less severe at low noise. Reduction of noise due to μ, δ asymmetry can be achieved by techniques such as reliable component selection and repetition codes. The optimal tuning of the parameters in the noise reduction techniques depends on the specific PUF properties.
The consequence of reduced channel capacity in practical scenarios lies in the fact that more PUF material is needed due to this loss of channel capacity . In fact, the channel capacity is inversely proportional to the size of required PUF input bits.

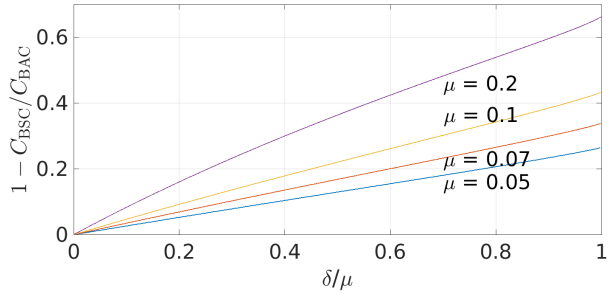


Figure 5.4: *Relative loss of channel capacity due to the extra Z-channel.*

5.4 — The drift problem

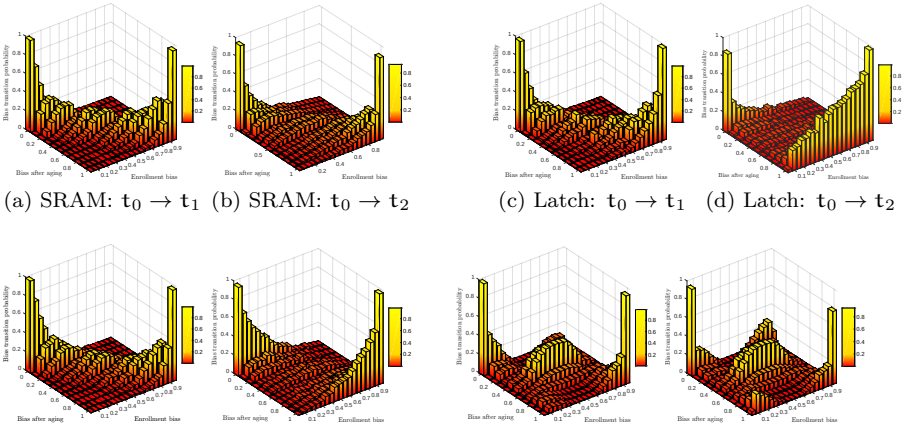
In some PUF instances individual cells have a bias towards either zero or one. We present measurements which show that these biases change over time; we call this the *drift* of a PUF. Furthermore, we provide a model for the drift and estimate the privacy leakage (and the induced key leakage) in Reverse Fuzzy Extractors due to the drift.

5.4.1 – Drift model. We adopt the bias-based PUF model proposed in [94]. We define the bias of a PUF cell i to be the probability that the cell ends up in a '1' state and denote the bias during enrollment as a value $b_i \in [0,1]$; it can be estimated by counting the number x_i of occurrences of a '1' response during k enrollment measurements: $\hat{b}_i = x_i/k \in \{0, \frac{1}{k}, \ldots, 1\}$. A PUF is fully characterized by a vector of biases, $\mathbf{b} = (b_i)i = 1^n$. Similarly, b'_i represents the bias of cell i at a later time. It can be estimated from the number x'_i of '1' responses in a series of l PUF responses: $\hat{b}'_i = x'_i/l \in \{0, \frac{1}{l}, \ldots, 1\}$.
The drift is modeled by a transition probability $\tau(\mathbf{b}'|\mathbf{b})$ indicating how likely it is that the PUF has bias vector $\mathbf{b}'$ at a later time given that it had $\mathbf{b}$ at enrollment. Assuming that the n PUF cell responses are mutually independent (this assumption seems justified as we did not see any correlation between cell responses in the PUF types under investigation [17,94]), and that drift behavior is the same for all bits, we can express the transition probability for the entire PUF as

$$\tau(\mathbf{b}'|\mathbf{b}) = \prod_{i=1}^{n} \tau_0(b'_i|b_i). \tag{5.8}$$

The function τ_0 does not depend on the cell index i. To estimate τ_0 we made histograms of drifted biases, conditioned on the enrolled bias, i.e., for each possible value of $\hat{b}_i$ we computed a histogram counting $\hat{b}_i \to \hat{b}'_i$ occurrences. Here the $\hat{b}_i \to \hat{b}'_i$ transitions were collected from all cells. Finally, we converted the histograms to probability distributions.

5.4.2 – Drift data. We made use of PUF measurement data obtained in the UNIQUE project [49]. In this project custom ASICS with different PUF types, including SRAM, latch, D-Flip-Flop (DFF), Arbiter and Ring Oscillator (RO) PUFs, were developed and tested under different conditions. The UNIQUE data set includes measurements of PUFs which were exposed to an accelerated aging process. The simulation of aging is based on the Negative Bias Temperature Instability (NBTI) mechanism, carried out by operating the ASICs at an extreme temperature of $+85\,°C$ and with high supply voltage of 1.44V (120% of the 1.2V standard V_{dd}). The treatment lasted for 2150 hours corresponding to an aging factor of 18.2. This way, continuous use of the PUF device can be simulated in short time.
Three different datasets were available for our experiments: enrollment data taken right after manufacturing (referred to as time t_0), measurements at the beginning of the aging process (at time t_1) and measurements after the aging process had terminated (time t_2). Measurements at t_1 correspond to a simulated operating time of approximately 1 week with respect to t_0 whilst t_2 corresponds to approximately 4.5 years. For our bias transition model we compared t_0 versus t_1 and t_0 versus t_2.



(a) SRAM: $\mathtt{t}_0 \to \mathtt{t}_1$ (b) SRAM: $\mathtt{t}_0 \to \mathtt{t}_2$ (c) Latch: $\mathtt{t}_0 \to \mathtt{t}_1$ (d) Latch: $\mathtt{t}_0 \to \mathtt{t}_2$

(e) DFF: $\mathtt{t}_0 \to \mathtt{t}_1$ (f) DFF: $\mathtt{t}_0 \to \mathtt{t}_2$ (g) RO: $\mathtt{t}_0 \to \mathtt{t}_1$ (h) RO: $\mathtt{t}_0 \to \mathtt{t}_2$

Figure 5.5: *Bias transition probabilities* $\tau_0(\mathtt{b}'|\mathtt{b})$ *for SRAM, latch, DFF and RO PUFs corresponding to different time intervals (*$\mathtt{t}_0 \to \mathtt{t}_1$ *and* $\mathtt{t}_0 \to \mathtt{t}_2$*).*

Figure 5.5 shows τ_0 for SRAM, latch, DFF and RO PUFs, for $\mathtt{t}_0 \to \mathtt{t}_1$ and $\mathtt{t}_0 \to \mathtt{t}_2$. In Figures 5.5a, 5.5c and 5.5e we observe a diagonal 'saddle' between $(0, 0)$ and $(1, 1)$ for the $\mathtt{t}_0 \to \mathtt{t}_1$ data. This indicates that SRAM, latch and DFF PUFs have a stable bias over a short operating time. The RO PUF (Figure 5.5g) is an exception, featuring an 'island' of high probabilities in the middle of the plot area, indicating more transitions to bias 0.5 (random behavior); this is not unexpected, as ring oscillators can be used to generate random numbers as well. For the transition $\mathtt{t}_0 \to \mathtt{t}_2$ we see a flattening of the 'saddle' for all PUF types (Figures 5.5b, 5.5d, 5.5f and 5.5h). This indicates, as expected, that there is a significant drift after a longer operation time. Note that not all transition probabilities are symmetric under $0 \leftrightarrow 1$ reversal; this phenomenon mainly occurs for the latch and RO PUFs.

The FE reconstruction phase typically employs only a single measurement ($\mathtt{l} = 1$). Hence, in practice FEs usually do not use fine-grained information about biases during reconstruction. Instead, fine-grained bias information is used only for the selection of reliable cells. A FE will typically store pointers to stable cells (i.e., cells that have an enrollment bias close to '0' or '1'); only those are then used for key derivation.

For this context we introduce a simplified drift model in which the biases are binarized to 0/1 values, and only reliable PUF cells are taken into account. For this purpose, we regard cells as reliable, if they observe an enrollment bias $\mathtt{b}_\mathtt{i} \in [0, 0.05] \cup [0.95, 1]$. Although the intervals that define stable components are chosen somewhat arbitrarily, it turns out to be a workable choice. The

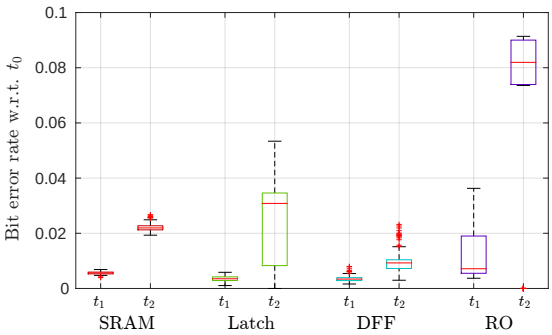


Figure 5.6: *Bit error rates of stable cells for various PUF types after* $\mathtt{t}_0 \to \mathtt{t}_1$ *and* $\mathtt{t}_0 \to \mathtt{t}_2$ *aging. The red line in each box indicates the median. The colored bottom and top of each box marks the* $25^{\mathrm{th}}/75^{\mathrm{th}}$ *percentile. The height of a box displays the inter quartile range (IQR). The whisker's ends indicate the lowest and highest bit error rates that are within 1.5 times the IQR. Single outsider values are marked by red plus signs.*

model has only two parameters: ad, the probability of a $0 \to 1$ bit transition due to drift, and β_d, the probability of a $1 \to 0$ transition due to drift. The numerical values of these parameters slowly vary as a function of time. Table 5.2 lists the transition probabilities of reliable cells, based on the empirical data from the UNIQUE project, and Figure 5.6 shows the same data as bit error rates graphically.[2] For SRAM and RO PUFs, the $\mathtt{t}_0 \to \mathtt{t}_2$ bit error rate is considerably higher than the $\mathtt{t}_0 \to \mathtt{t}_1$ bit error rate.

Table 5.2: *Transition probabilities* $0 \to 1$ *and* $1 \to 0$ *for* $\mathtt{t}_0 \to \mathtt{t}_1$ *and* $\mathtt{t}_0 \to \mathtt{t}_2$ *aging (biases* $\mathtt{b}_i \in [0, 0.05] \cup [0.95, 1]$*. (Mean±standard deviation)* $\times 10^{-4}$ *is listed).*

Period	Transition	SRAM	LATCH	DFF	RO
$\mathtt{t}_0 \to \mathtt{t}_1$	$0 \to 1$	36 ± 2	25 ± 7	17 ± 3	76 ± 55
	$1 \to 0$	20 ± 1	12 ± 3	19 ± 4	50 ± 43
$\mathtt{t}_0 \to \mathtt{t}_2$	$0 \to 1$	130 ± 4	222 ± 135	62 ± 28	350 ± 199
	$1 \to 0$	91 ± 4	41 ± 38	29 ± 10	323 ± 184

5.4.3 – Leakage analysis. The results of Section 5.4.2 show that aging indeed causes drifting of the PUF measurement X' over time. Thus, the noise $\mathsf{E} = \mathsf{X}' \oplus \mathsf{X}$

[2] Note that this figure incorporates fixed values,which have been found to be erroneous in [76].

in the Reverse FE protocol contains a part $\mathsf{D} \in \{0,1\}^n$ (the drift) that changes only over long time scales, while the rest of E consists of short-timescale random noise N unrelated to aging. We can represent E as $\mathsf{E} = \mathsf{D} \oplus \mathsf{N}$. This both has an impact on security and privacy.

Privacy. We first quantify the privacy leakage of the Reverse FE protocol caused by observation of the drift. A privacy leakage occurs due to device unique drift that is manifested in the public error patterns. Privacy-preserving protocols [6, 31] that leverage the Reverse FE therefore are vulnerable to a passive network attacker. By eavesdropping the communicated error patterns, the attacker is able to link *multiple* protocol executions to individual PUF instances. Equation (5.10) gives the exact information he gains about the drift from observing a set of error patterns. Thus, the attacker is able to effectively undermine potential privacy-preserving mechanisms in place.

Lemma 5.5. *Let* X_1 *and* X_2 *be the enrollment measurements of two different PUFs, uniformly distributed on* $\{0,1\}^n$. *Let* D_1 *and* D_2 *be their respective drifts. Let the drift be independent in each bit, with parameters* ad, β_d *as defined above. Then the Hamming distance between* D_1 *and* D_2 *is binomial-distributed, with parameters* n *and* P_{uneq}, *where*

$$\mathsf{P}_{\text{uneq}} = 2\frac{\mathsf{ad} + \beta_d}{2}\left(1 - \frac{\mathsf{ad} + \beta_d}{2}\right). \tag{5.9}$$

The proof is given in Appendix 5.7.3.

The following corollary shows how the uniqueness of individual drifts can be quantized in terms of Hamming distance.

Corollary 5.6. *Let* X_1 *and* X_2 *be the enrollment measurements of two different PUFs, uniformly distributed on* $\{0,1\}^n$. *Let* D_1 *and* D_2 *be their respective drifts. Let the drift be independent in each bit, with parameters* ad, β_d. *Then the expected Hamming distance between* D_1 *and* D_2 *is* $\mu_{\text{HD}} = n\mathsf{P}_{\text{uneq}}$, *and the variance is* $\sigma^2_{\text{HD}} = n\mathsf{P}_{\text{uneq}}(1 - \mathsf{P}_{\text{uneq}})$.

Proof. Follows from Lemma 5.5 and the properties of the binomial distribution. □

If the short-timescale noise N does not mask the drift, then the observed noise pattern E, via the constant part D, becomes a unique characterizing property for each PUF, as quantified in Corollary 5.6.

For the further analysis we introduce the following notation. Given multiple observations of the protocol run, we denote the set of observed error patterns as $\mathcal{E} = (\mathsf{E}_a)_{a=1}^k$, where k is the number of observations. Similarly, we define $\mathcal{N} = (\mathsf{N}_a)_{a=1}^k$, with $\mathsf{E}_a = \mathsf{D} \oplus \mathsf{N}_a$. We write $\mathsf{X}_{\text{drifted}} = \mathsf{X} \oplus \mathsf{D}$.

Theorem 5.7. *Let* $\mathcal{N}_{av} \in \{0,1\}^n$ *be the pattern obtained by averaging* $\mathcal{N}$*:* $\mathcal{N}_{av} = [\frac{N_a}{k}]$*. The amount of information about* D *gained from observing* $\mathcal{E}$ *is given by*

$$I(\mathcal{E}; D) = H(D \oplus \mathcal{N}_{av}) - H(\mathcal{N}_{av}). \tag{5.10}$$

Proof: see Appendix 5.7.4.
If the noise N_a is data-independent, then the adversary can get a good estimate of D by averaging the error patterns, and we can almost say that observing $\mathcal{E}$ is the same as observing D and $\mathcal{E}$ (or, equivalently, D and $\mathcal{N}$). The D can be used by the attacker as an identifier. In the case of data-dependent noise, N_a leaks information about $X_{drifted}$. This too can be used by the attacker as an identifier.
Security. Next we analyze the security implications if the adversary is able to link multiple instances of the authentication protocol run by the same PUF device. (Either because of the above explained privacy problem or by some other means.)
Since we did not specify the `KeyDeriv` algorithm, we cannot compute the mutual entropy between $\mathcal{E}$ and the PUF key K in general. Instead, we derive a bound on the mutual information between $\mathcal{E}$ and X.

Theorem 5.8. *The leakage about* X *caused by observation of the error patterns* $\mathcal{E}$ *can be upper bounded as*

$$I(\mathcal{E}; X) \leqslant I(D; X) + I(\mathcal{N}; X_{drifted}), \tag{5.11}$$

Proof: see Appendix 5.7.5.
The two leakage terms in Theorem 5.8 are very similar. The $I(\mathcal{N}; X_{drifted})$ term is exactly the leakage shown in Figure 5.2, but now about $X_{drifted}$ instead of X, which is practically the same from a security point of view, since the attacker has access to D. The mutual information $I(D; X)$ is precisely given by Lemma 5.1 where the error pattern E is replaced by the drift D, and the parameters α, β by ad, β_d. The $I(D; X)$ is nonzero if the drift is asymmetric.
Note that, in contrast to the leakage $I(\mathcal{N}; X_{drifted})$, the existence of the $I(D; X)$ leakage does not necessarily imply that there is a grave security problem: The drift D is a single error pattern, whereas measurements of short-term asymmetric noise reveal new information every time. A properly designed extraction procedure `KeyDeriv` can compensate for the leakage $I(D; X)$ by sufficiently compressing X. In case privacy is not important, we see the leakage $I(D; X)$ primarily as an issue that reduces the efficiency of the Fuzzy Extractor.
Finally we briefly comment on the case where the adversary observes the helper data W as well as the communicated noise patterns $\mathcal{E}$.

Theorem 5.9. *The leakage caused by observing* W *and* $\mathcal{E}$ *can be bounded as*

$$I(W\mathcal{E}; X) \leqslant I(W; X) + I(\mathcal{E}; X). \tag{5.12}$$

Proof: See Appendix 5.7.6.

The bound in Theorem 5.9 is tight, since $\mathsf{H}(\mathcal{E}|W) \approx \mathsf{H}(\mathcal{E})$. Thus we can also read Theorem 5.9 as $\mathsf{I}(W\mathcal{E};\mathsf{X}) \approx \mathsf{I}(W;\mathsf{X}) + \mathsf{I}(\mathcal{E};\mathsf{X})$, i.e., leakage from W plus *almost independent* leakage from $\mathcal{E}$.

5.5 — Solving the drift problem

In this section we present a modified Reverse Fuzzy Extractor in which the protocol messages do not cause leakage, even if there is PUF drift. The modified protocol assumes a passive network adversary, who is able to observe the channel between the prover and the verifier and hence is able to capture the communicated error patterns E. In the modified Reverse Fuzzy extractor the prover keeps track of the computed error patterns E over time. If E starts to exhibit behavior constant in time (a drift D), then the prover device modifies its stored helper data in such a way that the drift is compensated; future error patterns E will thus not reveal the drift. This technique is compatible with the addition of a Z-channel as described in Section 5.3.2.

5.5.1 – Proposed solution for the drift problem. In a nutshell our proposal is as follows. The prover device has additional non-volatile memory in which it stores an estimated drift vector $\hat{\mathsf{D}} \in \{0,1\}^n$ and a list $\mathcal{L}$ of up to $\mathsf{N}_{\max}$ error patterns observed during previous executions of the protocol. The $\hat{\mathsf{D}}$ serves to keep track of how far the PUF has drifted away from the enrolled PUF measurement X. The reconstruction protocol does error correction with respect to the (estimated) drifted PUF value $\hat{\mathsf{X}}_{\text{drifted}}$, and then shifts the result by the amount of $\hat{\mathsf{D}}$. Taking the drifted value $\hat{\mathsf{X}}_{\text{drifted}}$ as the zero point for error correction has the additional advantage that the number of bit errors is reduced. The stored helper data is always equal to $\tilde{W} = \mathtt{Syn}(\hat{\mathsf{X}}_{\text{drifted}})$. A detailed description of our proposal is given below.

System setup:
The same as in Section 5.2.4.

Enrollment:
The same as in Section 5.2.4. The enrolled helper data is $\tilde{W} = \mathtt{Syn}(\mathsf{X})$. In addition, the prover's list $\mathcal{L}$ is initialized to the empty string $\emptyset$, and $\hat{\mathsf{D}}$ is initialized to the zero string.

Reconstruction:
1) The prover
   1. performs a fresh measurement $\mathsf{Y} \in \{0,1\}^n$,
   2. adds (pseudo-)random Z-channel noise R, yielding $\mathsf{Y}' = \mathsf{Y} \oplus \mathsf{R}$.
   3. computes $\Sigma = \tilde{W} \oplus \mathtt{Syn}(\mathsf{Y}')$ and sends Σ to the verifier.

2) The verifier computes the error pattern $\tilde{\mathsf{E}} = \mathtt{SynDec}(\Sigma)$ and sends $\tilde{\mathsf{E}}$ to the prover.

3) The prover computes $\hat{X}_{\text{drifted}} = Y' \oplus \tilde{E}$ and the estimators $\hat{X} = \hat{X}_{\text{drifted}} \oplus \hat{D}$ and $\hat{K} = \texttt{KeyDeriv}(\hat{X})$.
4) If $\hat{K} = K$ then the prover performs the following actions.
   1. Add the error pattern $\tilde{E} \oplus R$ to the list $\mathcal{L}$. If necessary, the oldest entry in $\mathcal{L}$ is discarded to make place.
   2. If $\mathcal{L}$ contains $N_{\max}$ entries, check if there are bit positions that are '1' in the majority of the entries. If so, construct an error pattern $e \in \{0,1\}^n$ consisting of these positions, replace $\hat{D}$ by $\hat{D} \oplus e$, and replace the helper data $\tilde{W}$ by $\texttt{Syn}(\hat{X}_{\text{drifted}} \oplus e)$. Xor all entries in $\mathcal{L}$ with e.

5.5.2 – Privacy of the proposed protocol. We have $Y' = X \oplus D \oplus N \oplus R$, where N is short-timescale BAC noise, and (in case of correct reconstruction of X) we have $\hat{X}_{\text{drifted}} = X \oplus \hat{D}$. This gives

$$\tilde{E} = Y' \oplus \hat{X}_{\text{drifted}} = (D \oplus \hat{D}) \oplus (N \oplus R). \tag{5.13}$$

Thus, the error pattern $\tilde{E}$ observed by the adversary is a combination of (i) Z-channel-compensated (and hence symmetric) short-timescale noise $N \oplus R$, and (ii) a small long-timescale component $D \oplus \hat{D}$ which vanishes if the estimator $\hat{D}$ is accurate.
Given an accurate $\hat{D}$, there is no long-timescale structure to be observed in $\tilde{E}$. Furthermore, the symmetry of the noise $N \oplus R$ (as opposed to N) guarantees that the adversary learns nothing about the data X_{drifted}. Thus, both privacy aspects are solved.
We checked the accuracy of the estimator $\hat{D}$ of D, by simulating the proposed protocol on the same data that was used for evaluating the systematic drift in Section 5.4.2. Figure 5.7 shows the fractional Hamming distance between $\hat{D}$ and D as a function of $N_{\max}/20$ for various PUF types at time periods t_1 and t_2. In particular, we evaluated the protocol on the following PUF types and their respective PUF averages of 40 individual PUF instances. As expected, with increasing $N_{\max}$, the accuracy of $\hat{D}$ improves up to the point where the entire data set is considered ($N_{\max} = 20$), resulting in $\hat{D} \approx D$, i.e., a fractional Hamming distance close to zero. Note, that the accuracy of $\hat{D}$ is not exactly zero due to quantization noise. The results show that $\hat{D}$ deviates only by 2% from the actual drift in the worst case (considering only two measurements), for most of the PUFs. Only the RO PUFs, exhibiting very large asymmetry, of up to 10% deviation. If the prover implements sufficient memory to store even more error pattern instances, accuracy of estimator $\hat{D}$ can be further improved. In our experiments deviation of $\hat{D}$ is already at 0.5% when using 20 measurements for SRAM, Latch and DFF PUFs. For RO PUFs, more then 20 measurements must be stored, to limit deviation of $\hat{D}$ to under 5%.
Thus, a prover keeping track of only two error patterns already results in estimator $\hat{D}$ that is accurate enough, in order to mask the long-timescale drift, thus demonstrating the efficiency of the proposed protocol.

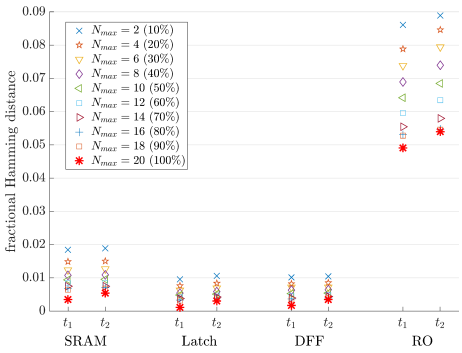


Figure 5.7: *Accuracy of the approximated drift vector* $\hat{\mathsf{D}}$ *compared to the actual drift vector* D*, given as fractional Hamming distances. A Hamming distance of zero indicates a perfect estimator* $\hat{\mathsf{D}}$*. The actual drift* D *is based on 20 reconstruction measurements, taken at time periods* t_1 *&* t_2*. The percentage values depict the fraction of reconstruction measurements used for computation of* $\hat{\mathsf{D}}$*. Fractional Hamming distances are given as mean values over 40 PUF instances.*

5.5.3 – Security of the proposed protocol. As mentioned above, an adversary who observes $\tilde{\mathsf{E}}$ learns nothing about $\mathsf{X}_{\text{drifted}}$; this is the case because of the Z-channel insertion just as in Section 5.3.2. However, in the new protocol we have to store additional public data $\hat{\mathsf{D}}, \mathcal{L}$ together with $\tilde{W}$ in memory. We study how the security is affected by this additional information.

Theorem 5.10. *An adversary who observes the prover's memory has the following amount of information about* X*,*

$$\mathrm{I}(\mathsf{X}; \tilde{W}\hat{\mathsf{D}}\mathcal{L}) = \mathrm{I}(\mathsf{X}; W) + [\mathrm{H}(\tilde{W}|\hat{\mathsf{D}}) - \mathrm{H}(W)] + \mathrm{I}(\hat{\mathsf{D}}; \mathsf{X}) + [\mathrm{H}(\mathcal{L}|\hat{\mathsf{D}}\tilde{W}) - \mathrm{H}(\mathcal{L}|\mathsf{X}_{\text{drifted}})].$$

In Theorem 5.10, the term $\mathrm{I}(\mathsf{X}; W)$ is the 'known' result, for the ordinary code offset method. The corresponding proof can be found in the Appendix.

- The $\mathrm{I}(\hat{\mathsf{D}}; \mathsf{X})$ is nonzero if the drift is asymmetric. As mentioned in Section 5.4.3, a nonzero leakage here is not a severe problem and can be dealt with by properly choosing the parameters in the function `KeyDeriv`.
- The term $\mathrm{H}(\tilde{W}|\hat{\mathsf{D}}) - \mathrm{H}(W)$ is small, since $\tilde{W}$ equals the original helper data W compensated by the drift.
- The term $\mathrm{H}(\mathcal{L}|\hat{\mathsf{D}}\tilde{W}) - \mathrm{H}(\mathcal{L}|\mathsf{X}_{\text{drifted}})$ is negligible, since the noise stored in $\mathcal{L}$ is symmetric and hence data-independent.

Note that our scheme has moved the leakage term $\mathrm{I}(\mathsf{X}; \hat{\mathsf{D}})$ from the eavesdropping domain to the domain of the physical memory of the prover device. In particular, an attacker who is able to access the data stored on the prover device,

i.e., list $\mathcal{L}$ and estimator $\hat{\mathsf{D}}$, is able to extract as much information about X, as a passive network adversary, eavesdropping on protocol runs of the original FE protocol scheme.

5.6 — Conclusion

We addressed two leakage issues of the Reverse Fuzzy Extractor protocol. In particular (a) data-dependent short-timescale noise poses a severe security problem, rendering insecure any protocol that uses the Reverse FE in its original form. Furthermore, (b) there is privacy-sensitive leakage if the long-timescale PUF drift depends on the PUF value X.

Our study of experimental data confirms the existence of asymmetric (data-dependent) drift in several types of PUFs.

In this chapter we introduce two modifications to the Reverse FE scheme which together eliminate both leakage problems, (i) additional Z-channel noise that turns a BAC into a BSC. This solves the security problem; and (ii) drift compensation by storing the estimated drift and recent error patterns in the prover device. This solves the privacy problem.

The first modification turns the noisy channel $\mathsf{X} \to \mathsf{X}'$ into an even more noisy channel $\mathsf{X} \to \mathsf{X}''$, compensating its previous asymmetry. The second modification 'moves' the drift problem from eavesdropping to the physical attack scenario, involving an attacker with access to the prover device. In the new scheme, an eavesdropper will not be able to identify the PUF device. Physical access to the prover device's nonvolatile memory in the new scheme will yield as much information as eavesdropping in the original scheme.

Note that the noise parameters (α, β) need to be estimated accurately, otherwise the second Binary Asymmetric Channel will not fully compensate the asymmetry; this results in residual leakage, given by Equation (5.4) with modified parameters. In practice it is difficult to do an entirely accurate estimate. Hence one has to perform a suitable amount of privacy amplification in the `KeyDeriv` operation (Section 5.2.4) in order to eliminate Eve's knowledge.

5.7 — Appendix

5.7.1 – Proof of Lemma 5.1. Since all bits are independent we consider a single bit i. We have $\Pr[\mathsf{E}_\mathsf{i} = 1] = \Pr[\mathsf{E}_\mathsf{i} = 1|\mathsf{X}_\mathsf{i} = 0]\Pr[\mathsf{X}_\mathsf{i} = 0] + \Pr[\mathsf{E}_\mathsf{i} = 1|\mathsf{X}_\mathsf{i} = 1]\Pr[\mathsf{X}_\mathsf{i} = 1] = (1-p)\alpha + p\beta$. Thus, $\mathrm{H}(\mathsf{E}_\mathsf{i}) = \mathrm{h}\Big((1-p)\alpha + p\beta\Big)$. Next we have $\mathrm{H}(\mathsf{E}_\mathsf{i}|\mathsf{X}_\mathsf{i}) = \Pr[\mathsf{X}_\mathsf{i} = 0]\mathrm{h}(\alpha) + \Pr[\mathsf{X}_\mathsf{i} = 1]\mathrm{h}(\beta)$. We obtain $\mathrm{I}(\mathsf{X}_\mathsf{i}; \mathsf{E}_\mathsf{i}) = \mathrm{H}(\mathsf{E}_\mathsf{i}) - \mathrm{H}(\mathsf{X}_\mathsf{i}|\mathsf{E}_\mathsf{i}) = \mathrm{h}\Big((1-p)\alpha + p\beta\Big) - [(1-p)\mathrm{h}(\alpha) + p\mathrm{h}(\beta)]$. Multiplying by the number of bits n gives (5.2). Eq. (5.3) follows from $\mathrm{H}(\mathsf{X}|\mathsf{E}) = \mathrm{H}(\mathsf{X}) - \mathrm{I}(\mathsf{X}; \mathsf{E})$.

5.7.2 – Proof of Theorem 5.2. The bit error probabilities for the $X \to X''$ channel are

$$\begin{aligned} \Pr[X'' = 1|X = 0] &= \alpha(1-\bar{z}) + (1-\alpha)\alpha_z \\ \Pr[X'' = 0|X = 1] &= \beta(1-\alpha_z) + (1-\beta)\bar{z} \end{aligned} \tag{5.14}$$

We want to impose the constraint that these probabilities are equal and then mimimize the bit error rate under this constraint. We use Lagrange multipliers formalism. We introduce the notation $\alpha_z = x^2$ and $\bar{z} = y^2$, thus enforcing $\alpha_z, \bar{z} \geqslant 0$. The Lagrangian for this minimization problem is

$$L(x,y,\lambda) = \alpha(1-y^2)+(1-\alpha)x^2+\lambda\big[\alpha(1-y^2)+(1-\alpha)x^2-\beta(1-x^2)-(1-\beta)y^2\big] \tag{5.15}$$

where λ is the Lagrange constraint multiplier. Note that the quantity to be minimized is the first expression in (5.14); we could equally well have taken the second expression, or some combination. Setting the derivatives of the Lagrangian to zero gives

$$\begin{aligned} \frac{\partial L}{\partial x} &= 2x(1-\alpha) + \lambda\big[2x(1-\alpha) + 2x\beta] = 0 \\ \frac{\partial L}{\partial y} &= -2y\alpha + \lambda\big[-2y\alpha - 2y(1-\beta)] = 0 \\ \frac{\partial L}{\partial \lambda} &= \alpha(1-y^2)+(1-\alpha)x^2-\beta(1-x^2)-(1-\beta)y^2 = 0. \end{aligned} \tag{5.16}$$

The first two lines of (5.16) simplify to

$$\begin{aligned} x = 0 \text{ or } \lambda &= \frac{\alpha-1}{1-\alpha+\beta} \\ y = 0 \text{ or } \lambda &= \frac{-\alpha}{1+\alpha-\beta} \end{aligned} \tag{5.17}$$

This leaves two possible solutions of the whole set of equations,

$$\begin{aligned} \alpha_z &= \frac{\beta-\alpha}{1+\beta-\alpha}, \quad \beta_Z = 0, \qquad \text{if } \beta \geqslant \alpha \\ \bar{z} &= \frac{\alpha-\beta}{1+\alpha-\beta}, \quad \alpha_Z = 0, \qquad \text{if } \alpha \geqslant \beta \end{aligned}$$

Substituting α_z and $\bar{z}$ into (5.14) yields ε.

5.7.3 – Proof of Lemma 5.5. In bit i we have the following conditional probabilities,

$$\text{Prob}[D_{1,i} \neq D_{2,i}|X_1 = x_1, X_2 = x_2] = \begin{cases} 2ad(1-ad) \text{ if } x_{1,i} = x_{2,i} = 0 \\ 2\beta_d(1-\beta_d) \text{ if } x_{1,i} = x_{2,i} = 1 \\ ad(1-\beta_d) + (1-ad)\beta_d \text{ if } x_{1,i} \neq x_{2,i} \end{cases} \tag{5.18}$$

We compute $\mathsf{P}_{\text{uneq}} \stackrel{\text{def}}{=} \mathsf{Prob}[\mathsf{D}_{1,\mathfrak{i}} \neq \mathsf{D}_{2,\mathfrak{i}}] = \mathbb{E}_{\mathfrak{x}_1\mathfrak{x}_2}\mathsf{Prob}[\mathsf{D}_{1,\mathfrak{i}} \neq \mathsf{D}_{2,\mathfrak{i}}|\mathsf{X}_1 = \mathfrak{x}_1, \mathsf{X}_2 = \mathfrak{x}_2] = \frac{1}{4}\sum_{\mathfrak{x}_{1,\mathfrak{i}}\mathfrak{x}_{2,\mathfrak{i}}} \mathsf{Prob}[\mathsf{D}_{1,\mathfrak{i}} \neq \mathsf{D}_{2,\mathfrak{i}}|\mathsf{X}_1 = \mathfrak{x}_1, \mathsf{X}_2 = \mathfrak{x}_2]$. Performing the summation and then simplifying the result yields (5.9). The drift in each bit position is independent; therefore the Hamming weight is the result of $\mathfrak{n}$ independent events, each of which increments the Hamming weight with probability P_{uneq}.

5.7.4 – Proof of Theorem 5.7.

$$\begin{aligned}
\mathsf{I}(\mathcal{E};\mathsf{D}) &= \mathsf{H}(\mathcal{E}) - \mathsf{H}(\mathcal{E}|\mathsf{D}) && (5.19)\\
&= \mathsf{H}(\mathsf{D}+\mathcal{N}) - \mathsf{H}(\mathcal{N}) && (5.20)\\
&= \mathsf{H}(\mathsf{D}+\mathcal{N}_{\text{av}}, \mathsf{D}+\mathcal{N}) - \mathsf{H}(\mathcal{N}_{\text{av}}, \mathcal{N}) && (5.21)\\
&= \mathsf{H}(\mathsf{D}+\mathcal{N}_{\text{av}}) + \mathsf{H}(\mathsf{D}+\mathcal{N}|\mathsf{D}+\mathcal{N}_{\text{av}}) - [\mathsf{H}(\mathcal{N}_{\text{av}}) + \mathsf{H}(\mathcal{N}|\mathcal{N}_{\text{av}})] && (5.22)\\
&= \mathsf{H}(\mathsf{D}+\mathcal{N}_{\text{av}}) + \mathsf{H}(\mathcal{N}|\mathcal{N}_{\text{av}}) - [\mathsf{H}(\mathcal{N}_{\text{av}}) + \mathsf{H}(\mathcal{N}|\mathcal{N}_{\text{av}})] && (5.23)\\
&= \mathsf{H}(\mathsf{D}\oplus\mathcal{N}_{\text{av}}) - \mathsf{H}(\mathcal{N}_{\text{av}}). && (5.24)
\end{aligned}$$

5.7.5 – Proof of Theorem 5.8. We have

$$\begin{aligned}
\mathsf{H}(\mathsf{X}|\mathcal{E}) &\geqslant \mathsf{H}(\mathsf{X}|\mathsf{D}\mathcal{N}) && (5.25)\\
&= \mathsf{H}(\mathsf{X}|\mathsf{D}) + \mathsf{H}(\mathcal{N}|\mathsf{X}\mathsf{D}) - \mathsf{H}(\mathcal{N}|\mathsf{D}) && (5.26)\\
&= \mathsf{H}(\mathsf{X}|\mathsf{D}) + \mathsf{H}(\mathcal{N}|\mathsf{X}_{\text{drifted}}) - \mathsf{H}(\mathcal{N}|\mathsf{D}) && (5.27)\\
&\geqslant \mathsf{H}(\mathsf{X}|\mathsf{D}) + \mathsf{H}(\mathcal{N}|\mathsf{X}_{\text{drifted}}) - \mathsf{H}(\mathcal{N}) && (5.28)\\
&= \mathsf{H}(\mathsf{X}|\mathsf{D}) - \mathsf{I}(\mathcal{N};\mathsf{X}_{\text{drifted}}). && (5.29)
\end{aligned}$$

In (5.25) we used that D and $\mathcal{N}$ together contain more information than $\mathcal{E}$. In (5.27) we used that $\mathcal{N}$ depends on X and D only through $\mathsf{X}\oplus\mathsf{D}$. Finally we take $\mathsf{H}(\mathsf{X})$ minus the whole inequality (5.25,5.29).

5.7.6 – Proof of Theorem 5.9.

$$\begin{aligned}
\mathsf{H}(\mathsf{X}|W\mathcal{E}) &= \mathsf{H}(\mathsf{X}|W) + \mathsf{H}(\mathcal{E}|\mathsf{X}W) - \mathsf{H}(\mathcal{E}|W)\\
&= \mathsf{H}(\mathsf{X}|W) + \mathsf{H}(\mathcal{E}|\mathsf{X}) - \mathsf{H}(\mathcal{E}|W) && (5.30)\\
&\geqslant \mathsf{H}(\mathsf{X}|W) + \mathsf{H}(\mathcal{E}|\mathsf{X}) - \mathsf{H}(\mathcal{E}) && (5.31)\\
&= \mathsf{H}(\mathsf{X}|W) - \mathsf{I}(\mathcal{E};\mathsf{X}). && (5.32)
\end{aligned}$$

In (5.30) we used the fact that W is a function of X. Finally we take $\mathsf{H}(\mathsf{X})$ minus the whole inequality derived above.

5.7.7 – Proof of Theorem 5.10. We write

$$\mathsf{H}(\mathsf{X}|\tilde{W}\hat{\mathsf{D}}\mathcal{L}) = \mathsf{H}(\mathsf{X}\tilde{W}\hat{\mathsf{D}}\mathcal{L}) - \mathsf{H}(\tilde{W}\hat{\mathsf{D}}\mathcal{L}). \tag{5.33}$$

Applying the chain rule again we expand these terms as

$$\begin{aligned} \mathsf{H}(\mathsf{X}\tilde{W}\hat{\mathsf{D}}\mathcal{L}) &= \mathsf{H}(\mathsf{X}) + \mathsf{H}(\tilde{W}\hat{\mathsf{D}}\mathcal{L}|\mathsf{X}) = \\ &= \mathsf{H}(\mathsf{X}|W) + \mathsf{H}(W) + \mathsf{H}(\hat{\mathsf{D}}|\mathsf{X}) \\ &+ \underbrace{\mathsf{H}(\tilde{W}|\hat{\mathsf{D}}\mathsf{X})}_{0} + \underbrace{\mathsf{H}(\mathcal{L}|\tilde{W}\hat{\mathsf{D}}\mathsf{X})}_{\mathsf{H}(\mathcal{L}|\hat{\mathsf{X}}_{\text{drifted}})} \end{aligned} \tag{5.34}$$

and

$$\mathsf{H}(\tilde{W}\hat{\mathsf{D}}\mathcal{L}) = \mathsf{H}(\hat{\mathsf{D}}) + \mathsf{H}(\tilde{W}|\hat{\mathsf{D}}) + \mathsf{H}(\mathcal{L}|\hat{\mathsf{D}}\tilde{W}). \tag{5.35}$$

In (5.34) we have used the fact that $\mathcal{L}$ is noise on $\mathsf{X}_{\text{drifted}}$ and therefore can depend at most on $\mathsf{X}_{\text{drifted}}$ itself. We substitute (5.34) and (5.35) into (5.33).

Chapter 6

Conclusions

6.1 — Enhancing Helper Data Systems

This thesis deals with two important topics in security: privacy-preserving use of biometrics, and key storage in the form of Physically Obfuscated Keys. Biometric authentication and POKs have found their way into consumer electronics devices such as smartphones. The large scale deployment of these technologies makes it important to have good understanding and control of all their privacy and security properties. The main issue is how to reconcile the requirement of error resilience with the privacy/security requirement. A central role is played here by Helper Data Systems, special error-correcting schemes that are designed to leak as little information as possible through their redundancy data. HDS algorithms not only have to be secure but must also run efficiently. After more than a decade of HDS research many problems have been solved. However, as mentioned in Chapter 1, at the beginning of the PhD project several issues were unresolved regarding the optimization and deployability of highly efficient HDS constructions:
(i) Can we maximize the entropy extracted by a Zero Leakage Helper Data System quantizer for a given source distribution and noise level?
(ii) Can we construct a high-performance HDS based on fingerprint minutiae? Here high performance means high accuracy of the matching decision as well as fast processing.
(iii) Is it feasible to use the Reverse Fuzzy Extractor trick when the noise is data dependent and the POK has drift?
The contributions in this thesis improve the state of the art in a number of ways, in terms of theoretical analysis as well as scheme construction:
(i) optimization of the Zero Leakage HDS quantization;
(ii) new fixed-length representations of fingerprint minutia lists, and a complete two-stage HDS construction from it;

(iii) a solution for the leakage problem that occurs when the error correction is outsourced and the bit errors are data-dependent.

These improvements make it easier to implement good privacy protection for stored biometric data, and to use the highly efficient error-correction outsourcing trick. The latter is relevant for Physically Obfuscated Keys as well as biometric authentication on constrained devices such as smartcards.

6.2 — Summary of results

To apply helper data systems (HDS) to biometrics and Physical Unclonnable functions (PUF) numerous challenges need to be addressed.

The first challenge is low entropy of fingerprints. A low-entropy secret protected by a one-way function can be guessed by a brute force attack, which is unacceptable in security applications.

The second challenge is high bit error rate. The 2nd stage HDS needs a good ECC which is capable of handling a high BER while still having a good code rate, even for short codewords.

The third challenge is appearance/disappearance of minutiae. The number of minutiae may be different on every image capture while an error correction code requires input to have a fixed-length.

The fourth challenge is recognition performance degradation caused by template protection. Securing the template requires an extra processing step, which results in an extra information loss and lower recognition performance.

The fifth challenge is data dependent noise while outsourcing the error correction to an external party. The revealed error pattern leaks information about the secret data. With each repetition of the protocol, different data may leak.

The sixth challenge is PUF drift. The drift makes PUF recognizable when outsourcing of the error correction is used. This has an impact on privacy. Additionally the reconstruction will fail after certain amount of drift.

For the first stage HDS we optimized the quantization boundaries and derived the optimal reconstruction boundaries. Our results allow to extract more information in comparison to the state-of-the-art and significantly reduce the bit error rate.

For fingerprints we introduced a new spectral function. The approach allows to obtain a fixed-length representation on a small grid which is faster than the state-of-the-art while keeping the same recognition performance. Additionally the matching performance does not degrade much under image rotation.

We built a complete privacy-preserving fingerprint template protection scheme based on a two stage HDS. When the biometric has high quality, the transition from the analog unprotected spectral function to fully protected can be done with almost no performance penalty. The realization of the scheme was enabled by Polar codes which correct high bit error rate in short codewords.

For the data dependent noise problem we modified the protocol between a resource constrained device and strong verifier to eliminate the possible leakage. The main technique is to introduce extra binary asymmetric noise in the communication channel and make the noise data independent. This modification makes the protocol secure while remaining practical; it reduces the channel capacity by an acceptable amount.
For the PUF drift problem we introduced a separate buffer that stores recent error patterns and the estimated drift of the PUF. If the error pattern changes in time, the stored helper data is modified to compensate the drift.

6.3 — Directions for future research

In Chapter 2, we optimized the first-stage ZLHDS quantization levels for a given source and noise distribution. Our work improves the state-of-the-art for an intermediate level of noise. For very low or large level of noise our results converge to the previous work (de Groot et al. [19]). It must be noted that the model that we use assumes that one has the perfect knowledge about the distributions (noise and the original data). A mismatch between the actual source distribution and the distribution used to build the HDS may cause leakage about the secret. The amount of leakage should be further investigated.
In our derivation of optimal reconstruction boundaries, we considered only two noise distributions. For other distributions the optimal reconstruction boundaries can be derived by using Lemma 2.4.
Even though the scheme is called zero leakage, it is inevitable that the helper data leaks about the enrolled value. A ZLHDS may still reveal more about the source than what is considered acceptable in terms of privacy. For instance, the first-stage helper data w in the continuum limit can reveal whether a source value $x \in \mathbb{R}$ has large absolute value: $w \approx 0$ implies a large probability that x lies in the left tail of the distribution, and analogously $w \approx 1$ for the right tail. This kind of leakage might be problematic for some components of the enrollment vector. (Note that the leakage about the absolute value is vastly reduced at small subdivision parameter M.) It would be very interesting to see how the ZLHDS approach compares to the "sparse coding with ambiguization" approach (Section 1) in this respect.
In Chapter 3 and 4, we proposed methods to transform fingerprint data (i.e., location and orientations of minutiae) to the fixed-length representation and combined it with the ZLHDS to obtain two-stage HDS. Two-stage HDS provides security of the template, however the recognition performance is much lower than for standard methods for unprotected matching. This happens due to the fact the that fixed-length representation approach extracts less information than the set of minutiae.
Our scheme extracts little entropy from a finger, even when we combine multiple enrollment images. The scheme can be further improved by trying $N = 3$ or even

$N = 4$ for grid points that have a good signal-to-noise ratio. An alternative idea is to apply a four-dimensional grid based on R_{ab}, ϕ_{ab}, β_a, and β_b instead of a two-dimensional grid. In all the existing spectral functions, the information from the four important minutia-pair variables $R_{ab}, \phi_{ab}, \beta_a, \beta_b$ gets mixed together. Building a fixed-length representation on a four-dimensional grid based on these variables may result in less information loss. It would be interesting to see if this idea leads to an improved matching performance, and how much computational overhead would be caused by the increased dimension of the grid.

The resuts of Section 4 show that template protection can be done while only slightly reducing the recognition performance. Does it hold *in general* that ZLHDS causes little performance loss?

In Chapter 5, we eliminated the leakage in the protocol between a resource constrained device and a strong verifier by turning asymmetric noise into symmetric. The parameters of the asymmetric noise may be difficult to estimate accurately. If the HDS is built based on a wrongly estimated noise model, some leakage still remains. This was not discussed in details in Chapter 5 and remains as a future work.

Bibliography

[1] http://bias.csr.unibo.it/fvc2000/participants/results/NEUR_db2_a.asp.

[2] http://bias.csr.unibo.it/fvc2002/results/res_db2_a.asp.

[3] VeriFinger SDK. Available online, www.neurotechnology.com.

[4] Zynq ultrascale+ device. technical reference manual. https://http://www.xilinx.com/support/documentation/user_guides/ug1085-zynq-ultrascale-trm.pdf. Accessed: 2019-04-08.

[5] E. Arıkan. Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, 2009.

[6] Aydin Aysu, Ege Gulcan, Daisuke Moriyama, Patrick Schaumont, and Moti Yung. End-to-end design of a PUF-based privacy preserving authentication protocol. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 556–576. Springer, 2015.

[7] C.H. Bennett, G. Brassard, C. Crépeau, and M. Skubiszewska. Practical quantum oblivious transfer. In *CRYPTO*, pages 351–366, 1991.

[8] Charles H Bennett, Gilles Brassard, Claude Crépeau, and Ueli M Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6):1915–1923, 1995.

[9] C. Böhm and M. Hofer. *Physical Unclonable Functions in Theory and Practice*. Springer, 2013.

[10] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 82–91. ACM, 2004.

[11] J. Bringer, V. Despiegel, and M. Favre. Adding localization information in a fingerprint binary feature vector representation. In *Proc. SPIE 8029, Sensing Technologies for Global Health, Military Medicine, Disaster Response, and Environmental Monitoring; and Biometric Technology for Human Identification VIII*, page 80291O, 2011.

[12] James DR Buchanan, Russell P Cowburn, Ana-Vanessa Jausovec, Dorothee Petit, Peter Seem, Gang Xiong, Del Atkinson, Kate Fenton, Dan A Allwood, and Matthew T Bryan. Forgery:'fingerprinting'documents and packaging. *Nature*, 436(7050):475, 2005.

[13] R. Canetti, B. Fuller, O. Paneth, L. Reyzin, and A. Smith. Reusable fuzzy extractors for low-entropy distributions. In *Eurocrypt 2016*, 2016.

[14] Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam Smith. Reusable fuzzy extractors for low-entropy distributions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 117–146. Springer, 2016.

[15] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[16] B. Chen and F.M.J. Willems. Secret key generation over biased physical unclonable functions with polar codes. *IEEE Internet of Things Journal*, 6(1):435–445, 2019.

[17] Mathias Claes, Vincent van der Leest, and An Braeken. Comparison of SRAM and FF PUF in 65nm technology. In *Nordic Conference on Secure IT Systems*, pages 47–64, 2011.

[18] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[19] J. de Groot, B. Škorić, N. de Vreede, and J.P. Linnartz. Quantization in Zero Leakage Helper Data Schemes. *EURASIP Journal on Advances in Signal Processing*, 2016. 2016:54.

[20] Gerald DeJean and Darko Kirovski. Rf-dna: Radio-frequency certificates of authenticity. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 346–363. Springer, 2007.

[21] Jeroen Delvaux, Dawu Gu, Ingrid Verbauwhede, Matthias Hiller, and Meng-Day Mandel Yu. Efficient fuzzy extraction of PUF-induced secrets: Theory and applications. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 412–431. Springer, 2016.

[22] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy Extractors: how to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[23] Y. Dodis, K. Pietrzak, and D. Wichs. Key derivation without entropy waste. In *EUROCRYPT 2014*, LNCS, pages 93–110. Springer.

[24] Y. Dodis, M. Reyzin, and A. Smith. Fuzzy Extractors: How to generate strong keys from biometrics and other noisy data. In *Eurocrypt 2004*, volume 3027 of *LNCS*, pages 523–540. Springer-Verlag, 2004.

[25] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

[26] F. Farooq, R.M. Bolle, T.-Y. Jea, and N. Ratha. Anonymous and revocable fingerprint recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2007.

[27] Faisal Farooq, Ruud M Bolle, Tsai-Yang Jea, and Nalini Ratha. Anonymous and revocable fingerprint recognition. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–7. IEEE, 2007.

[28] D. Frumkin, A. Wasserstrom, A. Davidson, and A. Grafit. Authentication of forensic DNA samples. *FSI Genetics*, 4(2):95–103, 2010.

[29] B. Gassend. Physical Random Functions. Master's thesis, Massachusetts Institute of Technology, 2003.

[30] Blaise Gassend, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. Silicon physical random functions. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 148–160. ACM, 2002.

[31] Gerben Geltink. Concealing Ketje: A Lightweight PUF-Based Privacy Preserving Authentication Protocol. In *International Workshop on Lightweight Cryptography for Security and Privacy*, pages 128–148. Springer, 2016.

[32] S Golomb. The limiting behavior of the Z-channel (Corresp.). *IEEE Transactions on Information Theory*, 26(3):372–372, 1980.

[33] J. Guajardo, S.S. Kumar, G.-J. Schrijen, and P. Tuyls. FPGA intrinsic PUFs and their use for IP protection. In *Cryptographic Hardware and Embedded Systems (CHES) 2007*, volume 4727 of *LNCS*, pages 63–80. Springer.

[34] Jorge Guajardo, Sandeep S Kumar, Geert-Jan Schrijen, and Pim Tuyls. Fpga intrinsic pufs and their use for ip protection. In *International workshop on cryptographic hardware and embedded systems*, pages 63–80. Springer, 2007.

[35] Jorge Guajardo, Sandeep S. Kumar, Geert Jan Schrijen, and Pim Tuyls. FPGA Intrinsic PUFs and Their Use for IP Protection. In *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems*, pages 63–80, 2007.

[36] Jorge Guajardo, Boris Škorić, Pim Tuyls, Sandeep S Kumar, Thijs Bel, Antoon HM Blom, and Geert-Jan Schrijen. Anti-counterfeiting, key distribution, and key storage in an ambient world via physical unclonable functions. *Information Systems Frontiers*, 11(1):19–41, 2009.

[37] Ghaith Hammouri, Aykutlu Dana, and Berk Sunar. Cds have fingerprints too. In *Cryptographic Hardware and Embedded Systems-CHES 2009*, pages 348–362. Springer, 2009.

[38] Ryan Helinski, Dhruva Acharyya, and Jim Plusquellic. A physical unclonable function defined using power distribution system equivalent resistance variations. In *2009 46th ACM/IEEE Design Automation Conference*, pages 676–681. IEEE, 2009.

[39] Anthony Herrewege, Stefan Katzenbeisser, Roel Maes, Roel Peeters, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. Reverse Fuzzy Extractors: Enabling Lightweight Mutual Authentication for PUF-Enabled RFIDs. In *Financial Cryptography and Data Security*, pages 374–389. 2012.

[40] Tanya Ignatenko and Frans MJ Willems. Information leakage in fuzzy commitment schemes. *IEEE Transactions on Information Forensics and Security*, 5(2):337–348, 2010.

[41] Ronald S Indeck and Marcel W Muller. Method and apparatus for fingerprinting magnetic media, November 15 1994. US Patent 5,365,586.

[42] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

[43] Z. Jin, M.H. Lim, A.B.J. Teoh, B.M. Goi, and Y.H. Tay. Generating fixed-length representation from minutiae using kernel methods for fingerprint authentication. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(10):1415 – 1428, 2016.

[44] Z. Jin, A.B.J. Teoh, T.S. Ong, and C. Tee. Generating revocable fingerprint template using minutiae pair representation. In *International Conference on Education Technology and Computer*, pages 251–255. IEEE, 2010.

[45] Zhe Jin, Andrew Beng Jin Teoh, Thian Song Ong, and Connie Tee. A revocable fingerprint template for security and privacy preserving. *KSII Transactions on Internet & Information Systems*, 4(6), 2010.

[46] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[47] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security (CCS) 1999*, pages 28–36, 1999.

[48] J.-P. Kaps, K. Yüksel, and B. Sunar. Energy scalable universal hashing. *IEEE Trans. Computers*, 54(12):1484–1495, 2005.

[49] Stefan Katzenbeisser, Ünal Kocabaş, Vladimir Rožić, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. PUFs: Myth, fact or busted? A security evaluation of physically unclonable functions (PUFs) cast in silicon. In *Cryptographic Hardware and Embedded Systems–CHES 2012*, pages 283–301. Springer, 2012.

[50] Christoph Keller, Frank Gürkaynak, Hubert Kaeslin, and Norbert Felber. Dynamic memory-based physically unclonable function for the generation of unique identifiers and true random numbers. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2740–2743. IEEE, 2014.

[51] Patrick Koeberl, Jiangtao Li, Anand Rajan, and Wei Wu. Entropy loss in PUF-based key generation schemes: The repetition code pitfall. In *Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on*, pages 44–49. IEEE, 2014.

[52] Sandeep S Kumar, Jorge Guajardo, Roel Maes, Geert-Jan Schrijen, and Pim Tuyls. The butterfly puf protecting ip on every fpga. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 67–70. IEEE, 2008.

[53] Daihyun Lim, Jae W Lee, Blaise Gassend, G Edward Suh, Marten Van Dijk, and Srinivas Devadas. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(10):1200–1205, 2005.

[54] J.-P. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *Audio- and Video-Based Biometric Person Authentication*. Springer, 2003.

[55] Keith Lofstrom, W Robert Daasch, and Donald Taylor. Ic identification circuit using device mismatch. In *2000 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No. 00CH37056)*, pages 372–373. IEEE, 2000.

[56] R. Maes. *Physically Unclonable Functions: Constructions, Properties and Applications*. Springer, 2013.

[57] Roel Maes, Pim Tuyls, and Ingrid Verbauwhede. Intrinsic pufs from flip-flops on reconfigurable devices. In *3rd Benelux workshop on information and system security (WISSec 2008)*, volume 17, page 2008, 2008.

[58] Roel Maes, Vincent van der Leest, Erik van der Sluis, and Frans Willems. Secure key generation from biased PUFs. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 517–534. Springer, 2015.

[59] Roel Maes and Ingrid Verbauwhede. Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions. In *Towards Hardware-Intrinsic Security*, Information Security and Cryptography, pages 3–37. 2010.

[60] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer, London, 2nd edition, 2009.

[61] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino. Impact of artificial "gummy" fingers on fingerprint systems. In *Proc. SPIE, Optical Security and Counterfeit Deterrence Techniques IV*, volume 4677, pages 275–289, 2002.

[62] S.M. Moser, P.-N. Chen, and H.-Y. Lin. Error Probability Analysis of Binary Asymmetric Channels, 2010.

[63] K. Nandakumar. A fingerprint cryptosystem based on minutiae phase spectrum. In *Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2010.

[64] Karthik Nandakumar. A fingerprint cryptosystem based on minutiae phase spectrum. In *2010 IEEE International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2010.

[65] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez, V. Espinosa, A. Satue, I. Hernaez, J.J. Igarza, C. Vivaracho, D. Escudero, and Q.I. Moro. MCYT baseline corpus: A bimodal biometric database. In *Vision, Image and Signal Processing, Special Issue on Biometrics on the Internet*, volume 150, pages 395–401. IEEE, 2003.

[66] RS Pappu. Physical one-way functions [ph. d. thesis]. *Massachusetts Institute of Technology, Cambridge, Mass, USA*, 2001.

[67] Jaishanker K Pillai, Vishal M Patel, Rama Chellappa, and Nalini K Ratha. Sectored random projections for cancelable iris biometrics. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1838–1841. IEEE, 2010.

[68] Jaishanker K Pillai, Vishal M Patel, Rama Chellappa, and Nalini K Ratha. Secure and robust iris recognition using random projections and sparse representations. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1877–1893, 2011.

[69] Lorenzo Pupillo, Afonso Ferreira, and Gianluca Varisco. Software vulnerability disclosure in europe: Technology, policies and legal challenges. report of a ceps task force. ceps task force reports 28 june 2018. 2018.

[70] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40:614–634, 2001.

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

[72] Behrooz Razeghi, Slava Voloshynovskiy, Dimche Kostadinov, and Olga Taran. Privacy preserving identification using sparse approximation with ambiguization. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2017.

[73] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[74] Ulrich Rührmair, Heike Busch, and Stefan Katzenbeisser. Strong PUFs: Models, Constructions, and Security Proofs. In *Towards Hardware-Intrinsic Security*, pages 79–96. 2010.

[75] A.-R. Sadeghi and D. Naccache, editors. *Towards hardware-intrinsic security*. Springer, 2010.

[76] André Schaller, Boris Škorić, and Stefan Katzenbeisser. On the systematic drift of physically unclonable functions due to aging. In *Proceedings of the 5th International Workshop on Trustworthy Embedded Devices*, pages 15–20, 2015.

[77] André Schaller, Taras Stanko, Boris Škorić, and Stefan Katzenbeisser. Eliminating leakage in reverse fuzzy extractors. *IEEE Transactions on Information Forensics and Security*, 13(4):954–964, 2018.

[78] Steffen Schulz, Ahmad-Reza Sadeghi, and Christian Wachsmann. Short Paper: Lightweight Remote Attestation Using Physical Functions. In *Proceedings of the Fourth ACM Conference on Wireless Network Security*, pages 109–114, 2011.

[79] X. Shao and R.N.J. Veldhuis. A real helper data scheme. In *IAPR Asian Conference on Pattern Recognition*, pages 84–89. IEEE, 2013.

[80] Boris Škorić. A trivial debiasing scheme for helper data systems. 2016.

[81] T. Stanko, F.N. Andini, and B. Škorić. Optimized quantization in Zero Leakage Helper Data Systems. *IEEE Transactions on Information Forensics and Security*, 12(8):1957–1966, 2017.

[82] T. Stanko and B. Škorić. Minutia-pair spectral representations for fingerprint template protection. In *IEEE Workshop on Information Forensics and Security (WIFS)*. IEEE, 2017.

[83] T. Stanko and B. Škorić. Fingerprint template protection using minutia-pair spectral representations, 2018. `https://arxiv.org/pdf/1804.01744`.

[84] D.R. Stinson. Universal hashing and authentication codes. *Designs, Codes, and Cryptography*, 4:369–380, 1994.

[85] Ying Su, Jeremy Holleman, and Brian Otis. A 1.6 pj/bit 96% stable chip-id generating circuit using process variations. In *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, pages 406–611. IEEE, 2007.

[86] B. Topcu, H. Erdogan, C. Karabat, and B. Yanıkoglu. Biohashing with fingerprint spectral minutiae. In *Int. Conf. of the Biometrics Special Interest Group (BIOSIG)*, pages 254–265. IEEE, 2013.

[87] B. Topcu, Y.Z. Isik, and H. Erdogan. GMM-SVM fingerprint verification based on minutiae only. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, pages 155–160. IEEE, 2016.

[88] P. Tuyls, A.H.M. Akkermans, T.A.M. Kevenaar, G.-J. Schrijen, A.M. Bazen, and R.N.J. Veldhuis. Practical biometric authentication with template protection. In *International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, pages 436–446. Springer-Verlag, 2005.

[89] P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, R. Verhaegh, and R. Wolters. Read-proof hardware from protective coatings. In *Cryptographic Hardware and Embedded Systems (CHES) 2006*, volume 4249 of *LNCS*, pages 369–383. Springer-Verlag, 2006.

[90] P. Tuyls, B. Škorić, and T. Kevenaar. *Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting*. Springer, London, 2007.

[91] Pim Tuyls, Anton HM Akkermans, Tom AM Kevenaar, Geert-Jan Schrijen, Asker M Bazen, and Raimond NJ Veldhuis. Practical biometric authentication with template protection. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 436–446. Springer, 2005.

[92] Pim Tuyls, Geert-Jan Schrijen, Boris Škorić, Jan Van Geloven, Nynke Verhaegh, and Rob Wolters. Read-proof hardware from protective coatings. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 369–383. Springer, 2006.

[93] Pim Tuyls, Geert-Jan Schrijen, Frans Willems, Tanya Ignatenko, and Boris Škorić. Secure key storage with PUFs. *Security with Noisy Data-On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*, pages 269–292, 2007.

[94] Robbert van den Berg, Boris Škorić, and Vincent van der Leest. Bias-based Modeling and Entropy Analysis of PUFs. In *Proceedings of the 3rd International Workshop on Trustworthy Embedded Devices*, pages 13–20, 2013.

[95] Raymond NJ Veldhuis. The relation between the secrecy rate of biometric template protection and biometric recognition performance. In *2015 International Conference on Biometrics (ICB)*, pages 311–318. IEEE, 2015.

[96] E.A. Verbitskiy, P. Tuyls, C. Obi, B. Schoenmakers, and B. Škorić. Key extraction from general nondiscrete signals. *IEEE Transactions on Information Forensics and Security*, 5(2):269–279, 2010.

[97] Evgeny A Verbitskiy, Pim Tuyls, Chibuzo Obi, Berry Schoenmakers, and Boris Skoric. Key extraction from general nondiscrete signals. *IEEE Transactions on Information Forensics and Security*, 5(2):269–279, 2010.

[98] Serge Vrijaldenhoven. Acoustical physical uncloneable functions. *Philips internal publication PR-TN-2004-300300*, 2005.

[99] B. Škorić and N. de Vreede. The Spammed Code Offset Method. *IEEE Transactions on Information Forensics and Security*, 9(5):875–884, May 2014.

[100] B. Škorić and N. de Vreede. The Spammed Code Offset Method. *IEEE Transactions on Information Forensics and Security*, 9(5):875–884, 2014.

[101] C.I. Watson, M.D. Garris, E. Tabassi, C.L. Wilson, R.M. McCabe, S. Janet, and K. Ko. User's guide to export controlled distribution of NIST biometric image software, 2004. NISTIR 7391.

[102] H. Xu and R.N.J. Veldhuis. Spectral minutiae representations of fingerprints enhanced by quality data. In *Int. Conf. on Biometrics: Theory, Applications and Systems (BTAS) 2009*. IEEE, 2009.

[103] H. Xu and R.N.J. Veldhuis. Spectral representations of fingerprint minutiae subsets. In *Image and Signal Processing (CISP) 2009*, pages 1–5, 2009.

[104] H. Xu and R.N.J. Veldhuis. Complex spectral minutiae representation for fingerprint recognition. In *Computer Vision and Pattern Recognition Workshop*. IEEE, 2010.

[105] H. Xu, R.N.J. Veldhuis, A.M. Bazen, T.A.M. Kevenaar, A.H.M. Akkermans, and B. Gokberk. Fingerprint verification using spectral minutiae representations. *IEEE Transactions on Information Forensics and Security*, 4(3):397–409, 2009.

[106] Haiyun Xu and Raymond NJ Veldhuis. Complex spectral minutiae representation for fingerprint recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 1–8. IEEE, 2010.

[107] Shuiming Ye, Ying Luo, Jian Zhao, and Sen-ChingS Cheung. Anonymous biometric access control. *EURASIP Journal on Information Security*, 2009(1):865259, 2009.

Appendix A

Summary

Enhancing the performance and security of Helper Data Systems

This thesis deals with two important topics in security: privacy-preserving use of biometrics, and key storage by using Physical Unclonable Functions (PUF) in the form of Physically Obfuscated Keys (POK). Biometric authentication and POKs have found their way into consumer electronics devices such as smartphones. The large scale deployment of these technologies makes it important to have a good understanding and control of all their privacy and security properties. The main issue is how to reconcile the requirement of error resilience with the privacy/security requirement. A central role is played here by Helper Data Systems (HDS), special error-correcting schemes that are designed to leak as little information as possible through their redundancy data. HDS algorithms not only have to be secure but must also run efficiently. After more than a decade of HDS research many problems have been solved. However, at the beginning of the PhD project several issues were unresolved regarding the optimization and deployability of highly efficient HDS constructions. In this thesis we concentrate on the application of HDS for fingerprints and efficient HDS implementation in general.

We have introduced a new representation of fingerprint images. This representation yields similar recognition performance as state-of-the art template protection schemes, while being faster. We optimized the first step of extraction information extraction algorithm (for biometrics as well as POKs), which significantly reduces the bit error rate.

A HDS may create a bottleneck at the error correction decoding step. The error correction can be outsourced to a more powerful second party. However, an eavesdropper then learns the error pattern, which leads to security issues if the noise is data-dependent. Additionally experiments have shown that some PUFs are prone to drift (shift of the PUF properties from which the secret key is

generated). Thus, the PUFs become recognizable when the outsourcing is used. This has a potential impact on privacy. We have introduced an approach that eliminates leakage and compensates the drift. Thus, we ensure that outsourced error correction can be done in a secure and privacy preserving way.

We have built a privacy-preserving template protection scheme for fingerprints based on our new representation of fingerprints and the optimized information extraction algorithm. The best results were obtained by combining three enrollment images. The performance degradation due to added privacy is marginal for good quality fingerprints.

Our contributions make it more practical to implement good privacy protection for stored biometric data, and to use the highly efficient error-correction outsourcing trick for deployment of HDSs on resource-constrained devices such as smartcards.

Appendix B

Curriculum Vitae

Taras Stanko was born on July 14th, 1989 Boryslav, Ukraine. He studied Physics at Ivan Franko National University of Lviv, Ukraine. He completed his bachelor's and master's studies in respectively 2009 and 2011.
In 2015 he obtained a PDEng degree at the Mathematics for Industry program from Eindhoven University of Technology.
In 2015, he started his Ph.D. in the Security group at the Eindhoven University of Technology, under the supervision of Dr. Boris Škorić and Prof. Sandro Etalle. In his Ph.D. he worked in the ESPRESSO (Efficient and Strong template PRotection by Enabling Secure Sketch On-card) project, supported by NWO (Nederlandse Organisatie voor Wetenschappelijk Onderzoek). This project aims to provide strong template protection of fingerprints by enabling Helper Data System on a smart card. In this context he developed a technique to optimize the Helper Data System, improve privacy-preserving fingerprint matching, and improve the security of error correction outsourcing for smart cards. The present dissertation contains the results of his work from 2015 to 2019.

Appendix C

Publications

Publications on which this thesis is based

- T. Stanko, B. Chen and B. Škorić. Fingerprint template protection using minutia-pair spectral representations. In: EURASIP Journal on Information Security, Vol. 2019, Nr. 1, 12, 2019.

- A. Schaller, T. Stanko, B. Škorić and S. Katzenbeisser. Eliminating leakage in reverse fuzzy extractors. In *IEEE Transactions on Information Forensics and Security.* 13, 4, pages 954-964, 2018.

- T. Stanko, F. N. Andini and B. Škorić. Optimized quantization in Zero Leakage Helper data systems. In *IEEE Transactions on Information Forensics and Security.* 12, 8, pages 1957-1966, Aug 2017.

- T. Stanko, and B. Škorić. Minutia-pair spectral representations for fingerprint template protection. In *IEEE Workshop on Information Forensics and Security (WIFS).* Rennes, France 4-7 December 2017. Piscataway: Institute of Electrical and Electronics Engineers (IEEE) pages 1-6, 2017.

Other publications

- B. Razeghi, T. Stanko, B. Škorić and S. Voloshynovskiy. Single-Component Privacy Guarantees in Helper Data Systems and Sparse Coding with Ambiguation. In *IEEE Workshop on Information Forensics and Security (WIFS).* To appear in proceedings.