Security issues in helper data systems

Obi, C.O.

Award date:
2008

Security Issues in Helper Data Systems

Chibuzo Obianuju Obi
Masters Thesis
Eindhoven University of Technology
Department of Mathematics and Computer Science

Supervisors:
L.A.M. Schoenmakers and B. Škorić
Eindhoven University of Technology
Department of Mathematics and Computer Science

Eindhoven, August 18, 2008

Contents

List of Figures

List of Tables

Acknowledgements

It is a pleasure to thank the many people who made this thesis possible.

My most sincere appreciation goes to my supervisors Boris Škorić and Berry Schoenmakers for their supervision and guidance. Many thanks to Evgeny Verbitskiy, Pim Tuyls, Tom Kevenaar, Jorge Guajardo and Emile Kelkboom for their contributions to this thesis.

Most importantly, I am immensely grateful to God for the grace of successfully completing my masters program and to my family and friends for their love, encouragement and unwavering support. You are the best!

Dank u wel voor de moeite die u hebt gedaan.

Chibuzo Obianuju Obi

Eindhoven, August 2008.

Abstract

The security of traditional security primitives depend largely on their sensitivity to minute variations in input. However in several important applications, especially in recent times, the input is inherently noisy. The increasing prominence of such applications has generated a lot of research in the development and optimisation of systems that can achieve adequate security with noisy inputs.

In this thesis we present an in-depth study of security with noisy data, especially with regards to its effective combination with cryptography. We focus on improving the functionality and security of one of such applications achieving robust security with noisy data, the helper data system. In particular, we identify potential areas of threats and vulnerabilities in various components of the helper data system and advise on ways to prevent and minimise the associated risks.

Chapter 1

Introduction

1.1 Security with noisy data

In today's world, the importance of protecting information and systems by securely identifying and authenticating individuals and objects is becoming increasingly important. Cryptographic primitives and protocols are employed in protecting information and establishing secure transactions.

In order to perform authentication based on something closely linked or related to an individual or an object, the physiological characteristics and physical properties of individuals and objects are used respectively. The link between these physical objects and cryptographic primitives is established by measuring the physiological or physical properties of the individual or object. These measurements are inherently noisy due to causes such as repositioning errors, temperature and pressure variations and a slight damage of the measured object,

Traditional cryptographic systems demand that the same security parameter (key, unique identifier) should be presented every time access is required. The secrecy and constancy of identification/authentication parameters enforces a high level of protection. A necessary condition for the security of these systems is that when the input (authentication parameter) is modified in the least possible manner, identification, authentication, anti-counterfeiting or any other process using the modified parameter should be unsuccessful. These systems exhibit a strong avalanche effect[1].

However, in certain situations and for certain reasons, it may be necessary to use data that is inherently noisy for identification, authentication, anti-counterfeiting and key storage purposes. This is the situation with the above mentioned security applications using the physiological features of individuals or the physical properties of objects. We use the term Noisy Unique Physical Objects (NUPOs) to denote both biometrics and lifeless objects used in security primitives. Developing a systematic, efficient and robust method of identifying and authenticating individuals and objects based on their noisy measurement data is a non trivial task.

A typical NUPO system is composed of two major procedures, the enrolment procedure and the authentication procedure. As their names imply, the enrolment phase enrols a legitimate object while the authentication procedure authenticates the object when required. Enrolment and authentication are achieved by querying (challenging) the NUPO and measuring and comparing its responses. NUPOs map challenges to responses. A challenge is a stimulus that is applied to a NUPO while the response is the reaction of the NUPO obtained through measurement. At enrolment, reference information (e.g. a template containing the NUPO response) is stored to facilitate future authentication of enroled objects and detect intruders. During authentication a live measurement of the NUPO is collected, processed and compared with the stored reference material. Any two calls to the same NUPO will produce different but closely related responses, hence it is important to device an efficient means of identifying responses from the same NUPO obtained at different instances. A common practise is to

[1] The avalanche effect is evident if, when an input is changed slightly the output changes significantly.

store the raw response. However this approach is fraught with perils. One of the most important is the lack of privacy of the response in biometric and key storage applications. A breach of privacy may lead to risks such as identity theft and cross matching between biometric databases which can be used to track individuals without their consent. Moreover, once the stored response are compromised they are compromised for life. Is is particularly undesirable for biometrics, because humans have limited biometric features that can be used as NUPOs. Others issues include the entropy and the durability of the stored reference material. A common solution is to protect the raw response with a so called shielding function[18], however this solution does not remedy all the above listed problems.

Helper data algorithms emerged to solve the above mentioned (and other) problems associated with security using noisy data as well as to optimise system functionality. Motivations for the emergence of helper data systems include:

- The need for the privacy of stored NUPO responses, especially for biometrics.
- The need to combine NUPOs effectively with traditional cryptographic primitives.
- The desire to create an efficient method of storing cryptographic keys securely.
- The need for an efficient method of correcting the noise present in NUPOs in a way compatible with general security requirements.

Helper data systems always have the purpose of noise elimination, combined with one or both of the following:

- Hiding the NUPO response (privacy-preserving biometrics and anti-counterfeiting).
- Hiding the extracted value (key storage).

A helper data algorithm is a method of extracting noise free, uniform bits strings from noisy sources. Extracting a uniform bit string from the NUPO is a privacy-preserving method for NUPO authentication. For example, a bit string extracted from the photograph of a person's face reveals the identity of the person less easier than the photograph itself. Bit string extraction from NUPO responses is particularly attractive not only because it preserves the privacy of the NUPO but also because the extracted string can be combined in various cryptographic algorithms such as hash functions. Furthermore the extracted string can be used as a cryptographic key when the string is extracted from a lifeless NUPO.

Two important cryptographic primitives, the secure sketch and the fuzzy extractor, were introduced in [1] to facilitate the achievement of security with noisy data. The secure sketch handles the exact reconstruction of a noisy input and the fuzzy extractor, the extraction of a consistent, uniform, noise free bit string from a noisy source. To achieve these non-trivial feats, something called the helper data is used, hence both systems are called helper data systems. For correctness, the helper data should be able to correct the noise in the NUPO measurements. For the security and privacy of the NUPO system, the helper data should reveal the least amount of information possible about the NUPO and the extracted bit string. The definitions in [1] of the secure sketch and fuzzy extractors accommodate the above mentioned issues. However the exact application, implementation and reuse may generate other vulnerabilities and threats that are not covered in these definitions. While the helper data corrects noise and aids in the reconstruction of the NUPO measurement and/or the extracted string, it introduces a number of vulnerabilities and if not handled carefully can lead to a compromise of the functionality and security[2] of the system. In addition, the helper data is usually considered as public data, which makes it more susceptible to attacks.

Security issues associated with the helper data systems has received considerable attention of recent due to the increasing popularity of security applications using noisy data. A substantial portion of this research has gone into ensuring that the extracted bit string is sufficiently long [1, 4], making sure that the helper data reveals the least possible information about the NUPO measurement and extracted

[2]The security and functionality of helper data systems are somewhat interwoven because helper data systems are security systems. When we speak about one, we speak about both.

string [1] and ensuring that multiple usage of the extractor for any specific NUPO is secure against insider and outsider attacks (reusable fuzzy extractors)[2]. When we assume that parties communicate over an insecure channel, authentication of the helper data (which may have been maliciously modified by an adversary) is required. This has been researched under various contexts in [2–5].

Other salient issues which may affect the performance and security of the bit string extraction procedure include inadequate estimate of the NUPO and NUPO noise distributions, ineffectiveness of error correcting techniques and insufficient knowledge about the behaviour of noise as it is propagated through the extraction system. We call these vulnerabilities intrinsic vulnerabilities because they are not perpetuated by an adversary but considerably affect the performance and security of the extractor in the helper data system.

Our Contributions

We identify and discuss extensively various intrinsic vulnerabilities in the helper data system. The NUPO distribution is estimated by sampling. Starting from the point of data entry, we determine the relationships between the size of the sampling population in relation to the distance between the real and empirical distributions. We also compute probabilistic bounds between the sample size and certain security parameters. Our results here can also be used in estimating the distribution of the NUPO's noise.

Using a generic construction of an extractor in a helper data system, we investigate the behaviour of noise as it is propagated though the extraction procedure. In particular we determine its shape on the unit interval which can be made analogous to the extracted string space.

In the area of noise correction, we identify and compare various error correcting techniques for correcting noise in a noisy bit string. The aim of the comparison is to identify the most suitable technique for optimising the length of the extracted string while ensuring that the noise is corrected. Our work in this section is strongly motivated by biometrics.

The importance of helper data authentication motivated us to carry out an in-depth study of helper data authentication mechanisms to detect unauthorised modifications to the helper data. We do an extensive literature study of existing helper data authentication methods as well as propose an authentication method using number theoretic hash functions. To accommodate drifts in NUPO measurements over time, it may be necessary to modify the helper data. We use the notion of sanitizeable signatures[11], to achieve secure helper data modification.

In discussing in detail these vulnerabilities and threats we aim at not only enhancing security but also at improving the functionality, efficiency and the overall performance of the helper data system.

1.2 Outline

We begin with the definition of some important terms, in the remaining section of this chapter.

Chapter 2 is devoted to an extensive discussion of helper data systems. We give an introduction to helper data systems in Section 2.1, discussing the motivation for such systems using two of its most important applications, in biometrics systems and lifeless NUPOs systems. The major challenges of helper data systems, noise and non-uniformity are presented in Section 2.2, while the helper data, its features and properties are treated in Section 2.3.

In Chapter 3 we discuss security issues of helper data systems. We identify some of the major threats (attacks by an adversary) and vulnerabilities (intrinsic vulnerabilities [3]) in the helper data system. We present various ways of classifying these threats and vulnerabilities and discuss one of the classifications extensively. In particular, we characterise threats and vulnerabilities based on the system component/module/process which they affect.

[3] Vulnerabilities resulting from inaccuracies in design parameters, algorithms and processes.

Chapter 4 gives a detailed description of the extractor in the helper data system. In Section 4.1 we give a generic construction for NUPOs which produce data that have a continuous distribution. The functionality, security and privacy concerns of the construction are discussed in the remaining sections of Chapter 4.

Motivated by the threats and vulnerabilities identified in Chapter 3, Chapters 5, 6 and 7 are each devoted to preventing and countering specific threats and/or vulnerabilities.

Challenges and security implications associated with estimating the distribution of the NUPO are discussed in Chapter 5.

Chapter 6 handles attacks on the helper data. In particular, we discuss various methods of authenticating the helper data. In addition, a procedure enabling secure helper data modification using sanitizable signatures is given in Section 6.2. Secure helper data modification is particularly useful to tackle the problem of natural drift in NUPO over time.

In Chapter 7, we identify and compare various error correcting techniques, with the aim of maximising the number of bits extracted from a binarised biometric feature vector.

We end with a summary of our results, recommendations and discussions for future work in Chapter 8.

1.3 Terminology

In this section we give some useful definitions.

- **Active attack:** The adversary is be able to transmit data to one or both of the parties (involved in a cryptographic protocol), modify and/or block the data stream in one or both directions.
- **Authentication/Verification:** The "one-to-one" process of comparing a submitted sample against the reference template of a single enrollee whose identity is being claimed, to determine whether it matches the enrollee's template.
- **Challenge Response Pair (CRP):** The stimulus applied to a NUPO and corresponding response.
- **Channel capacity:** Maximum achievable information rate. Maximisation conducted over all possible choices of transmission and detection techniques.
- **Denial-of-service attack:** The concerted, malevolent efforts of a person or persons to prevent a system/service from functioning efficiently or at all, temporarily or indefinitely.
- **Fuzzy extractor:** Cryptographic primitive extracting a uniformly random bit string from its input in a noise tolerant manner.
- **Helper data:** Derived during NUPO enrolment. Used to correct noise and extract uniform randomness from NUPOs.
- **Identification:** A means of seeking to find an identity amongst a database rather than authenticate a claimed identity. It is the "one-to-many" process of comparing a submitted sample against all of the stored reference templates/data to determine whether a match can be found for it.
- **Noisy Unique Physical Object:** Physical objects whose unique (and unclonable) features make them suitable for identification, authentication and key storage purposes.
- **Non-repudiation:** The concept of ensuring that a party in a dispute cannot repudiate, or refute the validity of a statement or contract.
- **NUPO response:** Information extracted from the NUPO measurement. It is the result of analysing and summarising NUPO response data and contains the unique characteristics of an object.

- **Passive attack:** The adversary does not interact with any of the parties involved, but attempts to break the system solely based upon observed data. The adversary can eavesdrop on a communication channel and/or monitor transmissions.
- **Secure sketch:** Cryptographic primitive allowing for the precise reconstruction of a noisy input.
- **Spoof:** A spoofing attack is a situation in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage.
- **System vulnerability:** A design flaw or feature that creates a security weakness. Suboptimal design not making use of resources.
- **Threat:** A danger which could affect the security (confidentiality, integrity, availability) of assets, leading to a potential loss or damage. It is the possibility for an attack.

Chapter 2

Helper Data Systems

Helper data system is the general term for security applications extracting a uniform bit string from a noisy source (i.e. from a NUPO). NUPOs fall under two broad classes based on lifeness. The first class consists of biometrics while the second class is made up of lifeless NUPOs. In the following, we will discuss two major classes of helper data systems[1], biometrics and lifeless NUPOs sysems.

Biometrics are presented in Section 2.1.1, while lifeless NUPOs are discussed in Section 2.1.2. We distinguish between biometrics and lifeless NUPOs systems in Section 2.1.3. The major challenges of working with helper data systems are discussed in Section 2.2, while Section 2.3 focuses on the distinguishing component of helper data systems, the helper data.

2.1 Noisy Unique Physical Object (NUPO)

NUPOs as the name suggests are physical objects whose unique and unclonable features make them suitable for a variety of security applications. By querying (challenging) the NUPO, measuring, storing its responses and then comparing these responses to freshly generated query responses at authentication, the presence of the correct NUPO is determined. NUPOs can be classified based on lifeness, hence we have biometrics and lifeless NUPOs. Other criterion for classification include, for example, whether or not they are integrated, whether challenges/measurements on the NUPO are controlled or uncontrolled. We present a few of these below.

- Bare NUPO: The bare NUPO consists only of the physical object whose uniqueness (and sometimes randomness) property is to be exploited. The NUPO measurement reader has unrestricted contact with the physical structure. Bare NUPOs can be used in anti-counterfeiting, brand/copy protection and token protection. In this setting the most important property utilised is the physical unclonability of the NUPO. For example, biometrics are bare NUPOs. However some biometrics (e.g. finger prints) are easily clonable, hence the liveness detection property also plays an important role.
- Integrated NUPOs: In integrated NUPOs the physical object and devices that collect, measure and process the NUPO response are integrated. For example, to construct an optical lifeless NUPO (Optical PUF) we need the following: a radiation source, a challenge-modifying element, a radiation scattering element (physical object e.g. rough surface), a radiation-detecting element (e.g. a camera) and an image processing device. To obtain an integrated optical NUPO, certain components for example, the challenge-modifying element, the rough surface, the camera and the radiation processing device may be integrated in a way that is hard to dismantle without causing substantial damage to each of the integrated components.

[1] With a slight abuse of notation we often use the same name for the noisy object and the security system based on the noisy object. However, the intended meaning will always be clear.

- Controlled NUPO (CNUPO)[13]: A NUPO is called controlled if it can only be accessed via an algorithm that is physically linked to the NUPO in an inseparable way (i.e., any attempt to circumvent the algorithm will lead to the destruction of the NUPO). This algorithm (which provides the control layer) can be used to limit the challenges that are presented to the NUPO and the information about the response that is given to the outside world. This setup achieves stronger security. The functionality of the control layer can be extended leading to secure key storage, certified execution and certified measurements. A controlled NUPO is always integrated but and integrated NUPO is not always controlled.
- Uncontrolled NUPO: A NUPO that is not controlled.

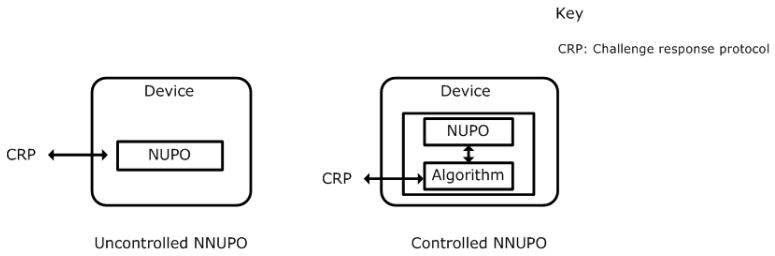

Figure 2.1: Controlled and uncontrolled NUPOs

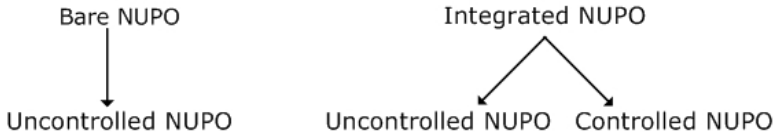

Figure 2.2: NUPO tree

Two important classes of security applications using NUPOs are:

- Identification, authentication and anti-counterfeiting using NUPOs.
  - Using biometrics
  - Using lifeless NUPOs
- Secure key generation/storage using lifeless NUPOs.

Other applications of noisy data in cryptography include key agreement by exchanging messages through a noisy channel[20] and true random number generation[21].

2.1.1 Biometrics

Definition 2.1.1 *Biometrics refer to the field of technology which focuses on identification of individuals by using measurements of physiological and/or behavioural characteristics, such as those based on retinal or iris scanning, hand geometry, fingerprints, face recognition, gait, ear prints and ear channel recognition. Liveness detection is usually involved in the authentication process.*

The following are examples of the various physiological and behavioural characteristics that are used in the construction of biometric identification and authentication schemes[19]:

- Face: analysis of facial characteristics.

- Fingerprint: analysis of an individuals unique fingerprints.
- Hand geometry: analysis of the shape of the hand and the length of the fingers.
- Iris: analysis of the coloured ring that surrounds the pupil.
- Signature: analysis of the way a person signs his name.
- Voice: analysis of the tone, pitch, cadence and frequency of a person's voice.

As the level of security breaches and transaction fraud increases, the need for highly secure identification, verification/authentication technologies is becoming apparent. Because biometrics identify individuals based on each person's unique physical or behavioural characteristics, it is often incorporated in an extensive array of highly secure identification and authentication solutions. These solutions are used in secure electronic banking, network security infrastructures, government ID's, law enforcement, access control, health services, border control and social services.

Though biometrics are used primarily for identification and authentication, they can also be integrated with other technologies such as digital signatures and encryption. In addition to uniquely identifying an individual (thereby providing an audit trail), using biometrics related technologies is attractive because they are convenient in that the user is not required to remember long passwords.

Furthermore, some of the weakness of standard identification and authentication schemes can be overcome by using biometrics. While passwords can be lost, biometrics cannot be lost or forgotten. Requiring that the person being authenticated be present at the time and point of authentication is an example of additional security feature that biometrics offers which cannot be easily enforced in regular authentication systems.

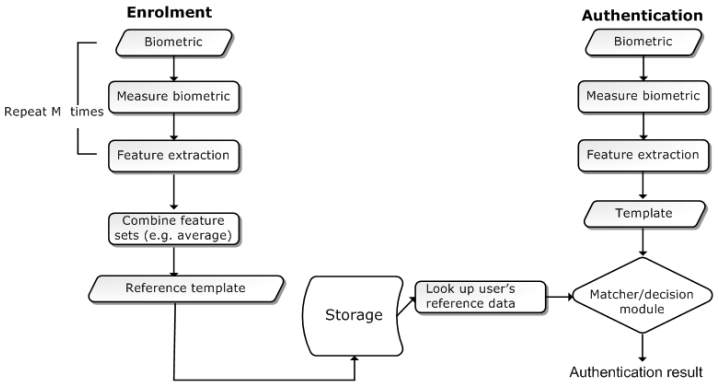

Figure 2.3: A biometric system

Biometric systems (Figure 2.3) work by matching patterns of live individuals in real time against enrolled records. The typical biometric authentication system is composed of the following: a measurement taking device, a feature extraction function, template creation (and further processing of biometric template), storage and comparison. A biometric template results from analysing and summarising biometric data. It contains the unique characteristics of a user's biometric information and is the master copy that each future data acquisition would be compared to.

While measuring biometrics, it is often unavoidable that noise and other aberrations occur. Noise in biometric data is caused by a number of reasons, for example by the biometric measurement system, temperature and humidity variations, small repositioning errors and a slight damage of the measured object[13]. The noisy biometric measurement cannot be used (without processing) as say passwords in cryptographic settings because the these settings require noise-robust inputs.

Because biometrics are noisy and the behavior of the noise cannot be determined completely, biometric systems work with probabilities. They are not exact methods (in contrast to methods based on knowledge or possessions like PINs, passwords and tokens). This limitation (noise) results in false acceptances and false rejections. The False Acceptance Rate (FAR) is the success probability for an unauthorised user or a user that does not exist within a biometric system to be falsely recognised as a legally registered user. In contrast, the False Rejection Rate (FRR) rate is the probability that the legally registered user is falsely rejected by the biometric system when presenting his biometric feature[17]. The FAR and FRR are negatively correlated. Depending on how one adjusts the sensitivity of the mechanism that matches biometric measurements, the FAR and the FRR can be made to vary significantly.

One of the major challenges with using biometrics for identification, authentication and in information security in general, is that once an individual's biometrics has been compromised, they can not be changed, recovered or reissued. To remedy this, various solutions have been proposed, in [12], Ratha et al. introduce the concept of cancelable biometrics. Another practical approach introduced in [6] is to use the biometric data to retrieve a secret value linked to each user, instead of using the biometric directly. This secret value is used for security applications instead of the biometric value itself.

Even though biometrics cannot be regarded as "real" secrets (because for example people leave their fingerprints everywhere), it is still important to protect people's privacy. The following dangers exist when biometric data is not properly protected: identity theft, ability to track peoples records, cross-matching between databases and medical information leakage from the biometric template.

To protect the privacy of biometric data, it is a common practise to store a hash or an encryption of the biometric template and not the biometric template explicitly. However, hashing and encryption do not solve all the problems of template protection. In [13], it is shown that the straightforward application of encryption does not provide adequate template protection when the verifier is malicious or against insider attacks. The challenges of efficient biometric template protection has motivated a large amount of research. Results include cryptographic primitives such as the fuzzy commitment[7], the fuzzy vault[22], the secure sketch[1] and the fuzzy extractor[1]. In this thesis we will discuss the last two primitives in detail.

Biometric templates can be stored on a reader or sensor, on a smart card or token or in a database. One of the most potentially damaging attack on a biometric system is against the biometric templates stored in the system's database. Attacks on stored reference templates include the follow[16]:

- A legitimate template can be replaced by an impostor's template.
- A template can be stolen and later replayed to the matcher to gain unauthorised access. The stolen template can also be used with other systems that accept biometric templates.
- Cross-matching between databases. The biometric identifiers can be used for purposes other than the intended. This results in invasion of privacy. For example a fingerprint template stolen from an bank's database may be cross linked to a persons health record.
- A physical spoof can be created from the template to gain unauthorised access to the system.

Given the possible attacks on the stored reference data, a good template protection scheme should possess the following qualities:

- It should be hard (at least computationally) to obtain the original measurement from the protected template.
- The protected template should not readily reveal the identity of its owner.
- A good template protection scheme must not allow for cross matching across databases, ensuring user privacy.

In the case of compromise, it is desirable that the protected template should be revocable, while allowing for the reissue of a new one based on the same biometric. It is difficult to produce an effective template protection scheme that achieves this. However, efforts have been made in this area, in Cancelable biometrics [12]. In the following, we present the state of the art in biometric template protection.

In practise it is common to apply a transformation function[16] or a shielding function[18] to the freshly generated biometric template before it is stored. The parameters of the transformation/shielding function are usually derived from a randomly chosen key. These functions can be divided into two classes, those that are invertible and those that are not. The security of invertible transformation function is based on the secrecy of the randomly chosen key. This template protection method is known as the Salting[16]. The security of the non-invertible shielding functions, is on the non invertibility of the shielding function. Table 2.1, adapted from [16] gives a summary of different template protection schemes.

Table 2.1: Template protection methods.

Approach	Method of achieving security of template	Entities stored	Intra-user variation handled by	Advantages	Limitations
Invertible transform (Salting)	Secrecy of key	Public: transformed template Secret: key	Quantisation and matching in transformed domain	Introduction of key results in low FAR. On template compromise, revocation is easy.	If user-specific key is compromised, template is insecure.
Non invertible transform	Non invertibility of transformation function	Public: transformed template Secret: key	Matching in transformed domain	Provides better security than salting.	Tradeoff between discriminability and noninvertibility of transformation function.
Bit-string binding	Level of security depends on how much information helper data reveals.	Public: helper data	Error correction and user specific quantisation.	Biometric securely linked to a key.	Needs proper error correcting mechanism. Helper data must be designed carefully.
Bit-string generating	Level of security depends on how much information helper data reveals.	Public: helper data Extracted bits	Error correction and user specific quantisation.	Most appealing template protection. Can be used for cryptographic purposes.	Difficult to extract bit string with high stability and entropy.

Extracting a bit string from a biometric provides better security for protecting the biometric template. For a bit string to be extracted a helper data method is needed. These helper data methods can be further classified into two, depending on how the helper data is obtained. The first method entails monolithically binding a key (that is independent of the biometric features) with the biometric template. The helper data here is the single entity that embeds the biometric template and an error correcting code (selected using the key). Matching involves recovery of the key from the helper data using the query biometric features. During authentication, if the new biometric query differs from the template within a certain error tolerance, the associated code word with similar amount of error can be recovered which can be decoded to obtain the exact code word and hence, recover the embedded key[16]. Juels and Wattenberg's fuzzy commitment scheme [7] is a well-known example of a key binding approach. In such systems, given only the helper data, it should be computationally hard to recover either the key or the original template [16]. In the second method, helper data is derived solely from the biometric template. Bit string extraction from biometrics is an appealing template protection approach which is useful in cryptographic applications. Its major limitation is that extracting a bit string that is stable and has sufficient entropy is a non-trivial task. This last category is discussed extensively in this thesis.

Security assumptions for biometric systems
The following assumptions are made about security in the construction and functionality of a typical biometric system:

- Enrolment is performed by a Trusted Authority (TA).
- During the authentication phase, an attacker is able to present artificial biometrics at the sensor.
- The sensor is trusted not to give out any information about the measured biometric. Biometric measuring and processing during authentication are assumed to be tamper resistant.

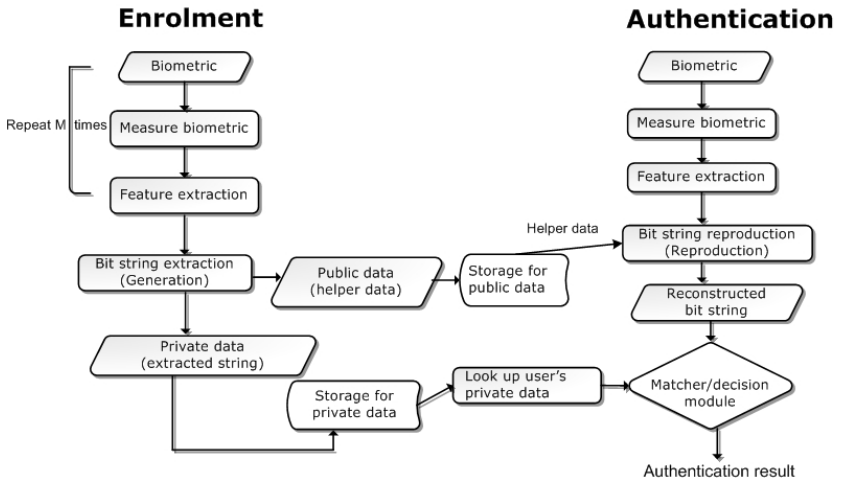

Figure 2.4: Biometric template protection by bit string extraction

- The communication channel between the sensor and the authentication authority is public and an attacker can mount both passive and active attacks on messages sent across the channel.

Security requirements for biometric systems

- The information in the storage should give the least information possible about the original biometric and should not give sufficient information to allow for successful impersonation attacks.
- During the authentication procedure, the verifier should not have access to the unprotected biometric measurements. The device that contains the sensor is trusted by the verifier but it does not reveal the biometric data to the verifier.

2.1.2 Lifeless NUPOs

Definition 2.1.2 *A lifeless NUPO is a physical object with the following properties:*

- *The physical object must be easy to make but hard to clone physically.*
- *It should be difficult to characterise the physical structure of the object.*
- *The object can be subjected to a large number of different challenges that yield unpredictable responses.*
- *Mathematical cloning of the challenge-response mechanism is (computationally) difficult.*
- *The interaction between the probe and the physical system produces an output quickly, but computationally simulating this interaction is difficult.*

Many physical objects have features that are unique and difficult to clone. These features can be exploited for identifying the object and for extracting a binary string, to be used for identification and authentication as well as, as a cryptographic key. The Integrated Circuit (IC) is a common example of such an object. The statistical variation in the delay of devices and wires within an IC enables the unique identification of manufactured ICs, even when they are from the same lot or wafer. These variations which do not affect the performance of the ICs can be exploited to provide a unique means of identification and source of randomness. This concept is summarized in the term Physical Unclonable Functions (PUFs), also called Physical One-Way Functions (POWFs), Physical Random Functions (PRF) and Physically Obscured Key (POK). In this thesis we adopt the terminology lifeless Noisy

Unique Physical Object (lifeless NUPO) or simply NUPO when talking about both biometrics and lifeless NUPOs.
The following physical systems exhibit uniqueness and randomness properties that can be exploited as lifeless NUPOs.

- Static Random Access Memory "(SRAM) start up values: The SRAM is a semiconductor memory. The startup values of an SRAM are unique and exhibit some form of randomness.
- FPGA Butterfly: The FPGA (Field-Programmable Gate Array) is a special type of semiconductor device containing programmable logic components called logic blocks and programmable interconnects. FPGA butterfly refers to a certain FPGA configuration that mimics the SRAM startup process.
- Spraying an Integrated Circuit (IC) with a coating mixture (Coating PUFS): In Coating PUFS, the mixture consists of a matrix material which is doped with random dielectric particles. An example of a coating mixture is TiO_2 and TiN particles in a matrix of aluminophosphate. The local capacitance measurement of each coated IC is unique.
- Diode breakdown voltage: This is the minimum reverse voltage to make the diode conduct in reverse.
- Read/write times in a non-volatile memory.
- Speckle pattern: A speckle pattern is a random intensity pattern produced by multiple scattering of coherent light. Prominent examples include the seemingly random pattern created when a coherent laser beam is reflected off a rough surface.

The uniqueness and unclonability property of lifeless NUPOs make them useful for a large number of security applications. They are used as a means of identification and authentication, as well as randomness extraction, hence they can be used for key storage. Lifeless NUPOs provide a source of high entropy but produce data (responses) that is both noisy and non-uniform. Variations in temperature and power supply are two of the primary causes of noise in NUPO responses. For example, in silicon PUFs, circuit delays which determine the NUPO response are sensitive to environmental variations such in temperature and power supply voltages.

Most lifeless NUPOs arise from random manufacturing variations, the manufacturer cannot make two identical NUPOs even if he wants to. Because lifeless NUPOs are difficult to clone, embedding them into devices, make the device unclonable. This attribute is useful in anti-counterfeiting. The challenge-response pattern of a NUPO changes significantly when the NUPO is damaged, this together with its unclonability property make suitable for securely storing cryptographic keys. Another attraction to the use of lifeless NUPOs in security systems is that their production and implementation are relatively inexpensive.

Figure 2.5 depicts a lifeless NUPO system.

During enrolment the lifeless NUPO is queried with a large amount of challenges. The responses to these challenges or a function of them is stored. During authentication, the lifeless NUPO is queried with a particular challenge chosen uniformly at random from the initial set of challenges. The response (or a function of the response) is compared to stored data. The presence of the correct lifeless NUPO in a security application is verified by measuring the lifeless NUPO's response to some specific challenge(s) and comparing it to the stored reference data. The stimulus/challenge (physical probe) and response are usually called a Challenge Response Pair(CRP). The challenge, which can be viewed as a random function[2] can only be evaluated with the help of a specific physical system. As with biometrics, the lifeless NUPO's response can be stored in various forms. It may be stored as a template or as a string of bits. Extracting a bit string from the lifeless NUPO response is the most secure method for storing the lifeless NUPO response.

[2] A function for which knowing some outputs does not help one guess the output for other inputs.

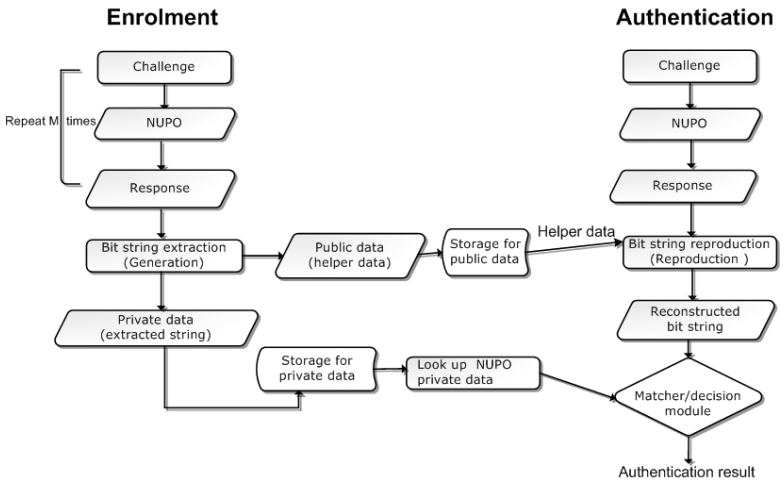

Figure 2.5: A lifeless NUPO system

Security requirements for lifeless NUPO systems
A lifeless NUPO security system is expected to fulfill the following requirements:

- The physical object used should satisfy all the properties listed in the definition of a lifeless NUPO .
- The response should not reveal any information about the structure of the lifeless NUPO.

2.1.3 Difference between biometrics and lifeless NUPO systems

Certain similarities and differences exist in the applications of biometrics and lifeless NUPOs. Biometrics and lifeless NUPOs both provide a means of uniquely identifying objects based on they physical and physiological properties and characteristics. Both produce data that are noisy and non-uniform and are applied in similar security applications. For example, a strong similarity exists between biometric authentication and anti-counterfeiting using lifeless NUPOs. In this application, the most important feature used by these systems is the uniqueness of the NUPO response. Uniformity of the extracted string is not strictly required. Strict secrecy is also not required. However these features help for privacy and provide extra defence against cloning.

In the following, we distinguish between biometric and lifeless NUPO systems.

- Biometrics can be tested for liveness while lifeless NUPOs (generally) cannot be tested for liveness.
- Although biometrics are secure they cannot be used as real secrets. However lifeless NUPOs may be used as secrets, for example, the Physical Obscured Key (POK).
- Lifeless NUPOs are physical objects and they offer their designer the freedom to design a physical structure that carries a lot of information, more information than biometric sources.
- For mass deployment (e.g. in the case of anti-counterfeiting), one can find lifeless NUPOs that are inexpensive to produce and that combine easily with low-cost readers. This is not the case with biometrics.
- Lifeless NUPOs perform all the functions of biometrics (except liveness detection), in addition, they provide a potentially secure source of generating truly random and uniform bit strings that can be used as cryptographic keys, i.e. lifeless NUPOs can serve as key storage.

- Biometrics systems focus primarily on identification and authentication using the extracted string as a means of providing adequate biometric template protection. On the other hand, security applications using lifeless NUPOs focus on the extraction of uniformly random bit strings for a wide range of cryptographic purposes.
- Though the compromise of a lifeless NUPO system is undesirable, its consequences are not as severe as in a biometric system. In the event of a compromise in a biometric system, the biometrics of users are compromised for life and cannot be reissued. This is because biometrics (in general) cannot be revoked or cancelled. In lifeless NUPO systems, a compromise of the system can be remedied by revoking affected NUPOs and replacing them.
- Copying a biometric is generally much easier than copying a lifeless NUPO.

2.2 Noise and non-uniformity in data

The secure sketch and the fuzzy extractor [1]are cryptographic primitives designed to address the issues of noise and non-uniformity of data in helper data systems, respectively.

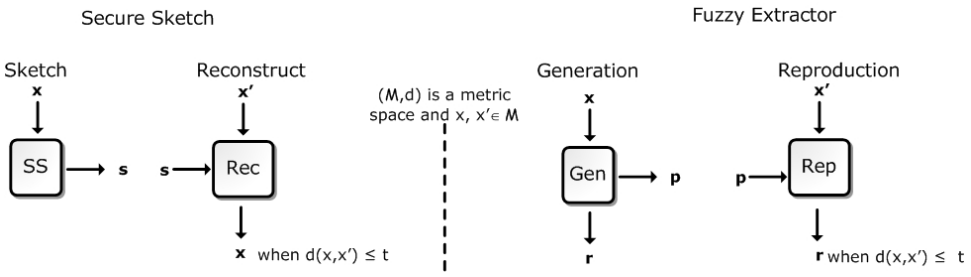

Figure 2.6: The secure sketch and fuzzy extractor

2.2.1 Addressing noise in data

The secure sketch allows for the precise reconstruction of a noisy input by using some public information derived from the NUPO measurement. On input x, the sketch SS procedure outputs a sketch (also called helper data) s which will be used to correct the errors in future noisy measurements of x. Given a noisy version x' of x and s, it is possible to recover x as long as the distance between x and x' is within certain tolerable bounds. The sketch is secure in the sense that it does not reveal a lot of information about x, so instead of storing x for fear that later readings will be noisy, one can store s, which will aid the recover of the x from future queries, x'. The sketch s is considered public. While the secure sketch attempts to correct noise in data, it does not address the issue of non-uniformity.

2.2.2 Addressing non-uniformity of data

NUPOs do not produce uniform data. Extractors are used to transform a non-uniform bit string to a nearly uniform bit string of shorter length. It is nearly uniform in the sense that the total variation distance (statistical distance) between the extracted string and a uniform string of the same length is negligible. The fuzzy extractor enables one to extract a uniformly random bit string that is easily reproducible from a noisy, non-uniform source in an error tolerant way. The extraction is error tolerant in the sense that the extracted "nearly uniform randomness" will be the same even if the input changes, as long as it remains reasonably close to the original input [1]. The fuzzy extractor unlike the secure sketch addresses both the problems of noisiness and non-uniformity in data. Uniformity of the extracted sting is necessary when the extracted string is to be used as key in cryptographic primitives

such as encryption and message authentication codes . Uniform strings contribute to the compact representation of enrolment data. Furthermore extracting a uniformly random bit string fastens the reference data retrieval process.

A fuzzy extractor consists of a string extraction/generation procedure `Gen` and a string reproduction procedure `Rep`. The generation procedure takes as input measurement data x and outputs a uniformly random bit string r and helper data p. The reproduction procedure, takes as input a noisy version x' of x and the helper data[3] p and outputs the bit string constructed in the generation procedure if x and x' are similar. The fuzzy extractor may be seen as an extension of the secure sketch.

We differentiate between helper data and data output from the enrolment procedure. We clarify terms by explaining what each stands for in both the secure sketch and the fuzzy extractor. In the secure sketch, the helper data, is any information that aids in the exact reconstruction of the noisy input. The output of the sketch procedure is called the helper data. In fuzzy extractors, a unique identifier bit string is output as well as the helper data. The extracted string may be kept secret (e.g. in the Physical Obscured Key), and is in this case called private data. We do not consider the extracted string as part of the helper data since it is used only in the comparison/authentication procedure after the reconstruction process. Because helper data is usually considered public, it is often called public data.

2.3 Helper data

Definition 2.3.1 *Let x be any given value. Let x' be a noisy version of x,*

- *The **helper data** is any information that aids in the exact reconstruction of x from x'.*
- *Let $f(x) = r$, for some function f. Then the **helper data** is any information that allows for the exact reconstruction of r, from a noisy version x', of x.*

2.3.1 Is the helper data necessary?

Theorem 2.3.2 [18] *For security applications involving bit string extraction from noisy data, helper data is necessary.*

We elucidate Theorem 2.3.2 by means of the following example. One might reason that since helper data is used to correct errors, one may use error correcting codes directly without having to use a helper data. However error correcting codes by themselves are not a full solution. We will see why in the example below.

Example 2.3.3 *The NUPO measurements are represented as elements of an n-dimensional vector space $V_n(q)$, over a finite field $\mathrm{GF}(q)$ with q elements. Let C be an $[n, k, \delta]$- linear code with $\delta \geq 3$ (See Appendix B for short introduction to error correcting codes). The measurement data x is viewed as the received word [4]. The extracted string is the code word nearest to that received word x. An authentication system for noisy data is constructed by partitioning the vector space, into disjoint spheres of radius $e = \lfloor \frac{\delta-1}{2} \rfloor \geq 1$ around the code words such that these spheres together cover $V_n(q)$. Every word in $V_n(q)$ is at distance at most e from a unique code word. Each code word $c \in C$ describes a class, which is defined by a ball of radius e around c.*

Without helper data: *At enrollment, the NUPO is measured and $x \in \{0,1\}^n$ is obtained. x belongs to class c_1 but lies around the boundary of classes c_1 and c_2 (see Figure 2.7). x is e away from c_1. x is $e+1$ away from c_2. Let v be a unit vector of length n. During the authentication phase, the noisy measurement $x' = x + v$ is obtained. From the picture we can see that the noise has "pushed" the measurement data x into class c_2. $d(x', c_2) < d(x', c_1)$, so x' is mapped to c_2. Note that even though the noise in the measurement does not*

[3] The helper data is denoted by s in the secure sketch and by p in the fuzzy extractor.

[4] In error correcting codes, the received word is a code word (transmitted through a noisy channel) plus errors due to transmission through the noisy channel.

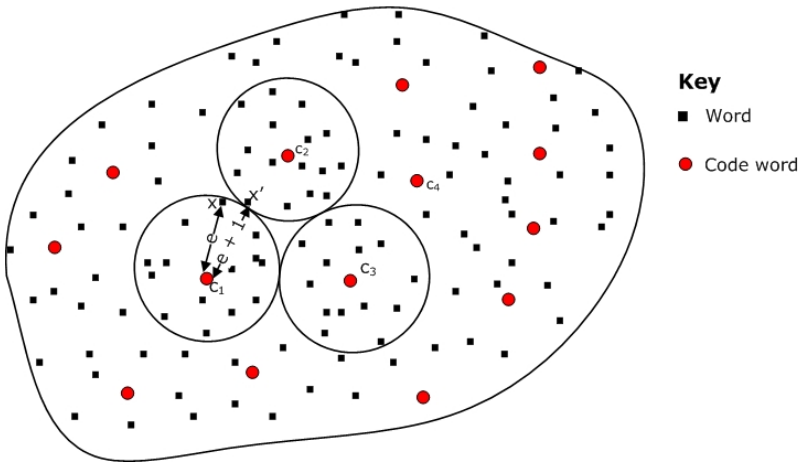

Figure 2.7: Helper data is necessary for consistent bit string extraction from with noisy data.

exceed the stipulated error threshold e, the reconstruction procedure will output c_2, which does not correspond to the value created in the enrolment phase and authentication will fail.

The presence of helper data this system, will ensure that the measurement data is always pushed towards the center of the class to which it belongs, this way as long as the noise is within the specified limit, the right value will always be returned.

With helper data: *Next, we see how the helper data effectively corrects the noise in the measurement x.*

On input x, a random codeword c_1 is selected. The helper data s, is the shift needed to get from c_1 to $x : s = x - c_1$. c_1 is the extracted bit string. During reconstruction $x' = x + v$ is input. Using the helper data s, we compute $c' = x' - s$. c' is decoded to obtain c_1 (because $d(x', x) = 1$, so is $d(c', c_1)$). So c_1 is reconstructed and authentication succeeds. □

2.3.2 Properties of the helper data

The helper data is derived during the enrolment phase. The helper data is a "personalised" quantity. If the helper data were to be the same for every NUPO, it would be included as a system parameter, this would mean that for such a security system, helper data is not needed. However Theorem 2.3.2, tells us that this cannot be this case. The nature of the helper data depends on the NUPO.

It is desirable that the helper data possesses the following qualities.

1. Ability to efficiently correct noise: The major function of the helper data is to correct the noise in future NUPO measurements. An good helper data should ensure (with very high probability), that noisy input x' similar to x is mapped to the same decision region taken by the x. As a consequence, exactly x or r is reproduced using s or p respectively.

2. Reveals the least information possible about the NUPO: It is important that helper data reveals the least information possible about NUPO and the extracted string. If not, an adversary seeing only the helper data will be able to reconstruct (with a reasonably high probability) r and/or the corresponding NUPO measurement. Ideally the helper data should be statistically independent of the extracted string.

3. The helper data should be indistinguishable: Let p_1 be the helper data from measurement data x_1. Let x_2 be an arbitrary measurement data of the same type as x_1 (possibly from a different object). Given p_1, x_1 and x_2 an adversary without access to the generator function, should not be

able to easily identify the measurement to which p_1 belongs. Indistinguishability is important in biometrics, because privacy is paramount.

The helper data may contain the following features;

- Pointers: The helper data may contain pointers to the special features of a NUPO that uniquely identify it. If the output from querying the NUPO is an image, the helper data may be, pointers to sections of the image that remain unchanged in different realisations of the same NUPO. Examples of such features are minutiae endings and bifurcations in fingerprints.

- Nudge/shift measurement to the middle of quantisation interval: When the NUPO system incorporates a quantisation mechanism, the helper data may be such that it nudges/pushes a particular measurement data to the centre of the quantisation interval to which it belongs. This is particularly important for measurements that lie on/around the boundary of two or more partitions. The helper data in this setting will be "how much shift or nudge" that is required to get a measurement to the centre of quantisation interval it belongs to.

  An example of this type of helper data is the Code Offset method[7]. Let $\mathcal{F}$ be a field and C an $[n, k, 2t + 1]_{\mathcal{F}}$ error correcting code (not necessarily linear). C is used to correct the in measurement data x. On input x, a random code word c is selected. The helper data s, is the shift needed to get from c to $x : s = x - c$. The reconstruction procedure takes as input the new measurement x' of the NUPO and the helper data s and computes $c' = x' - s$. c' is decoded to obtain c (because $d(x', x) \leq t$, so is $d(c', c)$). x is reconstructed by shifting back, $c + s = x$.

- Finally, if a unique identifier string is extracted from the NUPO using a randomness extractor, for example the 2-universal family of hash functions, the choice of the hash function used from this family may be specified as part of the helper data.

2.4 Chapter summary

In this chapter, we started by introducing the term, Noisy Unique Physical Object (NUPO), to stand for physical objects the whose uniqueness properties make them suitable for a variety of security applications. We presented the common NUPO systems and gave detailed descriptions of how they function. The NUPO systems discussed were biometrics 2.1.1 and lifeless NUPO systems 2.1.2. These systems have a lot of similarities as well as a few differences, these are discussed in Section 2.1.3. In Section 2.2 we discussed the challenges (noise and non-uniformity) of NUPO systems and these challenges are countered using helper data algorithms. Section 2.3 is devoted to an insightful discussion about the helper data, why they are necessary, their properties and features.

Chapter 3

Security Issues in Helper Data Systems

In this chapter we discuss security issues in helper data systems. We identify and classify the major threats and vulnerabilities in achieving security with noisy data using helper data systems. We classify these threats and vulnerabilities according to the system component which they affect. In Section 3.1 we present the vulnerabilities and threats associated with data collection. Threats associated with public data storage and transmission is given in Section 3.2, while vulnerabilities associated with error correction are discussed in Section 3.3.

In discussing security issues associated with bit extraction from noisy data, the fist logical questions will include,

- What exactly do we need to protect?
- Against what/whom?
- What do we hope to achieve and how can we achieve it?
- How does improved security affect the performance and functionality of the system?
- What is the cost of security?

In answering the questions above we aim at minimising the impact and effect of security incidents on the helper data system. The importance of security can not be overemphasised in NUPO systems, whose primary function is to provide a secure means of identifying and authenticating individuals/objects.

We start by mentioning the goals of NUPO systems. Mentioning these goals will give us a headway in answering the above listed questions. NUPO systems aim at providing a secure means of identifying objects, so that system/information access can be granted to authorised parties and denied to unnauthorised individuals and objects. The NUPO system also needs to guard against falsely rejecting legitimate users/objects, either because of some internal malfunctioning or as a result of being manipulated by an adversary. Depending on the system, privacy of individuals or objects may also be a major concern. This is particularly the case with biometrics.

The following characteristics are desirable of any NUPO system, and infact of every security application.

- Confidentiality: Access to sensitive information and processes should be restricted and disclosed only to authorised persons.
- Integrity: Information should be accurate and complete.
- Modification of information and processes should be performed only be authorised persons.

- Availability: The security services provided by NUPO systems should be available when needed. Services should be reliable.

Successful compromise of the security of the NUPO system results in intrusion[1] and/or denial-of-service, which leads to a breach in confidentiality, integrity, availability and lessened usability.

Security breaches are most commonly the result of exploited vulnerabilities so it is important to adopt an approach that assess all risks, as failing to assess any one aspect can lead to a catastrophic failure of system security. To guarantee strong security, systems are usually considered only as secure as their weakest component.

Adequate caution and security measures are required to prevent mistakes which make the system vulnerable to attacks and manipulation by an adversary. In addition to protecting the system against attacks by an adversary, for robust security, components of the system presenting loopholes should be treated appropriately. Failure to handle these subtle vulnerabilities properly, may present an attacker opportunities which she may exploit to compromise the security of the system.

We envisage that by carefully considering all vulnerabilities and threats and providing suitable corrections and counter measures, we improve the performance of the NUPO system.

Any system desiring a high level of security should be as accurate as possible, as inaccuracies create the opportunity for manipulation and compromise. By accurate, we mean accuracy in data input and the correctness and efficiency of the system's algorithms. NUPO systems are however characterised by noise and so we expect a certain degree of inaccuracy in NUPO systems. This is because the properties and behaviour of the noise in the NUPO cannot be determined with 100% accuracy. The vulnerabilities resulting from the inaccuracies in the design parameters, algorithms and processes in the NUPO system are called intrinsic vulnerabilities [16]. For example, insufficient knowledge about intra-user/object variations can lead to high FRR. Intrinsic failures are particularly dangerous because they occur even when there is no explicit effort by an adversary to circumvent the system. And so are known as zero effort-attacks[16]. Zero effort attacks pose a serious threat if the false acceptance rate or the false rejection rate is unusually high. In addition, an adversary may exploit an intrinsic vulnerability to mount an attack on the system.

In the following, we will discuss the threats and vulnerabilities in the NUPO system. These threats and vulnerabilities can be classified into various categories according to various criteria.

Classification A

- Intrinsic vulnerabilities.
- Attacks by an adversary.

Classification B

- Vulnerabilities and attacks– enrolment.
- Attacks– storage.
- Vulnerabilities and attacks –authentication .

Classification C
Classification of vulnerabilities and attacks based on the system component/module affected. We consider the following

- Vulnerabilities and threats associated with data collection.
- Threats associated with data storage and transmission.

[1] An illegitimate party gains information /access to system and system data

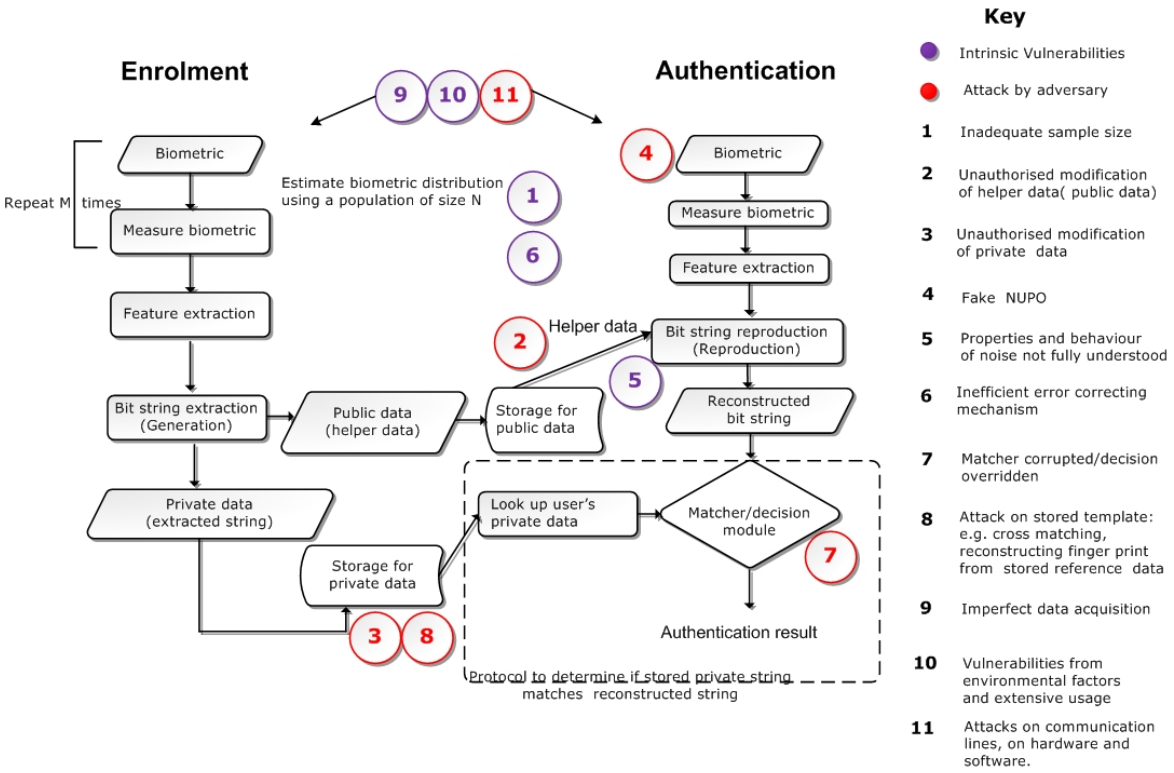

Figure 3.1: Threats and vulnerabilities in the NUPO system (biometrics)

- Vulnerabilities associated with error correction.

There exist overlaps among the above listed methods of classification. In this thesis we classify the threats and vulnerabilities in the NUPO system according to the system component/module which they affect.

3.1 Vulnerabilities and threats associated with data collection

We further classify the threats and vulnerabilities associated with data collection into intrinsic treats and attacks by an adversary.

- Intrinsic vulnerabilities
  - Estimating distribution of the NUPO: The NUPO system is designed to work with data from a specific input source. However as the distribution of the NUPO source is estimated and not fully known it is often the case that extraction system designed for distribution $\hat{P}$ has as input data with distribution P. This discrepancy introduces some vulnerabilities as an attacker can learn the distribution of P better than the original designer and thus have more knowledge about the bit string extracted from the system, than the designer. We elucidate more on this vulnerability and derive relationships and probabilistic bounds among the sample size N, the distance between P and $\hat{P}$, and certain security parameters in Chapter 5.

- Estimating distribution of the noise in the NUPO: The sample size affects our knowledge of the distribution of the noise, as this quantity also has to be estimated. The results of Chapter 5 can be used to estimate the distribution of the NUPO noise.
- Imprecise data collection: Another source of error in data collection is the error of inaccurate measurements due to the malfunctioning of instruments, poor procedures or significant damage of measured object. Biased observations due to inaccurate measurement can be innocent but very devastating, as they do not give a true representation of the object being measured. This can lead to high false rejections rates. To counter this vulnerability, sensor designs should acquire the biometric traits of an individual and the response from the lifeless NUPOs in a reliable and secure manner.

- Attacks by adversary
  - Threat resulting from presentation of fake NUPOs: This attack can occur during both enrolment and authentication. The presentation of fake NUPOs is the one of the most important attacks on the input device of a NUPO system. This attack can be relatively easily conducted as little or no technical knowledge of the system is required. The degree of vulnerability of the system to this type of attack largely depends on the nature of the NUPO. The vulnerability of the system to this attack is amplified because this attack is conducted at the point of entry to the system where so many of the digital protection mechanisms, such as encryption and the use of digital signatures, which are in place are not effective.

    While some NUPO are easy to forge, e.g. fingerprint and hand written signature, others are quite difficult to forge, e.g. iris and retina scan and integrated circuits. The difficulty of fake NUPO attacks depends on the implementation of a specific system. In some instances (especially for biometrics) a copy of object of authentication can be relatively easily obtained with or without the consent of the owner, for example, we leave our finger prints everywhere, digital cameras and recording technologies make it easy to acquire images and voice recordings[15].

    A general measure to counter falsification attacks in the biometric systems is to perform a "liveness" test, this verifies that the biometric sample presented to the input device is from a living person. In addition multimodal systems can be used to check this attack. A multimodal NUPO technology uses more then one NUPO identifier to identify an object.

    The fake NUPO attack also includes impersonation attacks, which involves changing an object's appearance (e.g. voice or hand written signature in biometrics) to match that of an authorised object. This problem can also be countered using multimodal NUPO system.
  - False Enrolment: The security and accuracy of the NUPO system is based on legitimate enrolments. Proper care should be taken to ensure that only valid objects are enrolled. Because once registered, the system will validate a false identity, once validated the false identity can gain access privileges (e.g. enrolling a blank finger print)[15].
  - Attacks on input device: The input data acquisition devices for NUPOs vary from cameras, to scanners, audio devices and desktop peripherals. Understanding device limitations is important for assessing the possibilities of attacks. A general requirement is that the input device must be consistent over time [14]

    A coercive attack is an attack where the legitimate objects's NUPO data is presented in an illegitimate scenario. For example an attacker physically forces a genuine user to identify herself to an authentication system. Tackling coercive and other implementation specific attacks such as this requires the creativity of the designer. For example, to curtail coercive attacks at the Automated Teller Machine(ATM), security cameras can be installed.

3.2 Threats associated with public data storage and transmission

The output of the enrolment phase is some public data (helper data) and some private data. While the private data is stored secretly and used only for comparison during the authentication phase, the helper data and other public data may be made public and/or need to be transported to the point of reconstruction/authentication when the need arises. As expected, more resources and effort are invested into securing the storage of the private data than for the public data. In most applications, for example in online authentication, the helper data has to be transported from its storage to the point of reconstruction (authentication). The communication channel through which the helper data is transported is usually susceptible to both passive and active attacks. These attacks are particularly important because their success may lead the attacker to

- Gain valuable information about the stored private data.
- Gain unauthorised access.
- Prevent a legitimate object access to the system.

To counter this attack, a helper data authentication mechanism is put in place. All helper data protection mechanisms have the following in common, whenever the helper data is modified by an unauthorised entity, authentication should fail (see Section 6.1 Chapter 6). This implies that the helper data system should be tamper evident.

In Chapter 6, we present an extensive study of helper data protection mechanisms.

3.3 Vulnerabilities associated with error correction

When the noise has been studied carefully and understood, a suitable error correction technique can be chosen to correct the noise. Choosing an error correcting technique for NUPO systems is particularly challenging because after binarisation of feature vectors, the resulting string has bits with varying bit error probabilities. The challenge here is to reduce this bit string whose bits have varying bit error probabilities to a shorter, consistent string that is easily reproducible and noise free. We aim at minimising the amount of bits lost while guaranteeing that the extracted bit string is easily reproducible (i.e. noise is corrected with high probability).

It is important that the final output of the extraction process is sufficiently long, if not the NUPO system will not be able to distinguish between a large number of objects. This leads to an unusually high FAR. In addition, if the robustness to noise of the final bit string is not sufficient, then there can be no guarantee that the extracted string can be easily reproduced. The vulnerability associated with choice of error correcting technique is an intrinsic vulnerability.

We compare various techniques of correcting the error in a bit string whose bits have two different bit error probabilities in Chapter 7.

It has been observed that over time slight changes/drifts occur in the NUPO measurement data. When this happens, the original produced helper data is not able to correct the noise in the NUPO measurement any more. As a result, the reconstructed bit string does not match the original extracted string, and a valid system user is denied access to the system. For example, in the case of biometrics some characteristics change slowly over time and the biometric system has to employ some techniques to check this to ensure continued usability. One means of achieving this is by renewal. Renewal is the re-enrolment of a user by the provision of a new enrolment template (and a new extracted string) for that individual. A second approach to this problem is some adaptation techniques used to keep the public data in step with the NUPO changes. This is achieved by modifying the helper data (and other public data) so that they continue to yield the same private data (extracted string) with future measurements. We discuss a means of securely modifying the helper data in Section 6.2 of Chapter 6 using sanitizable signatures.

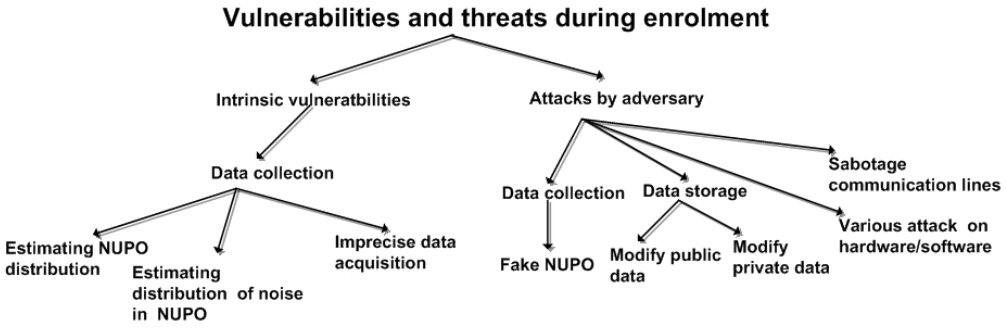

Figure 3.2: Vulnerabilities and threats at enrolment.

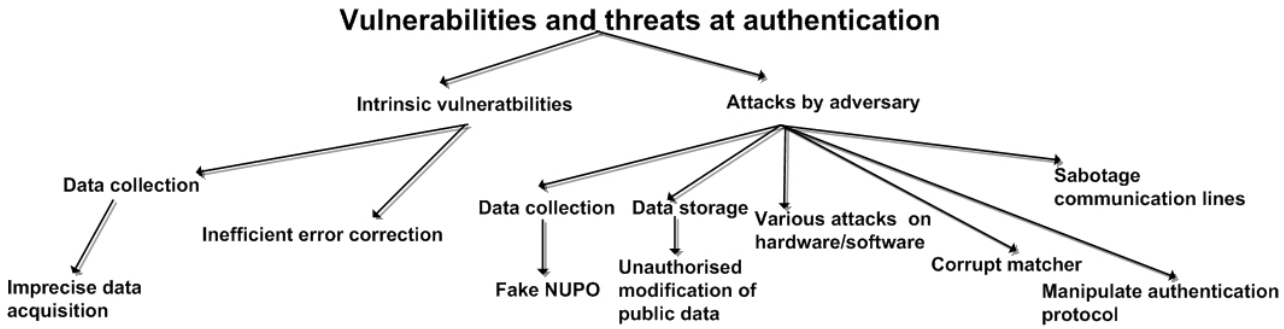

Figure 3.3: Vulnerabilities and threats at authentication.

3.4 Other threats and vulnerabilities

A: Matcher Corrupted/ Decision Overridden

An adversary with access to the internal workings of the system may corrupt the internal functions/mechanisms of the matcher. To ensure that this not happen, the private data storage medium and the matcher should mutually authenticate each other before the matching/comparison procedure.

Overriding the decision of the honest matcher/decision module is a result of an adversary sabotaging the internal communication of the NUPO system. Corrupting/manipulating communication modules is an attack that can be mounted throughout the system, whenever two different modules have to share information. If the security of the communication channels among different system modules cannot be guaranteed, cryptographic means such as encryption, digital signatures and message authentication codes should be used to protect the integrity of the network.

B: Vulnerabilities from environmental factors and extensive usage

Adverse/severe environmental conditions can adversely affect the security and functionality of a NUPO system. The range of environmental conditions (e.g. temperature, pressure, humidity) within which the equipment comprising the NUPO system function effectively is usually stated in the manual of such machines. Care should be taken to ensure that these machines function in the recommended environments. This vulnerability is intrinsic.

With time machines wear out and thus systems achieving a high level of security like NUPO

systems should be closely monitored, components changed when the need arises and carefully maintained.

C: Replay Attacks

Data related to previously authenticated NUPO may be captured and replayed. Security features likes nounces and time stamps may be used to check replay attacks.

D: Other treats and attacks on stored data in helper data systems include the following,

- A legitimate unique identifier string (in the private data storage) can be replaced by an attacker, if he has access to the storage system.
- A stolen unique identifier bit string can be replayed to the matcher to gain unauthorised access.
- Cross matching between databases. For example, an adversary may find out all the databases to which a particular user is subscribed, if the data bases store the same user biometric. This results in a breach of privacy.

These attacks can be curtailed by ensuring that the private data storage is as secure as possible. In addition, the private data should be stored in a masked form, for example, by encrypting or hashing it, thus preventing an attacker from gaining useful information from it even if he gains unauthorised access to the storage.

3.5 Chapter summary

In this chapter we have identified and discussed threats and vulnerabilities common to a NUPO system. We have differentiated between intrinsic vulnerabilities and attacks by an adversary. Furthermore, we classified treats and vulnerabilities according to the system components/modules which they affect. We give a more detailed problem description and solution to some of the threats and vulnerabilities mentioned here in Chapters 4,5,6 and 7.

Chapter 4

A Formal Description of an Extractor in a Helper Data System

The extractor the most important component of the helper data system, besides the NUPO itself. It extracts a unique, uniformly random noise free bit string from the NUPO. It has the ability to correct noise and reconstruct the original extracted string from future measurements of the NUPO. In this chapter we give a general construction of an extractor in a NUPO system. In particular we give the construction of a fuzzy extractor for continuous sources (NUPOs whose measurement data/response have a continuous distribution). An in-depth analysis of this construction is given by discussing issues such as mutual information between the NUPO measurement and the helper data, behaviour of the NUPO noise and the length of the extracted bit string.

This chapter is organised as follows: in Section 4.1 we give a generic construction of a fuzzy extractor for 1- D continuous sources. In Section 4.2, we discuss privacy issues of the construction. Section 4.3 investigates the length of the extracted string taking into account the system designer's lack of knowledge about the true distribution of the NUPO. Section 4.4 gives a detailed analysis of the shape of the noise on the unit interval (which can be made analogous to the extracted string space) for the construction given in Section 4.1.

4.1 Fuzzy extractors

A fuzzy extractor is a general primitive that allows one to extract a noiseless uniform bit string from a noisy source. It consists of two phases. In a first phase the source is challenged and a bit string as well as some helper data are extracted by means of a probabilistic procedure, usually denoted by Gen. The helper data is usually considered as publicly available data and hence can be observed by an attacker (in a strong attack model, the adversary can also modify the helper data, so the helper data needs to be authenticated). In the second phase, the extracted string is reconstructed from a fresh measurement of the noisy source. When the source is challenged under the same circumstances as during the first phase a noisy response is obtained which is slightly different from the one obtained in the first phase. The Rep procedure takes as input a fresh measurement of the source together with the helper data and reconstructs the original extracted string, if the fresh noisy measurement and the original NUPO measurement are similar.

In this section we give a construction of a fuzzy extractor for NUPOs that produce data that have a continuous distribution. We have chosen to give the example because most NUPO produce continuous data. Fingerprint templates for instance are represented by sequences of points in a continuous domain such as $\mathbb{R}$ and $\mathbb{R}^n$. Speckle patterns, the response from optical PUFs and capacitance measurements originating from coating PUFs are typically continuous data [24].

In order to extract a bit string from these continuously distributed measurements some quantisation

step turning them into discrete data has to be performed. The choice of this quantisation procedure is relevant since it determines the quality of the input for all the discrete procedures that follow. When such a quantisation procedure is not carefully chosen much entropy might be lost and only a small amount of bits extracted[24].

We present a geometric construction of a fuzzy extractor for continuous distributions. A pair $\{\mathcal{A}, \mathcal{B}\}$ of partitions of the underlying measurement space is used in the construction. The error correction and security properties of the extractor are formulated in terms of these partitions.

A major part of the work in this section first appeared in the manuscript[24].

Throughout this section, except when otherwise stated, log is taken to base 2. When an algorithm or a function f is randomised, we use the semicolon when we wish to make the randomisation explicit: i.e. we denote by $f(x; v)$, the outcome of computing f on input x with randomness v. $(\mathcal{M}, d)$ will represent a discrete metric space and $(\mathcal{X}, d)$ a continuous metric space [1]. We give some relevant definitions before proceeding with the construction.

Definition 4.1.1 *Let X and Y be random variables on a discrete space $\mathcal{M}$, with probability distributions P and Q respectively. The total variation distance (statistical distance) between P and Q is given as:*

$$\Delta(P, Q) = \frac{1}{2} \sum_{x \in \mathcal{M}} |P(x) - Q(x)|.$$

We often write $\Delta(X, Y)$ or $\Delta(P_X, P_Y)$ instead of $\Delta(P, Q)$ for $X \sim P$ and $Y \sim Q$.

Definition 4.1.2 *Let X be a random variable on a discrete space $\mathcal{M}$ with probability distribution P. The min-entropy of X is given by*

$$H_\infty(P) = -\log(\max_{x \in \mathcal{M}} P(x)).$$

For $X \sim P$, we often write $H_\infty(X)$ or $H_\infty(P_X)$ instead of $H_\infty(P)$. A random variable X with min entropy m is called an m-source.

Definition 4.1.3 *The conditional min entropy of X given Y is given by*

$$H_\infty(X|Y) = -\log \max_{x,y} \Pr[X = x | Y = y].$$

Definition 4.1.4 [1] *The average min-entropy of X given Y is*

$$\widetilde{H}_\infty(X|Y) = -\log \left(\mathbb{E}_{y \leftarrow Y} 2^{-H_\infty(X|Y=y)} \right).$$

Definition 4.1.5 [1] *Let $m, \widetilde{m}, t > 0$. Let $(\mathcal{M}, d)$ be a discrete metric space. An $(\mathcal{M}, m, \widetilde{m}, t)$-secure sketch is a pair of randomised procedures SS and Rec, satisfying the following:*

$\mathsf{SS} : \mathcal{M} \to \{0,1\}^* : x \mapsto s.$
$\mathsf{Rec} : \mathcal{M} \times \{0,1\}^* \to \mathcal{M} : (x', s) \mapsto \hat{x}.$

1. *Correctness: If $d(x, x') \leq t$, then $\mathsf{Rec}(x', s) = x$.*
2. *Security: For any random variable X over $\mathcal{M}$ with min entropy m, $\widetilde{H}_\infty(X|SS(X)) \geq \widetilde{m}$.*

Definition 4.1.6 [1] *Let $\ell, \epsilon, t, m > 0$. Let $(\mathcal{M}, d)$ be a discrete metric space. An $(\mathcal{M}, m, \ell, t, \epsilon)$-fuzzy extractor consists of a pair of randomised procedures, generate (Gen) and reproduce (Rep) satisfying the following:*

$\mathsf{Gen} : \mathcal{M} \to \{0,1\}^\ell \times \{0,1\}^* : x \mapsto (p, r).$
$\mathsf{Rep} : \mathcal{M} \times \{0,1\}^* \to \{0,1\}^\ell : (x', p) \mapsto \hat{r}.$

[1] We abuse notation by setting the same notation d for the metric in both $\mathcal{M}$ and $\mathcal{X}$. The precise meaning will however be clear from the context.

1. *Correctness: For x, x' with $d(x, x') \leq t$, we have $\mathsf{Rep}(x', p) = r$.*

2. *Security: For any distribution on $\mathcal{M}$ with min entropy m, we have $\Delta(\langle R, P\rangle, \langle U\ell, P\rangle) \leq \epsilon$, where U_ℓ is a uniformly distributed random variable on $\{0,1\}^\ell$.*

For examples of fuzzy extractor constructions in the Hamming distance metric, set-difference metric and edit distance metric, the interested reader is referred to [1].

Extractors are functions which take an m-source input and some random seeds (coins) and output nearly uniform bits.

Definition 4.1.7 [1, 33] *A strong (ℓ, m, ϵ)-extractor on the set $\mathcal{M}$, is a function $f : \mathcal{M} \times \{0,1\}^\tau \to \{0,1\}^\ell$, such that for any random variable X on $\mathcal{M}$ satisfying $H_\infty(X) \geq m$ and U_τ uniformly distributed over $\{0,1\}^\tau$, $\Delta(f(X, U_\tau)U_\tau, U_\ell \times U_\tau) \leq \epsilon$.*

Fuzzy extractors can be constructed from secure sketches and strong extractors. The secure sketch enables the exact reconstruction of the noisy input while the strong extractor extracts uniform randomness from the input data.

For continuously distributed sources, a quantisation scheme $\mathcal{Q}$ is applied to transform the continuous domain to a discrete domain. A fuzzy extractor for discrete domains is then applied. The fuzzy extractor of choice in our construction consists of a secure sketch and a strong extractor.

During reconstruction (Rep), the discretized version of the measurement data is reconstructed instead of the original x in the continuous domain. $\mathcal{Q}(X)$ is treated as the "discrete original"[8]. The entropy loss in this phase of the construction is given by $H_\infty(\mathcal{Q}(X)) - H_\infty(\mathcal{Q}(X)|P)$. $H_\infty(\mathcal{Q}(X)|P)$ is called the left-over entropy [8]. We aim at maximising left-over entropy because it is the "source entropy" for the strong extractor phase. A strong extractor is then applied to $\mathcal{Q}(X)$ to extract a secure bit string. The total entropy loss of the fuzzy extraction scheme using this construction is

$$(H_\infty(\mathcal{Q}(X)) - H_\infty(\mathcal{Q}(X)|P)) + (H_\infty(\mathcal{Q}(X)|P) - \ \ell) = H_\infty(\mathcal{Q}(\mathcal{X})) - \ell,$$

where ℓ is the length of extracted string.

The fuzzy extractor construction we describe in this section has the property that $H_\infty(\mathcal{Q}(X)) = H_\infty(\mathcal{Q}(X)|P)$, i.e. the helper data does not reveal any information about the extracted string. Furthermore, we have that the output after quantisation is uniform, so we do not require a strong extractor. Our construction is optimal when the true NUPO distribution is known. When the distribution is estimated, as is with most practical situations, things are slightly different.

4.1.1 Construction for continuous distributions

Background

Suppose we want to extract a uniform bit string from a noisy source whose data measurements live in the continuous space $\mathcal{X}$. Because the source is continuous and the bit string to be extracted lives in a discrete space, we are forced to discretise/quantise the continuous source. Discretising $\mathcal{X}$ usually involves partitioning. There are many ways to partition a continuous space. In fact the level sets σ_k of any measurable function $h : \mathcal{X} \to \{0,1\}^n$, $\mathcal{A}_{\sigma_k} = \{x \in \mathcal{X} : h(x) = \sigma_k\} = h^{-1}(\sigma_k)$, where $\sigma_k \in \{0,1\}^n$ and $\mathcal{X} = \bigcup_k \mathcal{A}_{\sigma_k}$, give such a partitioning. Partitioning can be either deterministic or randomised. To simplify notation/exposition, we discuss partitioning schemes that are deterministic.

Notation: Throughout this section we denote the helper data as j instead of p, to distinguish it from the P commonly associated with probability. In the same light, i is the extracted string instead of r.

Construction

We define a surjective function, $\mathcal{Q}$ on $\mathcal{X}$. $\mathcal{Q} : \mathcal{X} \to \{1, \ldots, n\} : x \mapsto i$. i is the extracted string and $\mathcal{Q}$ is called the quantiser. In most practical settings a Gray code is applied to i to obtain its binary representation. Here we use i as the string instead of its binary representation to keep our work simple and easily tractable.

$\mathcal{Q}$ defines a natural partitioning $\mathcal{A}$ of $\mathcal{X}$, which consists of n subsets $\{A_1, \ldots, A_n\}$,

$$A_i = \{x \in \mathcal{X} : \mathcal{Q}(x) = i\},$$

satisfying $A_i \cap A_k = \emptyset$ for $i \neq k$ and $\cup_{i=1}^{n} A_i = \mathcal{X}$.

To explicitly state the relationship between the quantiser $\mathcal{Q}$ and the partition $\mathcal{A} = \{A_1, \ldots, A_n\}$, we write $\mathcal{Q}_{\mathcal{A}} : \mathcal{X} \to \{1, \ldots, n\}$, $\mathcal{Q}_{\mathcal{A}}(x) = i$ if and only if $x \in A_i$. $\mathcal{Q}_{\mathcal{A}}$ induces a discrete probability distribution $P_{\mathcal{A}}$ on $\{1, \ldots, n\}$, $P_{\mathcal{A}} = (P(A_1), \ldots, P(A_n))$.

We incorporate a noise correction mechanism, because the measurement data is noisy. This is done by means of another partition $\mathcal{B}$ of size m, of $\mathcal{X}$. For the two partitions $\mathcal{A} = \{A_i\}_{i=1}^{n}$ and $\mathcal{B} = \{B_j\}_{j=1}^{m}$ of $\mathcal{X}$, the refinement of $\mathcal{A}$ and $\mathcal{B}$ is a partition consisting of the sets $\{A_i \cap B_j\} \forall i, j$. Its associated quantiser, $\mathcal{Q}_{(\mathcal{A},\mathcal{B})}$ is given as $\mathcal{Q}_{(\mathcal{A},\mathcal{B})}(x) = (i, j)$ if and only if $x \in A_i \cap B_j$. i is the extracted string and j is the helper data.
On input of a noisy observation $x' = x + e$ and helper data j, i is recovered as follows

$$\hat{i} = \arg\min_k \operatorname{dist}(x', A_k \cap B_j) = \arg\min_k \min_{\hat{x} \in A_k \cap B_j} d(x', \hat{x}),$$

where $\operatorname{dist}(U, V) = \min_{u \in U, v \in V} d(u, v)$.
$\hat{i} = i$ if $2e \leq \min_{i,j,k:\, i \neq k} \operatorname{dist}(A_i \cap B_j, A_k \cap B_j)$.
To ensure that the correct i is recovered, we demand that sufficient **gap** exists between each pair of partitions $A_i \cap B_j$, for a fixed j and $i = 1, \ldots n$. Formally, if the noise e in the measurement is such that,
$2e \leq d_{\min} = \min_{i,j,k:\, i \neq k} \operatorname{dist}(A_i \cap B_j, A_k \cap B_j)$, then we can correct all errors.

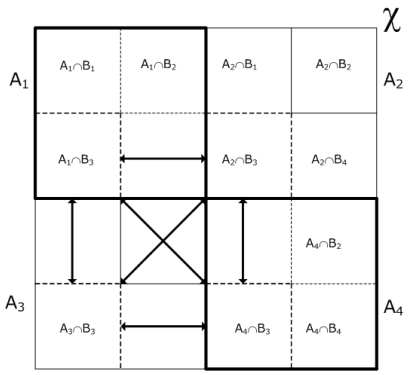

Figure 4.1: For each fixed j, large gaps exist between $A_i \cap B_j$ and $A_k \cap B_j, i \neq k$. This allows for efficient error correction.

It is desirable that the helper data j reveals the least possible information about the extracted string i. To enforce this, we demand that partitions $\mathcal{A}$ and $\mathcal{B}$ are independent, i.e. $P(A_i \cap B_j) = P(A_i)P(B_j) \forall i, j$. This implies that the helper data reveals no information about the extracted string. So $H_\infty(I|J) = H_\infty(I)$

To optimise our construction, we require that the probability distribution induced on $\{1, \ldots, n\} \times \{1, \ldots, m\}$ by the quantiser $\mathcal{Q}_{(\mathcal{A},\mathcal{B})}$ is uniform [2].i.e

$$P(A_i) = \frac{1}{n} \forall i \text{ and } P(B_j) = \frac{1}{m} \forall j.$$

Equivalently, $H_\infty(I) = \log n$ and $H_\infty(J) = \log m$.

To summarise this section we give a formal construction of a continuous space fuzzy extractor as follows:

Let $(\mathcal{X}, d)$ be a continuous metric space. Let $X \in \mathcal{X}$. Let $(\mathcal{A}, \mathcal{B})$ be a pair of partitions of $\mathcal{X}$ of sizes n and m respectively. Let partitions $\mathcal{A}$ and $\mathcal{B}$ be independent. Let $\mathcal{Q}_{(\mathcal{A},\mathcal{B})}$ induce a uniform distribution on $\{1, \ldots, n\} \times \{1, \ldots, m\}$. Let $d_{\min} = \min_{i,j,k:i\neq k} \operatorname{dist}(A_i \cap B_j, A_k \cap B_j)$. Then we define the generation Gen and reproduction Rep procedures of the 1-D continuous space fuzzy extractor as follows:

- Generation: On input $x \in \mathcal{X}$,

  $$\mathsf{Gen}_{(\mathcal{A},\mathcal{B})}(x) = \mathcal{Q}_{(\mathcal{A},\mathcal{B})}(x) = (i, j).$$

  i is the extracted string and j is the helper data.

- Reproduction: On input $x' : d(x, x') \leq \frac{1}{2} d_{\min}$ and j.

  $$\mathsf{Rep}_{(\mathcal{A},\mathcal{B})}(x', j) = \arg\min_k \inf_{\hat{x} \in A_k \cap B_j} d(x', \hat{x}) = i.$$

  With the following properties:

- No entropy loss: $H_\infty(\text{Extracted string}|\text{Helper data}) = H_\infty(I|J) = H_\infty(I)$.
- Security: $\Delta(\langle I, J\rangle, \langle U, J\rangle) = 0$, where U is a uniformly distributed random variable on $\{1, \ldots, n\}$.

4.2 Mutual information between the NUPO measurement and the helper data

In the case of biometrics, the privacy of the original measurement has to be preserved. Therefore it is interesting to investigate how much information is gained about the measurement X from the helper data J.

A natural measure for determining the amount of information about the measurement data that the helper data reveals is the mutual information of the measurement data and the helper data. In terms of partitions, the amount of information an adversary learns about X is given by

$$I(X; J) = H(J) - H(J|X) = H(J) = \log m$$

$H(J|X) = 0$ because given $x \in \mathcal{X}$, an adversary can easily compute $\mathcal{Q}_{(\mathcal{A},\mathcal{B})}(x) = (i, j)$, to determine the value of random variable J.

In order to reveal as little information as possible about X, m has to be chosen as small as possible. However, the choice of m has an impact on the amount of error the scheme can tolerate.

[2] Demanding the partitions be uniform implies that they are independent. However in our construction, independence of the partitions is a more important property, because it ensures that the helper data leaks no information about the extracted string. We treat uniformity of the partitioning as an additional requirement to achieve optimality.

4.3 Length of the extracted string

The construction given in Section 4.1.1 is optimal. The optimality of this construction is based on the fact that the NUPO distribution is known. When the NUPO distribution is known, the designer can choose a suitable pair of partitions $(\mathcal{A}, \mathcal{B})$ that achieves optimality. However, in general, we can not assume that the probability distribution ρ of the NUPO is known precisely. This is due to the fact that in practise one often has to learn the distribution of X and therefore typically obtains only an estimate $\hat{\rho}$ of the distribution ρ. In a strong attack model we allow for the possibility that the attacker puts more effort in learning the distribution ρ and therefore has more accurate knowledge of ρ than the designer. This implies of course that the schemes that we have described do not a priori guarantee security of the extracted string.

The fuzzy extractor in Section 4.1.1 was constructed using a secure sketch and a strong extractor. Because the NUPO distribution was assumed to be known, the right partitions were chosen and consequently the string obtained after discretisation was uniform and so there was no need for the strong extractor. However, applying the chosen partitioning scheme to the true NUPO distribution will neither yield a string that is uniform nor one that is independent of the helper data. This is depicted in Figure 4.2.

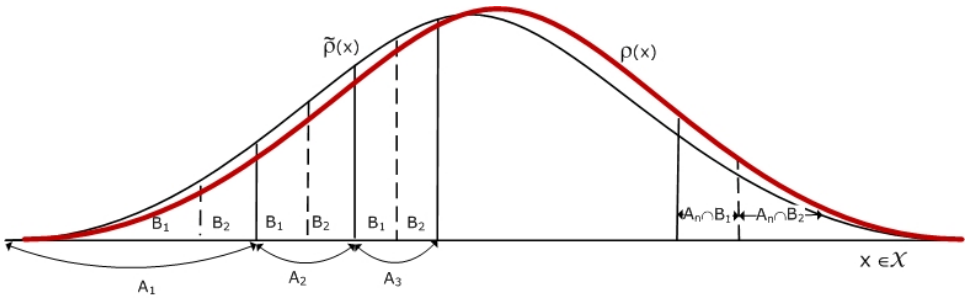

Figure 4.2: Effect of partitioning scheme designed using estimated distribution $\hat{\rho}$, on unknown true distribution ρ. Partitions $\mathcal{A}$ and $\mathcal{B}$ are sizes n and $m = 2$ respectively.

In this scenario, we have to apply a strong extractor to extract a uniform bit string. We make use of the version of the Leftover hash lemma (stated shortly) from [1] to extract an almost uniform bit string. The Leftover hash lemma is a useful tool in cryptography that enables us extract almost $H_\infty(X)$ bits that are almost uniformly distributed from a random variable X. It archives nearly optimal randomness. The Leftover hash lemma is a classical construction of extractors based on pairwise independent hash functions[3].

To be able to apply the Leftover hash lemma, we have to estimate the entropy after discretisation. We estimate the quantity $\tilde{H}_\infty(I|J)$ using the distance between the true and estimated NUPO distributions. The distance measure of choice is the total variation distance.

Let $0 \leq \xi \leq 1$ be a design parameter. The designer produces an estimate $\hat{\rho}$ of ρ (e.g. estimating from a sample population) such that $\Delta(\rho, \hat{\rho}) < \xi$. Next, the designer selects an appropriate partition based extractor scheme for $\hat{\rho}$.

From Section 4.1, we know that when the known approximate distribution $\hat{\rho}$, is used as the distribution on $\mathcal{X}$, $\mathcal{Q}_{(\mathcal{A},\mathcal{B})}$ induces a distribution $\hat{P}$ on $\{1, \ldots, n\} \times \{1, \ldots, m\}$ satisfying the following: Partitions $\mathcal{A}$ and $\mathcal{B}$ are independent with respect to $\hat{P}$, i.e. $\hat{P}(A_i \cap B_j) = \hat{P}(A_i)\hat{P}(B_j)$. Also $\hat{P}$ is uniform over $\{1, \ldots, n\} \times \{1, \ldots, m\}$. $\mathcal{Q}_{(\mathcal{A},\mathcal{B})}$ induces an unknown probability distribution P on $\{1, \ldots, n\} \times \{1, \ldots, m\}$, which follows from using the (unknown) true distribution ρ on $\mathcal{X}$. Let I

[3][33] A family H of functions $h : \{0,1\}^n \to \{0,1\}^\ell$ is pairwise independent if for all distinct $x, y \in \{0,1\}^n$ and for all $a, b \in \{0,1\}^\ell$, $\Pr_h[h(x) = a \text{ and } h(y) = b] = \frac{1}{(2^\ell)^2}$, where the probability is taken over a uniform selection of h from H.

and J be random variables that represent the true marginal distributions on $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$ respectively. Correspondingly, let $\hat{I}$ and $\hat{J}$ be the random variables of the approximate distribution.

$\Delta(\rho, \hat{\rho}) \leq \xi$, so $\Delta(P_{IJ}, P_{\hat{I}\hat{J}}) \leq \xi$. This holds because the total variation distance (statistical distance) is not increased by the application of a function[25].

Using $\Delta(P_{IJ}, P_{\hat{I}\hat{J}}) \leq \xi$, we estimate the average min-entropy[4] $\tilde{H}_\infty(I|J)$.

Lemma 4.3.1 *Let I and J be random variables as described in Section 4.1.1, then*

$$\tilde{H}_\infty(I|J) \geq \log \frac{1}{m\xi + \frac{1}{n}}.$$

Proof
We use the following result from [25].

Lemma 4.3.2 [25] *Let X be a random variable on $\{1, \ldots, n\}$. If $\Delta(P_X, U_n) \leq \delta$, then, $\max_x \Pr[X = x] \leq \frac{1}{n} + \delta$, where U_n is the uniform distribution on $\{1, \ldots, n\}$.*

Lemma 4.3.2 implies that is if $\Delta(P_{IJ}, U_n \times U_m) \leq \xi$ then $P_{IJ}(i, j) \leq \frac{1}{nm} + \xi \ \forall i, j$.

$$\begin{aligned}
\tilde{H}_\infty(I|J) &= -\log\left(\sum_{j\in\{1,\ldots,m\}} \Pr[J = j] \max_i \Pr[I = i|J = j]\right) \\
&= -\log\left(\sum_{j\in\{1,\ldots,m\}} \max_i \Pr[I = i, J = j]\right) \\
&\geq -\log\left(m \max_{i,j} \Pr[I = i, J = j]\right) \\
&\geq -\log m\left(\xi + \frac{1}{nm}\right) \\
&= \log n - \log(nm\xi + 1) \\
&= \log \frac{1}{m\xi + \frac{1}{n}}.
\end{aligned}$$

For a practical fuzzy extractor, $m\xi$ should be of order $O(\frac{1}{n})$ implying that $\tilde{H}_\infty(I|J)$ is of order $O(\log n)$.□

Lemma 4.3.3 (Leftover hash lemma) [11] *If X, Y are random variables such that $X \in \{0, 1\}^\tau$ $\tilde{H}_\infty(X|Y) \geq k$, and $\{H_v\}_{v\in V}$ is a family of pairwise independent hash functions from τ bits to ℓ bits, then,*

$$\Delta((Y, V, H_V(X)), (Y, V, U_\ell)) \leq \epsilon$$

as long as $\ell \leq k - 2\log\frac{1}{\epsilon} + 2$.

Lemma 4.3.3 implies that,

$$\Delta((J, V, H_V(I)), (J, V, U_\ell)) \leq \epsilon \text{as long as } \ell \leq \log \frac{1}{m\xi + \frac{1}{n}} - 2\log\frac{1}{\epsilon} + 2.$$

So we can extract a string of length $\ell \leq \log \frac{1}{m\xi+\frac{1}{n}} - 2\log\frac{1}{\epsilon} + 2$ that is ϵ-close to uniform, when the true and estimated NUPO distributions are ξ apart in total variation distance.

[4] We decided to use the average min entropy $\tilde{H}_\infty(I|J) = \left(\mathbb{E}_{j\leftarrow J} \max_i \Pr[I = i|J = j]\right)$ instead of the min-entropy $H_\infty(I|J) = -\log\max_{i,j} \Pr[I = i|J = j]$ because j is known and only the prediction of I is adversarial. The helper data j is made public, so the adversary does not need to guess its value. The min-entropy is too strict, because it takes the worst-case j, while for randomness extraction and predictability by an adversary, average-case j suffices. See [1] for more details.

4.4 Shape of NUPO noise during the extraction process

The distribution of the noise, like the distribution of the NUPO is estimated by sampling. However, because noise is a major issue to contend with in using NUPOs, it pays not only to know the distribution of noise in $\mathcal{X}$ but also to understand its behaviour during the extraction procedure.

Allowance is made for the measurement noise in the design of the fuzzy extractor depending on how well the behaviour of the noise is known. If the distribution of noise is not learnt well enough or its behaviour during extraction process not fully understood, then with very high probability, very different bit strings will be extracted from 'different measurements of the same NUPO even after error correction.

In the construction in Section 4.1, the fuzzy extractor is designed to extract a uniform bit string from a NUPO. However the distribution of the noise as it is propagated through the extraction system may not be uniform. Knowing the probabilities of all error patterns potentially allows for an efficient choice of error correcting technique, i.e. one that is good at correcting the patterns that occur with high probability.

In this section, we investigate the behaviour of the NUPO noise in the fuzzy extractor. In particular we consider the shape of noise in construction given in Section 4.1. We surmise that this knowledge will enable us in choosing an efficient error correcting technique, because it tells us which error patterns are more likely and which are less likely.

We assume that the enrolment measurement has a Gaussian distribution and is noise free. We also assume the noise to be normally distributed and independent of the source. This assumption mirrors many practical scenarios (many lifeless NUPOs) and is susceptible to complete analysis.

Given is a Gaussian variable X on the measurement space $\mathbb{R}$. The noise $Y \in \mathbb{R}$ is also Gaussian. We map X to the uniform distribution on the unit interval[5] before discretising by dividing into equiprobable intervals (this is the same as discretising $\mathbb{R}$ using equiprobable partitions as is done in Section 4.1.1). We determine the probability distribution and the behaviour of the noise after the noisy measurement $X' = X + Y$ has been mapped onto the unit interval.

Definition 4.4.1 *Let X be a random variable with density function f_X. X is mapped to the unit interval by the function F defined as,*

$$F : \mathbb{R} \to [0, 1] : x \mapsto \int_{-\infty}^{x} f_X(t)dt.$$

Lemma 4.4.2 *Let F be the function in Definition 4.4.1. Then $F(X)$ is uniformly distributed on* $[0, 1]$.

See Appendix A for proof.

Corollary 4.4.3 *Let X be a random variable with density function $f_X = \mathcal{N}(\mu, \sigma)$. If X is mapped to random variable Z on* $[0, 1]$ *by the function F in Definition 4.4.1, then,*

$$z = F(x) = \frac{1}{2}\left(1 + \text{erf}\frac{x - \mu}{\sqrt{2\sigma^2}}\right).$$

$$x = F^{\text{inv}}(z) = \mu - \sigma\sqrt{2}\text{erfc}^{\text{inv}}(2z).$$

The function F maps X and X' into random variables Z and Z' respectively, on the unit interval. We will compute the density function of Z and Z' and also their joint density function. On $\mathbb{R}$, the noise does not depend on X, however after mapping X' to $[0, 1]$, the noise properties will depend on Z. As $X' - X$ is the noise on the source space, $Z' - Z$ is the noise on the unit interval. However, while the noise $Y = X' - X$ on the source space is Gaussian, we cannot immediately draw any conclusions

[5] Any 1-D distribution can be mapped to the uniform distribution on the unit interval, using a function defined shortly.

about the corresponding noise in unit interval, which is $\delta_z = z' - z$, for any fixed z. Computing the distribution parameters of the random variable $Z' - Z$ is not an easily tractable problem. To obtain information about the distribution of the noise on the unit interval we compute the joint probability density function $f_{ZZ'}$, of Z and Z'. From this quantity we derive the conditional density function $f_{Z'|Z=z}$ of Z' given that $Z = z$.

We formulate our results in terms of the following lemmas and theorems, their proofs can be found in Appendix A.

Theorem 4.4.4 *Let $X \in \mathbb{R}$ be a random variable with density function f_X. Let $Y \in \mathbb{R}$ be the random variable representing the noise in measuring X. Y has probability density f_Y. X and Y are independent. Let $X' = X + Y$ be the noisy version of X. X' has density function $f_{X'}$.*

Let X and X' be mapped to random variables Z and Z' respectively, on the unit interval by the function F, (Definition 4.4.1). Let $f_{ZZ'}$ be the joint probability density function of Z and Z'. Then,

1. $f_Z(z) = 1$.
2. $f_{Z'|Z=z}(z', z) = f_{ZZ'}(z, z')$.
3. $f_{Z'}(z') = \frac{f_{X'}(F^{\text{inv}}(z'))}{f_X(F^{\text{inv}}(z'))}$.
4. $f_{ZZ'}(z, z') = \frac{f_Y(F^{\text{inv}}(z') - F^{\text{inv}}(z))}{f_X(F^{\text{inv}}(z'))}$.

Corollary 4.4.5 *Let X, Y, Z, Z' and F be as defined in Theorem 4.4.4. Furthermore let $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ and $Y \sim \mathcal{N}(0, \sigma_y^2)$. Let $\frac{\sigma_x}{\sigma_y} = c$. Then,*

1. $f_{Z'}(z') = \frac{c}{\sqrt{1+c^2}} e^{\frac{(\text{erfc}^{\text{inv}}(2z'))^2}{1+c^2}}$.
2. $f_{Z'|Z=z}(z, z') = f_{ZZ'}(z, z') = ce^{-c^2\left(\text{erfc}^{\text{inv}}(2z) - \text{erfc}^{\text{inv}}(2z')\right)^2 + (\text{erfc}^{\text{inv}}(2z))^2}$.

Observations

The density function f'_Z and the joint density function $f_{ZZ'}$, depend only on $c = \frac{\sigma_x}{\sigma_y}$. The mean μ_x, of the original noise free random variable X, does not affect the shape of the noise on the unit interval.

In $\mathcal{X}$ the noise is independent of the NUPO measurement data X. However on the unit interval, the distribution of Z' the noisy version depends heavily on the value of Z. The relationship between the two is given in Theorem 4.4.4 for general distributions and Corollary 4.4.5 for Gaussians.

Interpretation of result

Figures 4.3 and 4.4 give the graphs of f'_Z and $f_{ZZ'}$ when the variance of the noise (σ_y^2), is small and when it is large.

The main result of this section is Eqn.(2) of Corollary 4.4.5. We simplify this equation by replacing z' with its Taylor series expansion.

$z' = z + \delta_z$, with $|\delta_z| \ll 1$.$\text{erfc}^{\text{inv}}(2z') = \text{erfc}^{\text{inv}}(2(z + \delta_z))$.

We do a Taylor series expansion of $\text{erfc}^{\text{inv}}(2(z + \delta_z))$ about $\delta_z = 0$ (ignoring δ_z^k for $k \geq 4$). Next, we substitute the result into $f_{ZZ'}$ and simplify. This results in,

$$f_{ZZ'}(z, z + \delta_z) \quad = \quad c\left(\exp^{\left(\text{erfc}^{\text{inv}}(2z)\right)^2} \exp^{-\pi c^2 \delta_z^2 e^{2\left(\text{erfc}^{\text{inv}}(2z)\right)^2}} \exp^{-2\pi^{\frac{3}{2}} c^2 \delta_z^3 e^{3\left(\text{erfc}^{\text{inv}}(2z)\right)^2}} \exp^{O(\delta_z)^4}\right)$$

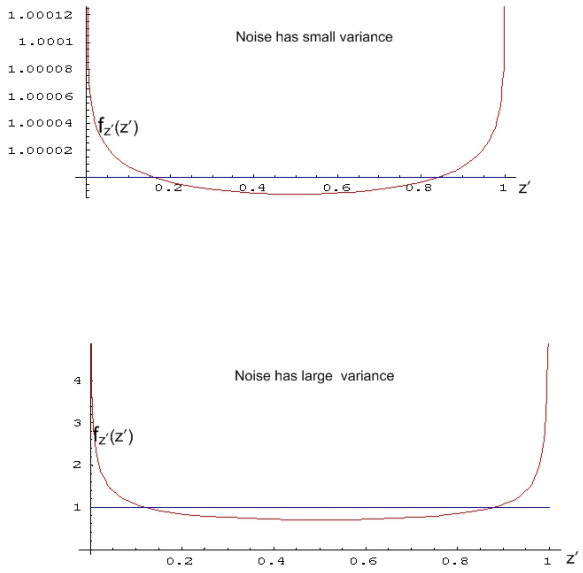

Figure 4.3: $f_{Z'}$ when $c = 200$ (noise has small variance) and when $c = 1$ (noise has large variance).

From the equation above, we see that the width of the noise distribution on $[0, 1]$ depends on Z. The term $\exp^{-\pi c^2 \delta^2 e^2 \left(\text{erfc}^{\text{inv}}(2z)\right)^2}$ gives it a Gaussian-like shape, while $\exp^{-2\pi^{\frac{3}{2}} c^2 \delta^3 e^3 \left(\text{erfc}^{\text{inv}}(2z)\right)^2}$ gives the "non-Gaussian" deformation of the noise. As the strength of the noise increases, the stronger its non-Gaussian deformation.

Plotting $f_{ZZ'}$ as a function of Z and Z' (see Figure 4.4) is informative but more difficult to read and interpret than a 1D plot. To derive some information about the behaviour of $f_{ZZ'}$ we make the following 1D plots.

- We observe $f_{ZZ'}$ for a fixed value of Z (for $Z = z$ we observe the graph of $f_{ZZ'}$ against Z').
    1. Error has a small statistical dispersion (σ_y is small): The resulting graph is a highly peaked narrow Gaussian curve symmetric about the point $Z' = z$. See Figure 4.5. This holds for all values of z. This implies that the extractor has a very high probability of outputting the correct value after noise correction. The actual probability of outputting the correct string depends on the width of the curve. This width is narrowed with error correction using a somewhat "mild" helper data.
    2. Error has a large statistical dispersion (σ_y is large): For fixed $Z < 0.5$, the resulting graph is positively skewed – more probability mass on the right than would be expected of a regular normal distribution . For $Z = 0.5$, the graph is Gaussian symmetric about $Z' = 0.5$ while for $Z > 0.5$, the resulting graph is negatively skewed. See Figure 4.6. The large spread of the curves show that a "very strong" helper data is needed to reduce the widths of these curves to be able to reconstruct the correct value z with high probability.

4.5 Chapter summary

In this chapter we gave a generic construction of fuzzy extractors for NUPOs which produce data that have a continuous distribution. The construction presented here works well on discrete spaces.

Based this construction, we highlight and carry out an in-depth study of various concerns in the extraction procedure. In Section 4.2, we considered the issue of privacy. We computed the mutual information between the NUPO measurement data and the helper data. In Section 4.3, we computed

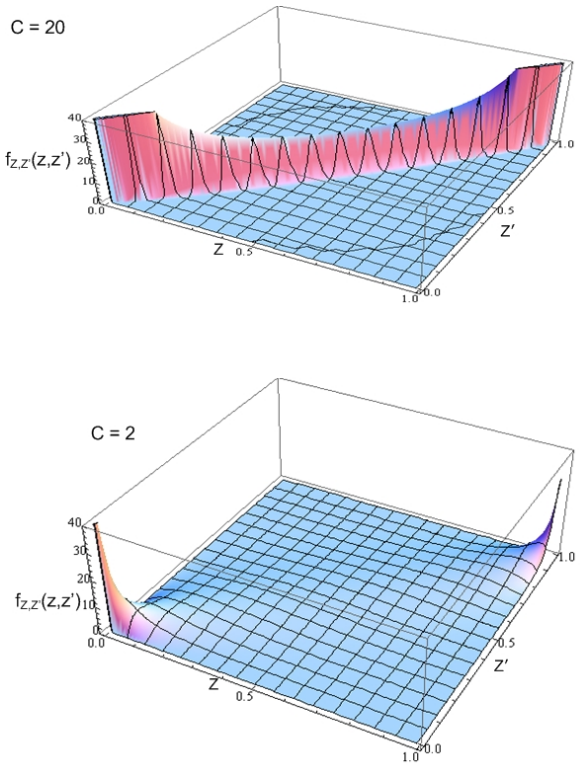

Figure 4.4: $f_{ZZ'}$ when $c = 20$ and when $c = 2$.

the length of the bit string extractable taking into account the designer's lack of knowledge about the distribution of the NUPO. Finally, the shape of the noise in the fuzzy extractor construction in Section 4.1, is treated in Section 4.4. This can help the designer to actually choose an efficient error correcting technique since he knows exactly which error patterns are more likely and which are less likely.

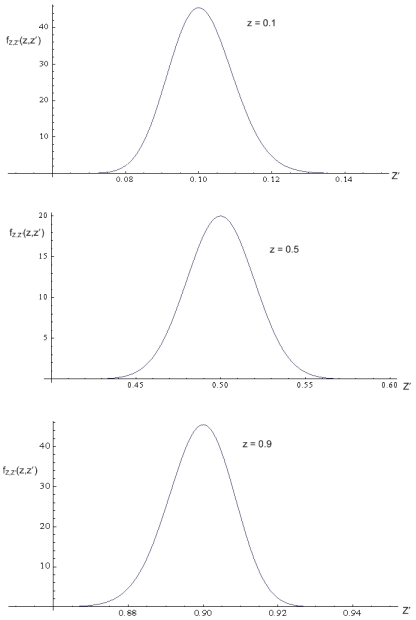

Figure 4.5: $f_{ZZ'}(z, z')$ against z' with parameters: $c = 20$, $z = 0.1, 0.5$ and 0.9.

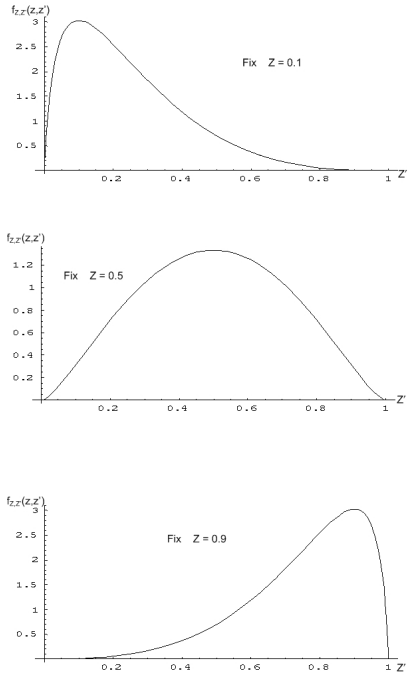

Figure 4.6: $f_{ZZ'}(z, z')$ against z' with parameters: $c = 1.333$, $z = 0.1, 0.5$ and 0.9

Chapter 5

Estimating the Distribution of the NUPO

In most instances, the designer of the NUPO system does not know the true distribution of the source for bit extraction and so either makes assumptions about the underlying probability distribution of the source or statistically estimates this distribution based on some finite set of experimental data.

The mismatch between the real and estimated distributions has a significant impact on the security of the extracted bit string. The significance of taking the appropriate sample size in order to achieve a strong notion of security, cannot be overemphasised. In the strong attack model, we assume that the attacker has more resources at his disposal and so is able to perform more measurements on the source than the designer. This implies that she has a more precise estimate of the real distribution than the designer, and therefore has better knowledge of the distribution of the extracted string. If the estimate of the adversary is significantly better than that of the NUPO system designer, then it is not safe to use the bit strings output from such a system for applications requiring strong security.

Intuitively, the larger the finite set of experimental data from which the NUPO distribution is estimated the nearer the empirical distribution is to the true distribution (this is explained by the law of large numbers and the central limit theorem[34]). In this chapter, we consider the relationship between the distance between the empirical and true distribution and the sample size. The distance measure of choice was the total variation distance (also called the statistical distance). However, we found that we could not arrive at an informative relationship between the two (statistical distance and sample size) and so decided to consider a related distance measure, the ℓ_2 distance. Various inequalities exist relating the two distance notions.

Let $D_{P,\hat{P}}$ be the square of the ℓ_2 distance between P and $\hat{P}$, the true and estimated NUPO distribution respectively. We derive a closed formula for $D_{P,\hat{P}}$ in terms of sample size N in Section 5.1. In Section 5.2, we compute probabilistic bounds relating $D_{P,\hat{P}}$ and some security parameter to the sample size N. In particular, we determine how large N must be chosen to ensure that the distance between P and $\hat{P}$ is below a certain threshold value with high probability.

5.1 Distance between the real and empirical NUPO distributions

Definition 5.1.1 *Let X and Y be random variables on a discrete metric space $\mathcal{M}$, with probability distributions P and Q respectively. Let $p_x = \Pr[X = x]$, and let $q_x = \Pr[Y = x]$. The ℓ_2 distance between P and Q is given by*

$$\| P - Q \|_2 = \left(\sum_{x \in \mathcal{M}} (p_x - q_x)^2 \right)^{\frac{1}{2}}.$$

Definition 5.1.2 *Let X be a random variable with alphabet $\mathcal{X}$ and probability mass function $p_x = \Pr[X = x]$, for $x \in \mathcal{X}$.*

- *The expected value of X, denoted by $\mathbb{E}[X]$ is,*

$$\mathbb{E}[X] = \sum_x x p_x .$$

- *The variance of X is,*

$$\mathrm{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 .$$

Given are N measurements of a stochastic variable X with distribution P. An estimate, $\hat{P}$ can be made of the probability distribution P. In the following, we determine the statistical properties (mean, variance) of the distance between P and $\hat{P}$. We consider two distance measures, the total variation distance and the square of the ℓ_2 distance.

The NUPO response is modeled by the the random variable X with probability distribution $P = \{p_1, \ldots, p_k\}$. Given are N measurements of X, from which the empirical distribution $\hat{P}$ is derived .

Let N_i be the random variable indicating the number of times outcome number i was observed over N observations of X. The random variable N_i satisfies $\sum_i^k N_i = N$. Let $\hat{P} = \{\hat{p}_1, \ldots, \hat{p}_k\}$, where $\hat{p}_i = \frac{N_i}{N}, \forall i$.

$\{N_i\}_{i=1,\ldots,k}$ is multinomially distributed, with parameters N and $p = (p_1, \ldots, p_k)$.

Let n_i be the actual value taken by the random variable N_i, then

$$\Pr[N_1 = n_1, \ldots, N_k = n_k] = \begin{cases} \frac{N!}{n_1! \cdots n_k!} p_1^{n_1} \cdots p_k^{n_k}, & \text{when } \sum_{i=1}^k n_i = N; \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned} \mathbb{E}[N_i] &= N p_i \\ \mathrm{Var}[N_i] &= N p_i (1 - p_i) \\ \mathrm{Cov}[N_i N_j] &= -N p_i p_j \quad \text{for } i \neq j \\ \text{Moment generating function, } M_N(t_1, \ldots, t_k) &= \left(\sum_{i=1}^k p_i e^{t_i} \right)^N . \end{aligned}$$

We compute the expectation and variance of each of the following distance measures, $\Delta(P, \hat{P})$ and $D_{P,\hat{P}} = \| P - Q \|_2^2$.

1. $\mathbb{E}\left[\Delta(P, \hat{P})\right]$ and $\mathrm{Var}\left[\Delta(P, \hat{P})\right]$.

   We immediately run into problems when we try to compute the expected value of the total variation distance because,

$$\begin{aligned} \mathbb{E}[\Delta(P, \hat{P})] &= \mathbb{E}\left[\frac{1}{2} \sum_i |p_i - \hat{p}_i| \right] \\ &= \frac{1}{2} \sum_i \mathbb{E}\left[\left| p_i - \frac{N_i}{N} \right| \right] \\ &\neq \frac{1}{2} \sum_i \left| \mathbb{E}[p_i] - \frac{\mathbb{E}[N_i]}{N} \right| \end{aligned}$$

Without the mean, we cannot compute the variance $\text{Var}\left[\Delta(P, \hat{P})\right]$. This challenge motivated us to consider a similar distance measure, the square of the ℓ_2-norm.

The total variation distance is half of the ℓ_1-norm. We compute the expected value and variance of the quantity $\sum_i (p_i - \hat{p}_i)^2$, which is the square of the ℓ_2-norm. Relationships between the ℓ_1-norm and the ℓ_2- norm are given in Appendix A.

2. $\mathbb{E}\left[D_{P,\hat{P}}\right]$ and $\text{Var}\left[D_{P,\hat{P}}\right]$. We formulate our results in terms of the following theorem.

Theorem 5.1.3 *Let X be a random variable with probability distribution $P = \{p_1, \ldots, p_k\}$, where $p_i = \Pr[X = i]$. Given are N measurements of X, from which the empirical distribution $\hat{P}$, of X is estimated. Let N_i be the random variable indicating the number of times outcome number i was observed over N observations of X. The random variables N_i satisfy $\sum_{i=1}^{k} N_i = N$. Let $\hat{P} = \{\hat{p}_1, \ldots, \hat{p}_k\}$, where $\hat{p}_i = \frac{N_i}{N} \forall i$.*

Let $D_{P,\hat{P}} = \sum_i (p_i - \hat{p}_i)^2$ be a distance measure between P and $\hat{P}$, then ,

1. $\mathbb{E}\left[D_{P,\hat{P}}\right] = \frac{1}{N}(1 - \sum_i p_i^2)$
2. $\text{Var}\left[D_{P,\hat{P}}\right] = \frac{2}{N^3}\left(\sum_i (Np_i^2 - p_i^2 + 4p_i^3 - 2Np_i^3) + (N-3)\left(\sum_i p_i^2\right)^2\right)$.

See Appendix A for the proof.

5.2 How large should the NUPO sample population be?

Determining an appropriate sample size is a standard statistics. Unfortunately, there is no simple, standard or general answer. The answer to this standard question depends on the statistical test being conducted, the specific application and the importance of accuracy in the application. In this section, we tailor a solution to suit our application. We determine how large must N be chosen to ensure that the distance between the empirical and real NUPO distributions is below a certain threshold value with high probability.

The formula for the variance, $\text{Var}\left[D_{P,\hat{P}}\right]$, given in Theorem 5.1.3 is rather unwieldy. We simplify this result in the following corollary, whose proof is given in Appendix A.

Corollary 5.2.1

$$\text{Var}\left[D_{P,\hat{P}}\right] \leq \frac{4}{N^2} \sum_i p_i^2.$$

The Markov's and Chebysev's inequalities readily provide us with tools for deriving an upper bound for the probability that a non-negative function of a random variable is greater than or equal to some positive constant.

Lemma 5.2.2 *[Markov's Inequality] Let X be a non-negative random variable. Let $\alpha \geq 0$, then $\Pr[X > \alpha] \leq \frac{\mathbb{E}[X]}{\alpha}$.*

Corollary 5.2.3 *Let $0 \leq \epsilon \leq 1$. Let $\alpha \geq 0$. If $N \geq \frac{1-\sum_i p_i^2}{\alpha\epsilon}$, then $\Pr[D_{P,\hat{P}} \leq \alpha] \geq 1 - \epsilon$.*

Proof Follows from Markov's Inequality. □

The bound derived using the Markov's inequality is weak (because it only utilizes knowledge of the expected value of the distribution). We use Chebysev's inequality to obtain a better bound.

Lemma 5.2.4 *[Chebysev's Inequality] Let X be a random variable with mean μ and variance σ^2. Let $\kappa \geq 0$, then* $\Pr[|X - \mu| \geq \kappa\sigma] \leq \frac{1}{\kappa^2}$.

Chebysev's inequality can alternatively be stated as $\Pr[|X - \mu| \geq \kappa] \leq \frac{\sigma^2}{\kappa^2}$.

Corollary 5.2.5 *Let $0 \leq \epsilon \leq 1$. Let $\kappa \geq 0$. If $N \geq 2\frac{\sqrt{\sum_i p_i^2}}{\kappa\sqrt{\epsilon}}$, then* $\Pr[D_{P,\hat{P}} \leq \mathbb{E}[D_{P,\hat{P}}] + \kappa] \geq 1 - \epsilon$.

Proof

If $N \geq 2\frac{\sqrt{\sum_i p_i^2}}{\kappa\sqrt{\epsilon}}$, then $\Pr[|D_{P,\hat{P}} - \mathbb{E}[D_{P,\hat{P}}]| \geq \kappa] \leq \epsilon$, from Chebysev's inequality.

$\Pr[|D_{P,\hat{P}} - \mathbb{E}[D_{P,\hat{P}}]| \geq \kappa] \leq \epsilon$, implies that $\Pr[D_{P,\hat{P}} \geq \mathbb{E}[D_{P,\hat{P}}] + \kappa] \leq \epsilon$. Hence we have that $\Pr[D_{P,\hat{P}} \leq \mathbb{E}[D_{P,\hat{P}}] + \kappa] \geq 1 - \epsilon$, if $N \geq 2\frac{\sqrt{\sum_i p_i^2}}{\kappa\sqrt{\epsilon}}$. □

Corollary 5.2.6 *Let $0 \leq \epsilon \leq 1$. Let $\kappa \geq 0$. If $N \geq \frac{2\sqrt{\sum_i p_i^2}}{\kappa\sqrt{\epsilon}} + \frac{1-\sum_i p_i^2}{\kappa}$, then* $\Pr[D_{P,\hat{P}} < \kappa] \geq 1 - \epsilon$.

See Appendix A for the proof.

Observation

In the main result of this Chapter, Corollary 5.2.6, it can be observed that the real probabilities, $p_i; i = \{1, \ldots k\}$ only occur in the form $\sum_{i=1}^{k} p_i^2$, this quantity is the collision probability[1] of the random variable X.

Interpretation of Result

ϵ and κ are systems parameters and they decide how large N should be.

From Corollary 5.2.6, we can infer for example, that if we set $\epsilon = 0.01$ and $\kappa = 0.001$, then $N \geq 10^3(1 - \sum_i p_i^2) + 2 \times 10^4 \left(\sqrt{\sum_i p_i^2}\right)$.

Another example, if we set $\epsilon = 0.0001$ and $\kappa = 0.0001$, then $N \geq 10^4(1 - \sum_i p_i^2) + 2 \times 10^6\sqrt{\sum_i p_i^2}$.

From the above we can deduce that given ϵ and k (as in Corollary 5.2.6), the size of the sample N is inversely proportional to the product $\kappa\sqrt{\epsilon}$. We illustrate with the following example.

Example 5.2.7 *Number of bins used in estimating distribution of random variable X is $k = 32$. Let $\sum_{1=1}^{k} p_i^2 = \frac{1}{32}$. Let security parameters $\kappa = 0.01$ and $\epsilon = 0.01$,*

Then we have that

$$\begin{aligned} N &\geq \frac{2\sqrt{\frac{1}{32}}}{0.01\sqrt{0.01}} + \frac{1 - \frac{1}{32}}{0.01} \\ &= 354 + 97 = 451. \end{aligned}$$

So with probability 0.99, $D_{P,\hat{P}} \leq 0.01$, *when sample size* $N \geq 451$.

[1] The collision probability of a random variable X is the probability that X takes on the same value twice in two independent experiments. Explicitly, for independent random variables X and Y over the same range, the collision probability of X is $\Pr[X = Y]$. For random variable X defined in Section 5.1, we have that $\frac{1}{k} \leq \sum_{i=1}^{k} p_i^2 \leq 1$.

Discussion

To our knowledge, there is no previous work relating the distance between the real and empirical distributions of a random variable and the size of the sample population used in estimating the empirical distribution.

The 3 major parameters in Corollary 5.2.6, are ϵ, κ and $\sum_{i=1}^{k} p_i^2$. ϵ and κ are system security parameters and can be chosen arbitrary. The leading term (makes the most contribution) in the inequality relating the sample size N to security parameters, κ and ϵ is $\frac{2\sqrt{\sum_i p_i^2}}{\kappa\sqrt{\epsilon}}$, and it tells us how the security parameters scale with the sample size.

5.3 Chapter summary

In this chapter derived the statistical properties of the distance between the real distribution P and empirical distribution $\hat{P}$, in terms of the sample size N. The distance measure of preference was the total variation distance, but we found that not lot of information could be derived from this quantity, so we considered a related distance measure, the square of the ℓ_2 norm, $D_{P,\hat{P}}$. We computed the expected value and variance of $D_{P,\hat{P}}$,

$$\mathbb{E}\left[D_{P,\hat{P}}\right] = \frac{1}{N}(1 - \sum_i p_i^2).$$

$$\mathrm{Var}\left[D_{P,\hat{P}}\right] = \frac{2}{N^3}\left(\sum_i (Np_i^2 - p_i^2 + 4p_i^3 - 2Np_i^3) + (N-3)\left(\sum_i p_i^2\right)^2\right).$$

The results of this chapter can aid the NUPO designer in choice of appropriate sample size N. We determined how large must N be chosen to ensure that $D_{P,\hat{P}}$ is below a given threshold value $\kappa \geq 0$, with high probability, $1 - \epsilon$, where $0 \leq \epsilon \ll 1$. This is given in the following,

If $N \geq \frac{2\sqrt{\sum_i p_i^2}}{\kappa\sqrt{\epsilon}} + \frac{1-\sum_i p_i^2}{\kappa}$, then $\Pr[D_{P,\hat{P}} < \kappa] \geq 1 - \epsilon$.

Conversely, given N, the system designer can use the results in Section 5.2 to determine the security parameters ϵ and κ.

Choosing N appropriately leads to an improved estimate $\hat{P}$. An improved estimate $\hat{P}$ implies that more entropy can be extracted from the NUPO, because the distance between the true and estimated distributions affects the length of the extracted string. Other effects of an improved estimate, include reducing false rejections and acceptances and improving overall performance of the system. The design parameters κ and ϵ can be chosen such the adversary has no significant advantage, even if she is able to perform more experiments to derive a better estimate of the real distribution P.

Chapter 6

Securing Helper Data Transmission and Modification

In some applications (for example, biometrics) the helper data is stored on a server that also executes the authentication procedure. In this situation, the helper data need not be transported across some insecure network to the authentication point during authentication. In other applications (e.g. online authentication) however, the helper data is stored on a storage device and is transmitted to the point of authentication when the NUPO needs to be authenticated.

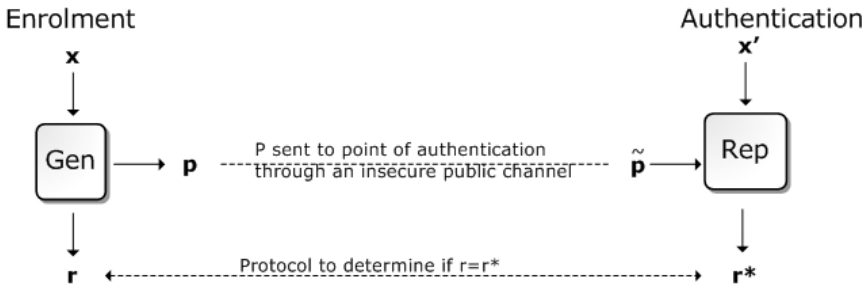

Figure 6.1: Online authentication

The notions of secure sketch and fuzzy extractor defined in Chapter 4, protect against a passive attack, i.e. an attack in which the adversary observes the helper data p and tries to learn something about the measurement data x and the extracted string r. However these definitions, do not provide security in the event that an adversary modifies the sketch/helper data as is sent from its storage to the authentication point. The success of such attacks may lead to the reconstruction of the wrong string. Implying that a legitimate system user is denied system access. Depending on the specific secure sketch/fuzzy extractor in use, an adversary who maliciously alters the public string sent to the authentication protocol may with high probability learn the measurement data and/or the original extracted secure string in its entirety[3].

The motivation for authenticating helper data is immediately clear after considering the following attacks.

- Yes/No decision based on biometrics
  - A typical biometric authentication scenario: A trusted reader takes a biometric measurement. Helper data is transported to point of authentication and extracted bit string is reconstructed. If the reconstructed string matches the stored original extracted string, a "yes" decision is taken, otherwise a "no" decision is taken.

  - Attack 1: Modify the helper data as it is being transmitted. This causes a mismatch between the reconstructed string and the original extracted string. Authentication fails and a legitimate user is falsely denied access to the system.
  - Attack 2: Replace the original extracted bit string with a modified string that matches a completely different object (e.g. the "empty" finger) and modify the helper data accordingly. Upon presentation of the false object, the modified helper data is transmitted to the reconstruction module, these two will yield an extracted string, that matches the corrupted stored string. A similar attack can be mounted on anti-counterfeiting using NUPOs.

- Key storage using integrated NUPOs
  - Attack: Remove part of the physical object, allowing access to the control electronics. This access may allow the attacker to see what the measurement result is of the damaged NUPO. The original extracted string is replaced with a string that is consistent with the damaged NUPO. This attack will lead to a chip that keeps operating, even if the string it reconstructs is totally wrong. This is a typical scenario of the case that "something" goes wrong with the system and because the helper data is unprotected, it is impossible to diagnose "what" went wrong.

The common idea for these attacks is to modify both the helper data and the private data. If the adversary modifies only one of the two, authentication will fail. If the adversary wants to mount a denial-of-service attack, all she needs do is to modify the helper data significantly. When the modified helper data is used in the reconstruction procedure, the string reconstructed will not match the original extracted string. An adversary may also modify the helper data with the intent of gaining information about the measurement data or the extracted string. In this setting, she modifies the helper data in such a way that the reconstructed string yields valuable information about the original extracted string (or the measurement data).

A potential solution to the problem of unaunthorised helper data modification is to make the system user/object store the helper data. Storage can be by memorising or having it stored on a smart card or token. This solution has the disadvantage of having the user store additional cryptographic information. This in someway defeats one of the motivations of using NUPOs (especially biometrics) which is to avoid the need for the user to store any additional cryptographic information (even if the information need not be kept secret).

We surmise that the need for authenticating helper data arises because we cannot guarantee the perfect security of the storage and/or transportation of the helper data (and other public data). In Section 6.1 we present different methods of authenticating the helper data.

It has been observed that there is a natural drift in NUPO responses over time. When this happens, the helper data produced at enrolment can no longer correct the noise in the NUPO measurement. This implies that the reconstructed string is not equal to the original extracted string, and a legitimate system user/object is denied access to the system. In this situation it may be necessary to modify the helper data to keep it in step with the drifting NUPO, so that the original extracted string is reconstructed at authentication. In Section 6.2 we present a secure method of modifying the helper data using Sanitizable signatures.

6.1 Helper data authentication

The method applied in authenticating the helper data depends largely on the helper data system in use. In secure sketches, the NUPO measurement data can be incorporated into the helper data authentication mechanism, while in the case of fuzzy extractors, the measurement data and/or part of the extracted string may be used.

We start by modifying the definitions of the secure sketch and the fuzzy extractor to allow for tamper-evident helper data.

A robust [1] secure sketch/fuzzy extractor protects against the modification of the helper data in a very strong way. The reconstruction Rec (secure sketch)/reproduction Rep (fuzzy extractors) procedure will output with high probability the error symbol $\perp$, whenever the helper data is modified, i.e they will detect with high probability any modifications to the helper data and abort the reconstruction procedure. Formally,

Definition 6.1.1 [?BDK05] *A secure sketch is called well-formed if*

- *Rec may return an element in $\mathcal{M}$ or the error symbol $\perp \notin \mathcal{M}$*
- *$\forall x' \in \mathcal{M}$ and arbitrary $\widetilde{P}$, if $\mathsf{Rec}(x', \widetilde{P}) \neq \perp$ then $d(x', \mathsf{Rec}(x', \widetilde{P})) \leq t$*

Definition 6.1.2 *A fuzzy extractor is called well-formed if*

- *Rep may return an element in $\{0,1\}^\ell$ or the error symbol $\perp \notin \{0,1\}^\ell$*
- *$\forall x' \in \mathcal{M}$ satisfying $d(x', x) \leq t$ and arbitrary $\tilde{P} \neq P$, there exists a negligible $0 \leq \gamma \lll 1$, such that $\Pr[\mathsf{Rep}(x', \tilde{P}) = \perp] \geq 1 - \gamma$.*

Remark In support of robustness of secure sketches and fuzzy extractors against unauthorised helper data modification, the helper data should be forge resistant. Seeing the helper data from a original measurement, it should be difficult for an adversary to construct a modified helper data that will successfully go through the authentication protocol.

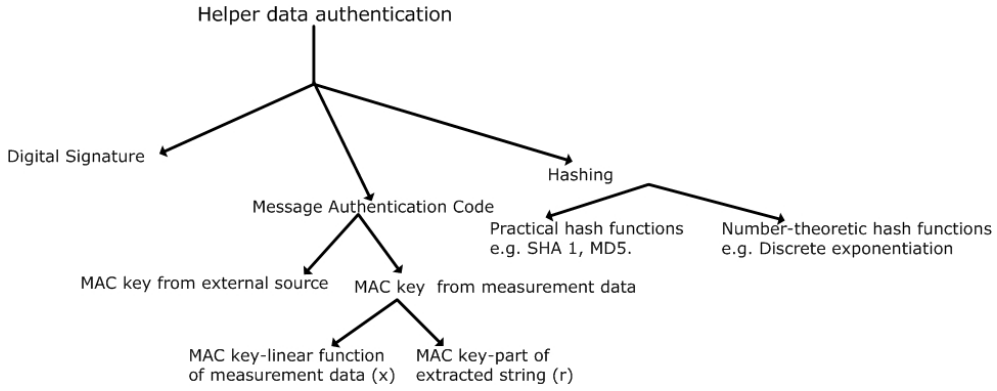

Figure 6.2: Various methods of authenticating helper data.

We give concrete constructions for helper data authentication for the class of fuzzy extractors constructed from secure sketches and strong extractors.

6.1.1 Digital signatures

Let (sk, pk) be the private and public key pair of some trusted party who is to sign the helper data. Let Sign and Verify be the signing and verifying algorithms respectively. SS, Ext and Rec are the secure sketch, strong extractor and reconstruction algorithms respectively. Figure 6.3 provides a method of authenticating helper data using digital signatures.

[1] Robustness of a system has been defined in different ways in the literature, in this chapter robustness implies security against active attacks on the helper data

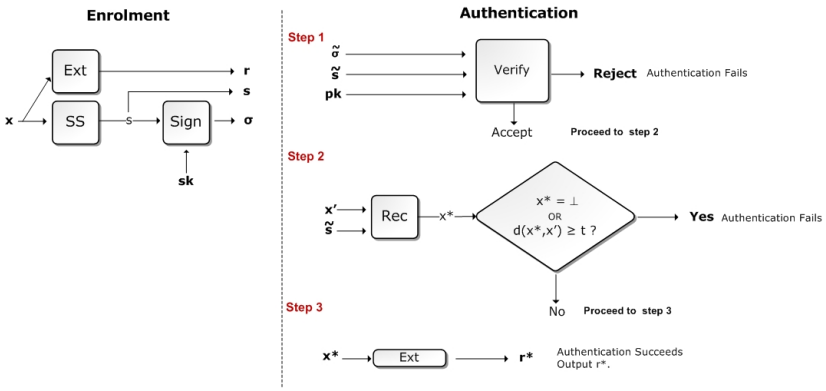

Figure 6.3: Authenticating helper data using digital signatures.

6.1.2 Hashing

Hashing can be used as a means of authenticating helper data. In this subsection we use two different methods of hashing for authenticating helper data. We use the practical hash functions and the number theoretic hash functions.

1. Practical hash functions: Let H be a hash function modeled by a random oracle. Figure 6.4 depicts helper data authentication by means of practical hash functions. This construction can be proven secure in the random oracle model[2][3].

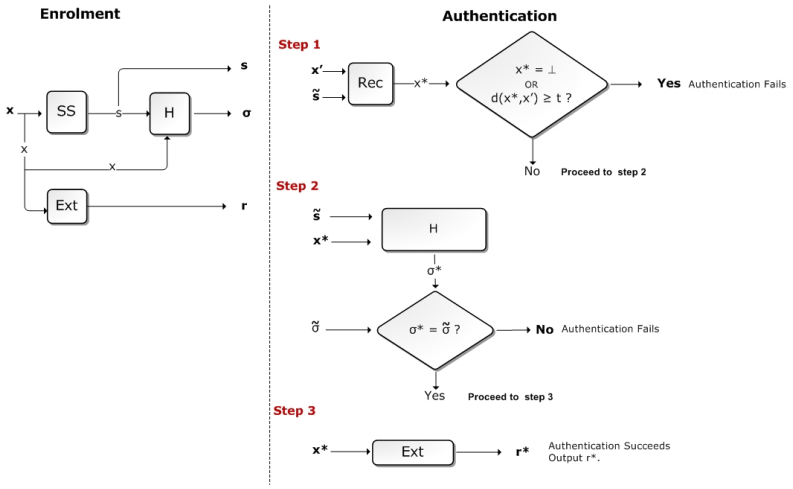

Figure 6.4: Authenticating helper data using practical hash functions.

2. Number-theoretic hash functions (Discrete exponentiation): Let G be a finite cyclic group. Let g be a generator of G, chosen uniformly at random and made public. Let DE be a function performing discrete exponentiation. Figure 6.5 gives a method of authenticating helper data using digital signatures. This helper data authentication mechanism is secure in the standard

[2] In the random oracle model, a cryptographic hash function is viewed as a genuinely random function i.e. a function that responds to every query with a (truly) random response chosen uniformly from its output domain, except that for any specific query, it responds the same way every time it receives that query.

model[3]. To guarantee collision resistance[4], one must ensure that order $G \geq s \geq 0$.

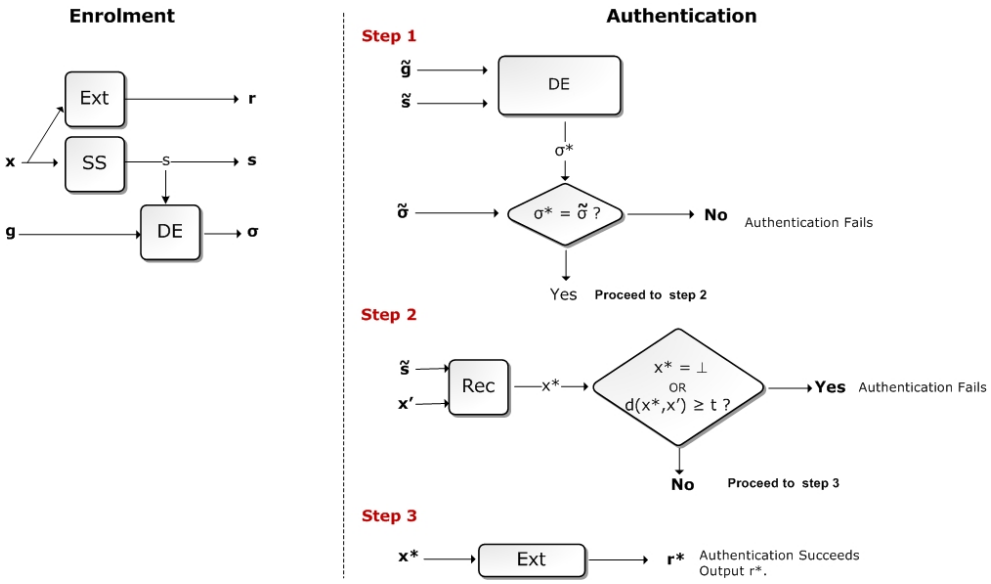

Figure 6.5: Authenticating helper data using discrete exponentiation.

6.1.3 Message Authentication Codes (MAC)

Using a MAC to authenticate the helper data requires a secret key, this key can be generated from the measurement data or the extracted string or input from an external source. In the first solution we present, the MAC key is generated from the measurement data in some linear way. In the second, part of the extracted string is used to key the MAC. We do not give any example of the third method because it is somewhat equivalent to authentication using digital signatures (in both solutions the keys are from an external source, the only difference is that while the MAC is based on symmetric key cryptography, digital signatures are based on public key cryptology).

1. MAC key is a linear function of measurement data: We assume the input x to be an n bit string. Let the secure sketch SS be a linear surjective function. $s \leftarrow \mathsf{SS}(x)$ and s is k bits long. Let $n' = n - k$. There exists a $k \times n$ matrix S of rank k such that $s = \mathsf{SS}(x) = Sx$. Let $S^{\perp}$ be an $n' \times n$ matrix such that the $n \times n$ matrix $\begin{pmatrix} S \\ S^{\perp} \end{pmatrix}$ has full rank. Figure 6.6 gives a method of authenticating helper data using a linear function of the measurement data as MAC key[5]. The security of this construction can be proven in the standard model[5].

2. MAC key is part of the extracted string: We give an example of this construction using a example from [4]. Let Init be a randomized function, which takes no input and outputs a random seed $i \in \{0,1\}^*$ for the extractor Ext. Figure 6.7, gives a method of authenticating helper data using part of the extracted string as MAC key.

   This method is secure in the Common Reference String(CRS)[5] model. Indeed this construction in the CRS model can be likened to a scenario where the public key of some trusted authority is

[3]The proof of the security of a cryptographic system is said to be secure in the standard model, if the proof makes use of only complexity assumptions.

[4]A hash function is collision resistant if it is hard to find two inputs that hash to the same output.

[5]In the common reference string model, all protocol participants have access to a common string that is sampled from a pre-specified distribution (for example the uniform distribution).

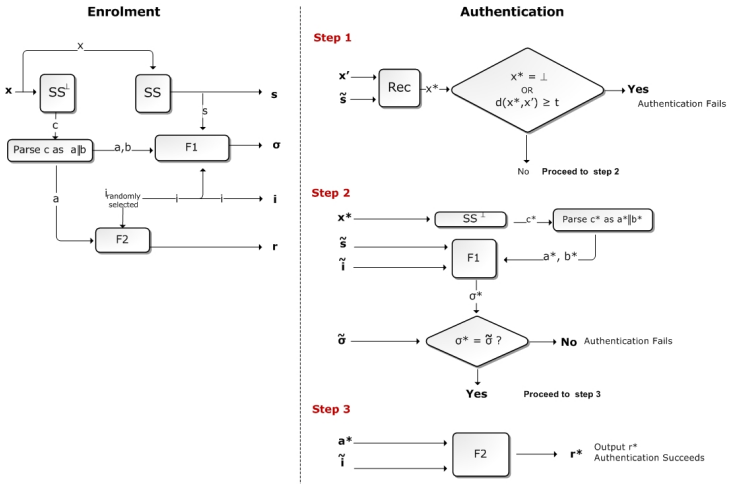

Figure 6.6: Authenticating helper data using a MAC: MAC key is a linear function of NUPO measurement.

used in authenticating the helper data (i.e. a public key is substituted for the common reference string i).

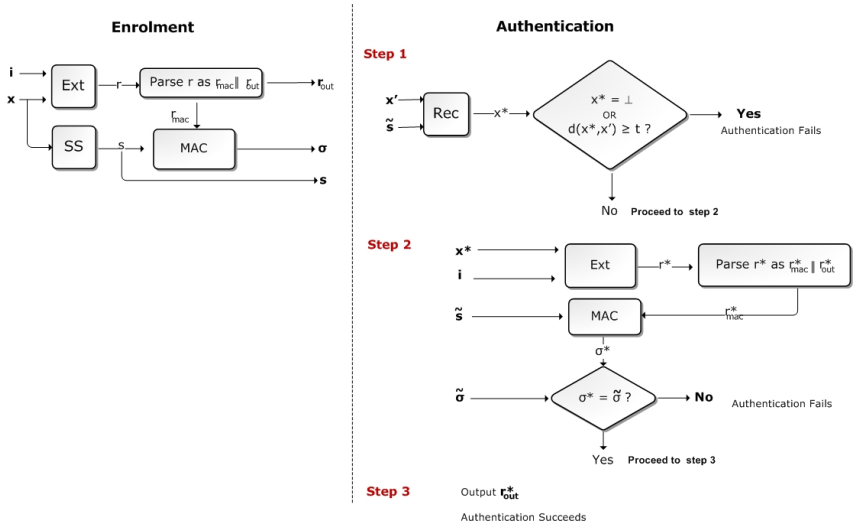

Figure 6.7: Authenticating helper data using a MAC: MAC key is part of the extracted uniform bit string.

Keying the MAC using part of the extracted bit string is more effective than using a linear function of measurement data. The reason for this is that in the former the message authentication code is built using a non-uniform string key, while in the later, the authentication code is keyed by a nearly uniformly random bit string. The necessary inefficiencies of the first method arise because authentication codes keyed by non-uniform randomness imply a non-trivial parameter degradation in the plain (standard) model[4, 26]. In the second instance, the number of bits from the extracted string used as MAC key can be minimised. The interested reader is referred to [4] for more details.

Discussion

The choice of the helper authentication mechanism used depends on the particular application of the helper data system, the amount of security needed, the desired length of the extracted string and the cost one is willing to pay to achieve an efficient helper data authentication mechanism. When the length of the secure bit string extracted is important, the solutions involving the use of signatures, or hash functions or MAC (with externally generated key) can be applied. When one wants to avoid security proofs of the helper data authentication mechanism in the random oracle model, solutions in Figures 6.5, 6.6 and 6.7 are recommended. Using digital signatures as means of authenticating the helper data provides extra security against insider attacks, since there is now a Trusted Third Party(TTP) involved in the authentication mechanism.

Above, we treated solutions for authenticating helper data for the class of fuzzy extractors derived from secure sketches and strong extractors. For fuzzy extractors outside this class, the same general principles apply.

6.2 Secure helper data modification

In the previous section, we discussed mechanisms for authenticating the helper data to detect malicious modification of the helper data, by an adversary. In this section however, we consider a method that allows for the secure modification of helper data by a trusted/semi-trusted censor. We start by giving the motivation for secure helper data modification.

Consider the scenario of a biometric system. An initial measurement x is taken, and helper data p, is produced. Helper data p is supposed to correct all noisy versions x' of the measurement data, as long as $d(x, x') \leq t$. However, it has been observed that the biometric changes slightly over time. As a result, over time future readings of the same biometric deviate more and more from the original measurement x, i.e. $d(x', x) \geq t$ and p is no longer capable of efficiently correcting the error. When this happens to allow for the efficient continual use of the authentication system, either of the following can be done:

1. Re-enrolment: Fix a current instance $x' = \bar{x}$ of the measurement data, compute the helper data $\hat{p}$ and extract $\hat{r}$ based on $\bar{x}$, replace x with $\bar{x}$ as the new original measurement, replace p with $\hat{p}$ and r with $\hat{r}$. Using this solution would entail changing all private and public data associated with that particular NUPO.

2. Using a current instance of the measurement data (and the extracted string) modify p to q, in such a way that future measurements, used with the modified helper data q, will reconstruct the original extracted string r.

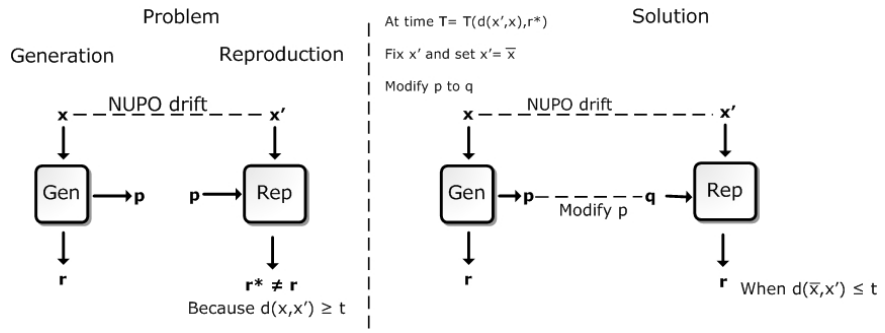

Figure 6.8: Helper data modification as a means of checking drifts in NUPO over time.

The difference between the two proffered solutions is that is that while the first corresponds to a re-enrolment of the same NUPO leading to the production of new helper-data and new private data, the second enables the private data generated in the initial enrolment phase to be continuously valid. The second solution is useful when we take into account the fact that the enrolment authority may not always be available. In particular it allows for delegation of duty and continual use of the NUPO system, in the event that the enrolment authority is not easily accessible. We discuss the second solution in detail in this section.

6.2.1 Sanitizable signatures

We achieve continual usability of NUPO systems by modifying only the helper data. The modifiable helper data system allows for controlled modification by designated parties only. Only a designated censor should be able to modify the helper data such that the output of the reconstruction procedure is not $\perp$.

Security requirements for modifiable helper data systems

- The modified helper data should not reveal substantially new information about the NUPO. Publishing more than one instance of the helper data from the same NUPO may reveal too much information about the NUPO when the different instances are studied together. To prevent this from happening, q should contain but not exhibit new information that was not already present in p.
- Helper data should be modified only by the designated censor. Helper data modified by the censor should be able to pass authentication. To achieve this, the censor possesses a secret that is used in constructing the helper data authentication mechanism.
- Witness hiding: Seeing the old helper-data and the new modified helper-data, an adversary should not be able to compute the censor's secret.
- Whenever an unauthorised entity modifies the helper data, authentication should fail.
- Non-repudiation: Necessary when we assume that censor is semi-trusted.

The Construction

Let $(sk_{\text{sign}}, pk_{\text{sign}})$ and $(sk_{\text{sanit}}, pk_{\text{sanit}})$ be the public/private key pair of the trusted third party who signs the helper data p and the semi-trusted censor respectively. The semi-trusted censor will be allowed to modify parts of the helper data p in a controlled way. That is, the trusted third party should be able to prove (to a court) that the censor modified helper data. We use the notion of sanitizable signatures introduced in [11].

Sanitizable signatures are most commonly implemented using chameleon hashes. A chameleon hash also called a trapdoor commitment has the properties of the regular cryptographic hash function, in particular, it provides collision resistance[6]. However, the owner of the private key sk corresponding to the public key pk used in the hashing algorithm can find collisions. We denote by $\text{CH}_{\text{pk}}(m, v)$, a chameleon hash computed over a message m with randomness v, under public key pk.

- Generation: $\mathsf{Gen}(x) = (r, p)$.

  Split $p = p_1, \ldots, p_\tau$. Specify parts of helper data $p_{i_1}, \ldots, p_{i_k}$ where $\{i_1, \ldots, i_k\} \in 1, \ldots \tau$, that can be modified by the censor with public key pk_{sanit}.

  Select random coins $c_i; i = i_1, \ldots, i_k$.

[6] A hash function is collision resistant if it is hard to find two inputs that hash to the same output.

Compute

$$\sigma_i = \begin{cases} CH_{pk_{\text{sanit}}}(p_i, c_i \parallel i) & \text{if } i = \{i_1, \ldots, i_k\}; \\ \\ p_i \parallel i & \text{otherwise.} \end{cases}$$

Compute signature $\sigma = \text{SIGN}_{sk_{\text{sign}}}(\tau \parallel pk_{\text{sanit}} \parallel \sigma_1 \parallel \ldots \parallel \sigma_\tau)$.

Output (c, p, σ), where c is the concatenation of random coins c_i for $i = i_1, \ldots, 1_k$.

- Reproduction: The inputs to the reproduction procedure are x' and $(\tilde{c}, \tilde{p}, \tilde{\sigma})$.
  - Step 1: Split $\tilde{p}$ into $\tilde{p}_1, \ldots, \tilde{p}_\tau$ and $\tilde{c}$ into $\tilde{c}_{i_1}, \ldots, \tilde{c}_{i_k}$. Compute

$$\tilde{\sigma}_i = \begin{cases} CH_{pk_{\text{sanit}}}(\tilde{p}_i, \tilde{c}_i \parallel i) & \text{if } i = \{i_1, \ldots, i_k\}; \\ \\ \tilde{p}_i \parallel i & \text{otherwise.} \end{cases}$$

$$\text{VERIFY}(pk_{\text{sign}}, pk_{\text{sanit}}, \tilde{p}, \tilde{c}, \tilde{\sigma}(\tilde{\sigma}_1 \parallel \ldots \parallel \tilde{\sigma}_\tau)) = \begin{cases} \text{True} & \text{Proceed to Step 2} \\ \\ \text{False} & \text{Output } \perp \end{cases}$$

  - Step 2: Compute $\mathsf{Rep}(x', \tilde{p}) = r^*$. If $r^* = \perp$, output $\perp$, else output r^*.

- Helper data modification: Since $\sigma_i = CH_{pk_{\text{sanit}}}(p_i, c_i \parallel i)$ for $i = \{i_1, \ldots, i_k\}$ and the censor owns private key sk_{sanit}, for any q_i, $i = \{i_1, \ldots, i_k\}$ he can find random coins d_i such that

$$\sigma_i = CH_{pk_{\text{sanit}}}(q_i, d_i \parallel i) = CH_{pk_{\text{sanit}}}(p_i, c_i \parallel i).$$

The new set of random coins is $d = \{d_i\}$, $i = \{i_1, \ldots, i_k\}$. The modified helper data q is,

$$q = q_1 \parallel \ldots \parallel q_\tau,$$

where

$$q_i = \begin{cases} q_i & \text{if } i = \{i_1, \ldots, i_k\}; \\ \\ p_i & \text{otherwise.} \end{cases}$$

The new signature is the same as the old one σ, because the chameleon hashes do not change.

$$\sigma = \sigma_1 \parallel \ldots \parallel \sigma_\tau.$$

$$\sigma_i = \begin{cases} CH_{pk_{\text{sanit}}}(q_i, d_i \parallel i) = CH_{pk_{\text{sanit}}}(p_i, c_i \parallel i) & \text{if } i = \{i_1, \ldots, i_k\}; \\ \\ p_i \parallel i & \text{otherwise.} \end{cases}$$

The triple, (d, q, σ) is valid and it will pass the authentication test.

As observed in [11], it is noteworthy to mention that not all chameleon hashes are suitable for construction of sanitizable signatures. Using the wrong[7] chameleon hash can lead censor to recover the secret key of the signer (trusted third party). They recommend the use of strongly unforgeable chameleon hashes which are related to the twin Nyberg-Rueppel signatures.

[7] Consider the following chameleon hash defined on (p, c). $CH_{\text{sk}_{\text{sanit}}}(p, c) = y^p g^c$, where $y = g^{\text{sk}_{\text{sanit}}}$ and g is the generator of a prime order cyclic group and sk_{sanit} is the private key. If the original helper data p and random string c is modified into (p^*, c^*), then the private key sk_{sanit} can be recovered as follows: from $g^p y^c = g^{p^*} y^{c^*}$, sk_{sanit} can be computed as $\text{sk}_{\text{sanit}} = \frac{p^* - p}{c^* - c}$.

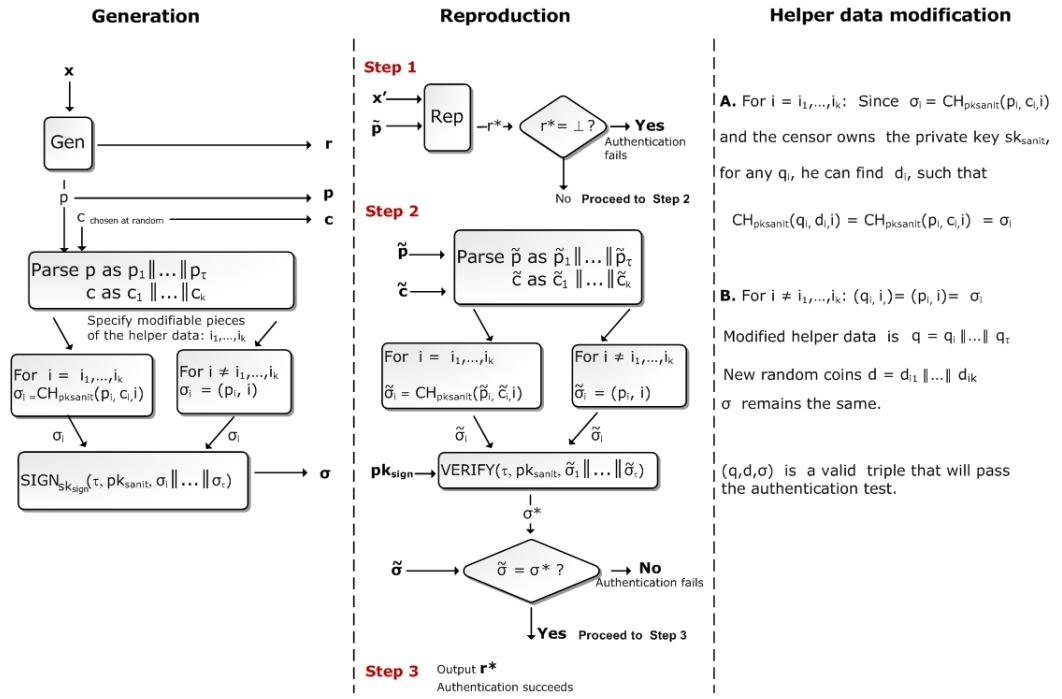

Figure 6.9: Securely modifying helper data using sanitizable signatures

Discussion

As was mentioned in the beginning of this section, secure helper data modification is not the only means of checking natural drifts in NUPO measurements. An intuitive method of checking NUPO drifts is by re-enrolment. The major advantage of our proposed scheme over re-enrolment is that it allows for delegation of duty. Its importance is evident in the situation where the enrolment authority is not readily available. In this situation, delegating the authority to (securely)modify the helper data to a censor who is readily available allows for the continual usage of the NUPO system.

6.3 Chapter summary

In this chapter we have elaborated on the importance of authenticating the helper data as a means of achieving robust security in helper data systems. A consensus for preventing an adversary from compromising the security of the helper data system using the helper data, is that if the helper data is illegally modified and used in the reconstruction of the extracted string, the reconstructed string should fail the authentication test. We have presented and discussed several methods achieving this objective.

In order to accommodate the changes in NUPO over time, it is necessary to consider methods that allow for secure modification of the helper data so that the string extracted from future measurements correspond with the original extracted string. We present a secure method of modifying the helper data using sanitizable signatures. By secure we mean that only an authorised censor is able to modify the helper data and the string reconstructed using the modified helper data will pass authentication.

Chapter 7

Finding the Most Suitable Error Correcting Technique

The chapter is motivated mainly by biometric systems. During enrolment, the NUPO response is converted into a feature vector, from which a binary vector is obtained (see Chapter 2). The process of extracting a feature vector from raw NUPO measurement is called feature extraction[1]. The feature vectors are binarised to produce the unique bit string that would uniquely represent the NUPO. The binarisation of the feature vector is executed by a quantiser. Due to the properties of the fuzzy extractor, the quantiser is constructed such that given two feature vectors that are only slightly different, the corresponding binary vectors are also only slightly different in terms of the Hamming distance. The binarised feature vector string is characterised by bits which have high and varied probabilities of flipping. The binarised feature vector string is reduced to a noise free string of shorter length by means of an error correcting procedure. Because the bits have varying bit error probabilities, the choice of the error correcting technique employed must be chosen with great care.

One may wonder why different bits from the the same NUPO have varying bit error probabilities. There are many reasons why this happens, we give a few below.

- The signal to noise ratio is better for some feature vector components than others. During enrolment, each NUPO is measured repeatedly. Lets assume that M measurements are taken of user A's fingerprint. The features that remain substantially unchanged in all the M measurements are called reliable features/components. These features remain the same inspite of the noise present in measuring the fingerprint M times. The bits extracted from such reliable components can be assumed to have a lower probability of flipping than bits extracted from the other fingerprint features.

- Even if the noise is uniform, binarising the feature vector leads to asymmetry in bit error probability. We explain this by means of the following example.

  Figure 7.1 gives the binary representation of $X \in \mathbb{R}$. If X falls in range "A", its binary representation is 000. Similarly, if it falls in range "B", it is represented by 001, and so on. Consider the following example: X falls into range "B", so its binary representation is 001. We compare the bit error probability of the most significant and least significant bits of the binary representation 001, of X. The least significant bit 1 has a higher probability of flipping than the most significant bit. The least significant bit flips to 0, when X' (a noisy representation of X) falls into range A or range C. However, the most significant bit 0, remains unchanged even if X' falls into range A, C or D.

[1] Feature extraction is a form of dimensionality reduction. Feature extraction involves transforming data into a reduced representation set of features (often called feature vector) as a means of reducing redundancy in the data.

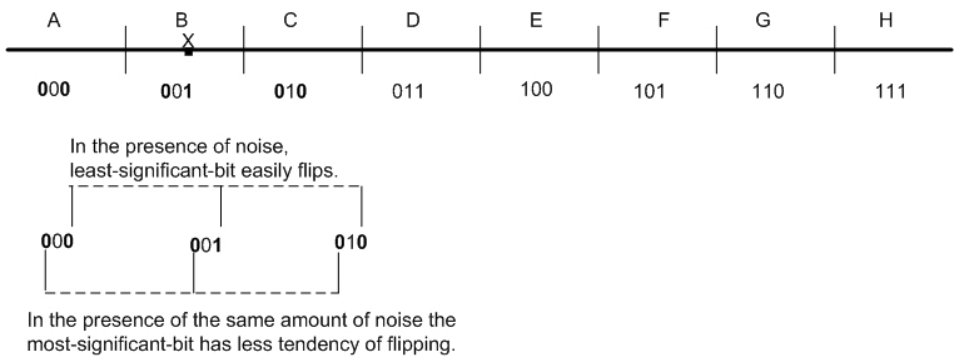

Figure 7.1: Binarisation of feature vectors influences bit error probabilities.

From binarised feature vector string, we extract the shorter noise free string which is used to represent the NUPO. It is desirable that the extracted string is sufficiently long and adequately noise free. The importance of the length of the unique identifier bit string is made apparent by the following loopholes which may occur when the extracted string is not long enough.

- The system will not be able to distinguish between a large number of objects. For instance, extracting ℓ bits from an fingerprint, implies that we can only distinguish between 2^ℓ people. Extracting a short bit string implies high FAR.
- An impersonator can use brute force to determine the unique identifier string for a particular NUPO.

The choice of error correcting technique plays an important role in determining the length of the extracted string. We investigate various error correcting techniques to determine which is most suitable for correcting the error in a string of bits with varying probabilities of flipping. In Section 7.1 we set the stage by carefully defining the problem which we attempt to solve and presenting the various error correcting techniques considered. Section 7.2 compares the listed error correcting techniques.

7.1 Error correcting techniques

Definition 7.1.1 *A (memoryless) channel (X,Y; P) consists of an alphabet X of input symbols, and alphabet Y of output symbols. For each x X and $y \in Y$, $\Pr[Y = y|X = x]$ is the conditional probability that symbol y is received, when symbol x was transmitted. This probability is independent of previous and later transmissions. P is the collection of all such conditional probabilities.*

Definition 7.1.2 *The Binary Symmetric Channel(BSC) is the channel $(X, Y; P)$ with both X and Y equal to $\{0, 1\}$ and P given by $\Pr[Y = 1|X = 0] = \Pr[Y = 0|X = 1] = p_b$ and $\Pr[Y = 0|X = 0] = \Pr[Y = 1|X = 1] = 1 - p_b$.*

Assumptions

- We assume the transmission channel is a Binary Symmetric Channel(BSC) (See Appendix B). The bit error probability p_b is the probability that the transmitted information bit is not the received information bit.
- We assume the bit error in different bits (on the binarised feature vector string) to be independent of each other.

Remark As was mentioned in the introduction, we use error correcting codes to extract a noiseless bits string from binarised feature feature in a noise tolerant manner. The way we use error correcting codes in this setting is not the way it is normally. We first explain how error correcting codes are used normally and then explain how we use them to correct noise NUPO systems. The standard usage of error correcting codes is in sending a message across a noisy communication channel.

Communication through a noisy channel can be described as follows: a string $\mathbf{x}$ of k message symbols needs to be sent from Alice to Bob via a noisy communication channel. The message $\mathbf{x}$ is encoded (using an $[n, k, d]$ code) into a code word c, which is a string of n channel symbols, where $n \geq k$. Encoding simply entails adding redundancy to the k message symbols. Because of the noise in the channel, the received word, c' might be different from the sent code word, c. The error in c' is corrected yielding c, and consequently $\mathbf{x}$.

In NUPO systems, we use the error correcting capability of error correcting codes to accommodate the noisiness of NUPO measurements. Let C be error correcting code with parameters $[n, k, d]$. Instead of starting with k message bits, we start with a received word (the binarised feature vector) g of n channel symbols. We select a code word $c \in C$ and use it to correct the noise in g. The helper data is $p = c + g$, the shift needed to get to c from g. The codeword is then decoded to its corresponding k message bits. The k message bits is output, it is the extracted string. At authentication, the NUPO is measured again, the resulting feature vector is binarised to obtain g'. The helper data p is XORed to g' to obtain c'. The error in c' corrected to yield c, if the noise is within tolerable limits. c is subsequently decoded to obtain k message bits, the reconstructed string.

Notation: Let $C_{n,p}(t) = \binom{n}{t} p^t (1-p)^{n-t}$.

Assuming we have an n bit received word whose bits have bit error probability p_b. The probability that this string has t specific (fixed) errors is $p_b^t (1-p_b)^{n-t}$.

The probability that it has any t errors is $C_{n,p_b}(t)$.

Let P_{ed} stand for the probability of erroneously decoding the received n bit word. The probability that received word has more than t errors is

$$P_{\text{ed}} = \sum_{i=t+1}^{n} C_{n,p_b}(i).$$

With an overwhelmingly high probability, this is the probability of incorrectly decoding a received word i.e. the probability that the error correction procedure fails. In most cases an error correcting code will decode a received word wrongly when $t > \lfloor \frac{d-1}{2} \rfloor$, d is the minimum distance of the code. However, $t \leq \lfloor \frac{d-1}{2} \rfloor$ is not a necessary condition for decoding correctly as correct decoding is also possible beyond $t > \lfloor \frac{d-1}{2} \rfloor$ in certain instances.

Settings and problem definition

Given is a binary string $g \in \{0,1\}^n$, fresh from the binarisation of the feature vector of NUPO. The objective is to extract a "consistent" noise free bit string [2] from these n bits, such that on receiving another bit string g' (from the binarisation of a feature vector from the same source,) of length n, we can reconstruct the extracted string with very high probability.

In practice, the n bits of g have differing error probabilities. However we are going to look at a somewhat simplified example. We surmise that our results in this specific case may offer insights to solving the general problem.

We sort the bits in ascending order according to their bit error probabilities (it is safe to do this since we assume that the bits of g are independent.) For a suitable $n_1 \in \mathbb{Z}$, we split g into two parts, the first n_1 bits which we call g_1 and the last $n_2 = n - n_1$ bits, g_2. n_1 is a function of the bit error of the bits that make up g.

[2] We aim at extracting a string whose bits have very low error probabilities after error correction.

We take the average r_1, of the bit error probabilities of the bits that make up g_1 and assign this as the bit error probability of each of the n_1 bits of g_1. We perform a similar operation for last n_2 bits. This results in an n bit strings whose first n_1 and last n_2 bits have bit error probability r_1 and r_2 respectively.

Let C be an error correcting code with parameters $[n, k, d]$. We select a (random) code word c and use it to correct the errors in g (see Section 2.3 of Chapter 2,). To extract a consistent, error free bit string from g, we consider g to be the received word, which is corrected (using the error correcting code) to a code word c', from which a k-bit error free message string is reconstructed. The k-bit string is the noise free extracted bit string.

We investigate the optimal means of correcting the error on g. Optimality is measured in terms of:

- Probability of erroneously (wrongly) decoding the received word, g. This is the probability that an error pattern happens that cannot be corrected by the chosen error correction technique.
- The maximum number of error free bits that can be extracted (i.e. maximum message length k.)

We compare the following error correcting techniques:

Technique A: Using a single $[n, k, d]$ code.

Technique B: Using 2 codes: One code of length n_1 correcting g_1 and the other of length $n_2 = n - n_1$ correcting g_2.

Technique C: Using code concatenation: Inner code of length N_1 and outer code of length N_2 such that $N_1 N_2 = n$.

Technique D: First correct one part of g, append the result to the other part and then correct with a code of appropriate length.

Technique E: First reduce the bit error probabilities of the bits using a suitable error correcting code, before attempting to extract noise free bits.

As a first step, we compute the formulae for the probability of erroneous decoding for the different techniques listed above.

Technique A: Using a single code.

Let t errors occur in the n-bit string, g. Let t_1 and t_2 errors occur in g_1 and g_2 respectively. t_1 and t_2 satisfy $t_1 + t_2 = t$.

The probability that bit string g_1 has t_1 errors and bit string g_2 has t_2 errors in any t_1 and t_2 positions in g_1 and g_2 respectively is $C_{n_1,r_1}(t_1)C_{n_2,r_2}(t_2)$.

The probability that g has any $t = t_1 + t_2$ errors is

$$\sum_{t_1,t_2:t_1+t_2=t} C_{n_1,r_1}(t_1)C_{n_2,r_2}(t_2) = \sum_{t_1=\max\{0,t-n_2\}}^{\min\{t,n_1\}} C_{n_1,r_1}(t_1)C_{n_2,r_2}(t-t_1).$$

Probability of erroneous decoding, using a single code with parameters $[n, k, d]$ (i.e. the probability that g has more than t-errors) is

$$P_{\text{ed}} = \sum_{j=t+1}^{n} \sum_{t_1=\max\{0,j-n_2\}}^{\min\{t,n_1\}} C_{n_1,r_1}(t_1)C_{n_2,r_2}(j-t_1), \quad \text{where } t = \lfloor \frac{d-1}{2} \rfloor. \tag{7.1}$$

The length of the error free message extracted from g is k.

Technique B: Using 2 codes, one code of length n_1 and another of length n_2.

This divide and correct technique entails correcting the noise using two error correcting codes, one code C_1 of length n_1 and a second code C_2 of length n_2.

Codes C_1 and C_2 have parameters $[n_1, k_1, d_1]$ and $[n_2, k_2, d_2]$ respectively. Let $t_1 = \lfloor \frac{d_1-1}{2} \rfloor$ and $t_2 = \lfloor \frac{d_2-1}{2} \rfloor$. Probability of decoding g_1 correctly is $\sum_{i=0}^{t_1} C_{n_1,r_1}(i)$. Similarly, the probability of decoding g_2 correctly is $\sum_{j=0}^{t_2} C_{n_2,r_2}(j)$.

Probability of successfully decoding g using Technique B is $\sum_{i=0}^{t_1} C_{n_1,r_1}(i) \sum_{j=0}^{t_2} C_{n_2,r_2}(j)$. We are allowed to multiply the probabilities because of our assumption that the bits are independent.

$$P_{\text{ed}} = 1 - \sum_{i=0}^{t_1} C_{n_1,r_1}(i) \sum_{j=0}^{t_2} C_{n_2,r_2}(j). \tag{7.2}$$

In this case, the number of error free bits extracted is $k_1 + k_2$.

Technique C: Code concatenation.

Let C_1 be a q-ary $[N_1, k_1, d_1]$ code and C_2 be a $[N_2, k_2, d_2]$ code with alphabet of size q^{k_1}, then there exists a q-ary, $[N_1N_2, k_1k_2, d_1d_2]$ code C. C_1 is called the inner code and C_2 the outer code. We consider only the binary case, i.e. $q = 2$. Let $t_1 = \lfloor \frac{d_1-1}{2} \rfloor$ and $t_2 = \lfloor \frac{d_2-1}{2} \rfloor$.

Given is a string of length $n = N_1N_2$. Split n into N_2 bit strings of size N_1 each. Let $z \in \mathbb{Z}$, be such that $zN_1 = n_1$ and $(N_2 - z)N_1 = n_2$.

Let $s_1 = \sum_{j=t_1+1}^{N_1} C_{N_1,r_1}(j)$. g_1 is divided into z stings of length N_1. The probability of erroneously decoding any of these length-N_1 strings in g_1 is s_1.

Let $s_2 = \sum_{j=t_1+1}^{N_1} C_{N_1,r_2}(j)$. Similarly, g_2 is divided into $N_2 - z$ stings of length N_1. The probability of erroneously decoding any of these length-N_1 strings in g_2 is s_2.

At the end of this first stage of decoding we have string of N_2 symbols. The first z symbols have symbol error probability s_1 and the last $N_2 - z$ have symbol error probability s_2.

Let τ_1 errors occur in the first z symbols and let τ_2 errors occur in the last $N_2 - z$ symbols. The string of N_2 symbols is decoded wrongly when $\tau_1 + \tau_2 > t_2$.

Hence, the probability of erroneously decoding g using the concatenation of codes C_1 and C_2 given above is

$$P_{\text{ed}} = \sum_{l=t_2+1}^{N_2} \sum_{\tau_1=\max\{0,l-(N_2-z)\}}^{z} C_{z,s_1}(\tau_1) C_{N_2-z,s_2}(l - \tau_1). \tag{7.3}$$

The length of error free message bits extracted from g is k_1k_2.

Technique D: Correct a section of the bit string, append the results to the uncorrected section and then correct the whole string.

D_1 We first correct g_2 , append the result to g_1 and then correct using a code of appropriate length.

The probability of successfully decoding g_2 using a code C_2 with parameters $[n_2, k_2, d_2]$ is $\sum_{i=0}^{t_2} C_{n_2,r_2}(i)$. The corrected n_2 bits are mapped to the corresponding k_2 message symbols (which are bits). This is then appended to g_1. The resulting $n_1 + k_2$ bit string is then decoded

using a single code C with parameters $[n_1 + k_2, k, d]$. This holds because in correcting the error in g_2, we map g_2 to a code word in C_2, and there are only 2_2^k of such code words. Let $t = \lfloor \frac{d-1}{2} \rfloor$.

If g_2 was decoded correctly, things can only go wrong if more than t errors occur in g_1. If less than t errors in g_1, then we are guaranteed a successful decoding of g.

Probability that less than t errors occur in g_1 given that the last n_2 bits were successfully decoded in the first stage (keeping in mind that g_1 can have at most n_1 errors) is $\sum_{j=0}^{\min\{n_1,t\}} C_{n_1,r_1}(j)$.

Probability of successfully decoding g using this technique is

$$\sum_{j=0}^{\min\{n_1,t\}} C_{n_1,r_1}(j) \sum_{i=0}^{t_2} C_{n_2,r_2}(i).$$

Again, we are allowed to multiply the probabilities because of our assumptions that the bits are independent. Hence

$$P_{\text{ed}} = 1 - \sum_{j=0}^{\min\{n_1,t\}} C_{n_1,r_1}(j) \sum_{i=0}^{t_2} C_{n_2,r_2}(i). \tag{7.4}$$

The length of the error free message extracted from g is k.

Remark We neglect the event that two wrong decodings accidentally yields the correct result.

D_2 We repeat a similar procedure, switching the role of g_1 and g_2. Correspondingly,

$$P_{\text{ed}} = 1 - \sum_{j=0}^{\min\{n_2,t\}} C_{n_2,r_2}(j) \sum_{i=0}^{t_1} C_{n_1,r_1}(i).$$

The first and second error correction are performed with codes with parameters $[n_1, k_1, d_1]$ and $[n_2 + k_1, k, d]$.

Similar to D_1, the length of the error free message extracted is k.

Technique E: First reduce the bit error probabilities of the bits using a suitable error correcting code, before attempting to extract noise free bits.

If the bit error probabilities of the bits that make-up g are relatively large, one may surmise that applying error correction techniques without any attempt to reduce the error probabilities, may lead to a relatively large probability of erroneous decoding and low information rate. In this technique, we first reduce the bit error probability of the bits in the g before proceeding to extract the unique identifier. Reducing the bit error probability is achieved using error correcting codes. We may reduce the bit error probability of one or both group of bits depending on the bit error probabilities of the respective groups. The decision of whether or not to reduce the bit error probability depends on the value of the error probability, the desired length of the extracted string and amount of accuracy desired. In this exercise, we reduce the bit error probability of both groups of bits.

Using a suitable error correcting code, we reduce the bit error probability of the bits in g_2 from r_2 to s_2. In doing so, the length of the bit string is reduced from n_2 to m_2. The bit error probability of the first n_1 bits is also reduced from r_1 to s_1, reducing the number of bits to m_1 in the process. The error correcting codes used in reducing r_1 to s_1 and r_2 to s_2 need not be the same.

Split g_2 into subsets of η_2 bits. Using an error correcting code of dimensions $[\eta_2, \kappa_2, \delta_2]$, η_2 bits with bit error probability r_2 are reduced to κ_2 bits with bit error probability s_2.

$$s_2 = \sum_{i=\lfloor \frac{\delta_2 - 1}{2} \rfloor + 1}^{\eta_2} C_{\eta_2, r_2}(i).$$

For simplicity we set $\kappa_2 = 1$, hence $[\eta_2, \kappa_2, \delta_2] = [\eta_2, 1, \delta_2]$.

Similarly, using an error correcting code with parameters, $[\eta_1, \kappa_1, \delta_1] = [\eta_1, 1, \delta_1]$, we reduce g_1s' n_1 bits with bit error probability r_1 to $\frac{n_1}{\eta_1}$ bits with bit error probability s_1.

$$s_1 = \sum_{i=\lfloor \frac{\delta_1 - 1}{2} \rfloor + 1}^{\eta_1} C_{\eta_1, r_1}(i)$$

At the end of the bit error probability reduction stage, we have an $m = m_1 + m_2 = \frac{n_1}{\eta_1} + \frac{n_2}{\eta_2}$ bits. We may now apply any of Techniques A, B, and D to extract a noise free unique identifier string. Before considering Technique E, we compared Techniques A, B and D. The technique of choice is Technique A, because it produced better results than Techniques B and D (See Table 7.1).

Applying Technique A, the probability of erroneous decoding, P_{ed} of the new m-bit string using a code with parameters $[m, \kappa, \delta]$ is

$$P_{\text{ed}} = \sum_{j=\tau+1}^{m} \sum_{\alpha=\max\{0, j-\eta_2\}}^{m_1} C_{m_1, s_1}(\alpha) C_{m_2, s_2}(j - \alpha),$$

where $\tau = \lfloor \frac{\delta - 1}{2} \rfloor$ and α is the number of errors in the m_1 bits.

The length of the extracted string is κ.

Remark It is not mandatory to perform bit error reduction for both classes of bits in g_1 and g_2. The choice of whether or not to reduce the bit error probabilities of one or both sets of bits depends on the value bit error probabilities of the affected bits.

7.2 Comparing error correcting techniques

We compare the techniques listed above. Our comparison will be based on the probability of erroneously decoding the received word g and the length k of message bits extractable. If g is erroneously decoded, authentication with the corresponding k bit message string (extracted string) will fail. If we assume that the received word g is the measurement/response from a valid system object, the probability of erroneously decoding g is the probability that a valid user is denied access to the system, also called the False Rejection Rate (FRR). A counterpart notion is the False Acceptance Rate (FAR) which is the probability that an imposter gains access to the system upon presenting his credentials. The FAR and FRR are negatively correlated quantities. High security in biometric systems demands a very low FAR($\approx 0.1\%$), this enforces a relatively high FRR($\approx 2.5\%$). For example, in the European 3D-face project[31], FAR $< 0.25\%$ and FRR $< 2.5\%$. We use this benchmark as the acceptable FRR in the following example:

Example 7.2.1 *Given is a received word, g of length $n = 256$. $r_1 = 0.15$, $r_2 = 0.2$, $n_1 = 64$, and $n_2 = 192$.*

We determine which of techniques A-E provides the best means of extracting a noise robust bit string from g.

We first compute the channel capacity[3], to help us compare our results to the theoretical optimum. To do this we need the Entropy function.

Definition 7.2.2 *The entropy function $h(p)$ is defined for $0 \leq p \leq 1$ by*

$$h(p) = \begin{cases} -p\log_2 p - (1-p)\log_2(1-p) & \text{if } 0 < p < 1; \\ 0 & \text{if } p = 0 \text{ or } 1. \end{cases}$$

Lemma 7.2.3 *The capacity of a binary symmetric channel $\mathcal{C}$ with error probability p_b is*

$$\mathcal{C} = 1 - h(p_b).$$

The capacity measures the overall error characteristics of a channel. The smaller the capacity the more frequently errors occur and an overly efficient error-correcting code will not build in enough error correction capability to counteract channel errors [27].

Channel capacity of Example 7.2.1.

- $\mathcal{C}_1 = 0.39016$ and $\mathcal{C}_2 = 0.2781$.
- $n_1\mathcal{C}_1 + n_2\mathcal{C}_2 = 78.36$
- $\mathcal{C} = \frac{n_1C_1+n_2C_2}{n} = 0.3061$.

Table 7.1: Comparing error correcting techniques for Example 7.2.1

Tech.	Error correcting code parameters	P_{ed}	Length of extracted string
A	$[n,k,d] = [256, 10, 123]$	0.0174	10
	$[n,k,d] = [256, 13, 120]$	0.0352	13
B	$[n_1,k_1,d_1] = [64, 7, 32]$, $[n_2,k_2,d_2] = [192, 2, 128]$	0.0248	9
	$[n_1,k_1,d_1] = [64, 7, 32]$, $[n_2,k_2,d_2] = [192, 3, 109]$	0.0273	10
C	$[N_1,k_1,d_1] = [3, 1, 3]$, $[N_2,k_2,d_2] = [85, 21, 28]$	0.0252	21
	$[N_1,k_1,d_1] = [7, 1, 7]$, $[N_2,k_2,d_2] = [36, 21, 7]$	0.0176	21
D_1	$[n_2,k_2,d_2] = [192, 02, 128]$, $[n_1+k_2,k,d] = [66, 7, 32]$	0.0248	7
	$[n_2,k_2,d_2] = [192, 3, 109]$, $[n_1+k_2,k,d] = [67, 8, 31]$	0.0273	8
	$[n_2,k_2,d_2] = [192, 4, 102]$, $[n_1+k_2,k,d] = [68, 5, 34]$	0.0282	5
D_2	$[n_1,k_1,d_1] = [64, 2, 42]$, $[n_2+k_1,k,d] = [194, 5, 99]$	0.0255	5
	$[n_1,k_1,d_1] = [64, 7, 32]$, $[n_2+k_1,k,d] = [199, 6, 100]$	0.0494	6
E_1	$[\eta_1,\kappa_1,\delta_1] = [3, 1, 3]$, $[\frac{n_1-1}{\eta_1}+n_2,k,d] = [213, 9, 103]$	0.0207	9
E_2	$[\eta_2,\kappa_2,\delta_2] = [7, 1, 7]$, $[n_1+\frac{n_2-3}{\eta_2},k,d] = [91, 15, 36]$	0.0141	15
E_3	$[\eta_1,\kappa_1,\delta_1] = [7, 1, 7]$		
	$[\eta_2,\kappa_2,\delta_2] = [7, 1, 7]$, $[\frac{n_1-1}{\eta_1}+\frac{n_2}{\eta_2},k,d] = [36, 21, 7]$	0.0176	21

We worked out three scenarios of Technique E using Example 7.2.1. In E_1 the bit error probability of only the first n_1 bits of g is reduced. In E_2 the bit error probability is reduced only for g_2, while in E_3 we reduce the bit error probabilities of both g_1 and g_2.

[3]The channel capacity is the tightest upper bound on the amount of information that can be reliably transmitted over a communications channel.

Interpreting the results

Comparing techniques $A - D$, Technique D proves to be the worst technique. Technique A gives a better result than Techniques B and D, because with standard error correcting codes, performance increases as the length of the code increases.

Technique C proves to be the best of the first four techniques, this is explained by Technique E.

First reducing the error probability of the bits before extracting the unique identifier bit string yields much better results that Techniques A, B and D. Reducing the bit error probability shortens the length of the bit string from which we extract. But the reduced error of the bits imply that more error free bits can be extracted with low probability of decoding wrongly. This is exactly what happens when we use the code concatenation technique in the Technique C, and that is why Technique C gave better results than A, B and D. The inner code C_1 (in Technique C) was used to reduce the error probability of the bits while the outer code C_2 extracted the output bit string. In reducing the bit error probabilities of the bits, the trivial repetition code was used.

Standard error correcting codes operate best when the bit error probability is low. The gap between high bit error probability and the error rate well tolerated by standard ECCs is bridged using error correcting codes. Among all existing ECC, the repetition code corrects the most error but has the lowest information rate. We compare the information rate and the error correcting capability of standard error correcting codes to see which is best suited for reducing bit error probabilities.

To compare error correcting codes to determine which is most suited for reducing the error probabilities of the bits, we make use of the Shannon information theory.

Theorem 7.2.4 *[Shannon's noisy channel coding theorem] All discrete memoryless channels have a non-negative channel capacity $\mathcal{C}$. For any $\lambda > 0$ and $R < \mathcal{C}$, for large enough n, there exists a block code of length n and rate $\geq R$ and a pair of encoding and decoding algorithm, such that the maximal probability of incorrectly decoding is $\leq \lambda$.*

For rates R greater than $\mathcal{C}$, no encoding and decoding can be made with probability of incorrect decoding tending to 0.

The Shannon theorem implies that theoretically, it is possible to transmit information nearly without error at any rate below a limiting rate, $\mathcal{C}$. The converse means that if $R > \mathcal{C}$, an arbitrarily small probability of incorrect decoding is not achievable. In this instance, all codes will have a probability of incorrect decoding greater than a certain positive minimal level, and this level increases as the rate increases[34].

From Theorem 7.2.4 we can infer any code that will be useful for reducing the bit error probability must satisfy $R < \mathcal{C}$. We provide more insight to this in the following example.

Example 7.2.5 *Let $g \in \{0,1\}^n$ be a bit string whose bits each have a bit error probability $r = 0.20$. We compare standard error correcting codes to determine which is most suited for reducing the bit error probability.*

A necessary condition to be satisfied by any suitable $[n, k, d]$ code C, is that $R < \mathcal{C}$.

$R = \frac{k}{n}$. And $\mathcal{C} = 1 - h(r) = 0.2781$. So any suitable code must satisfy $\frac{k}{n} < 0.2781$

Table 7.2 gives our comparison of standard error correcting codes.
Table 7.2, shows that the error correcting code used for reducing the bit error probability depends very strongly on the value of the bit error probability and on the length of g, n. If the bit error probability of the bits is relatively high (as in Example 7.2.5), using repetition code to reduce bit error probability is most effective. However when the error probability is not so high then the the BCH and the Reed-Muller codes may prove to be more advantageous because their information rate is closer to the channel capacity than the repetition code. In Example 7.2.5, the Hamming code is not suitable because

Table 7.2: Finding a suitable code for reducing bit error probability

Code	Parameters	Condition for $\frac{k}{n} < 0.2781$	$\frac{k}{n}$	Example
Repetition	$[n, 1, n]$	When $n \geq 5$	0.2	[5, 1, 5]
Hamming	$[2^m - 1, 2^m - m - 1, 3]$	$\nexists m \in \mathbb{Z} : \frac{2^m - m - 1}{2^m - 1} < 0.2781$	N/A	N/A
Reed-Muller[4], R(1,m)	$[2^m, m + 1, 2^{m-1}]$	When $m \geq 5$	0.1875	$[2^5, 6, 2^4]$
BCH	$[2^m - 1, \geq 2^m - 1 - mt, \geq 2t + 1]$	When $m \geq 4$	0.2667	[15, 4, 8]
Golay	[23, 12, 7]	$\frac{12}{23} > 0.2781$	0.5217	N/A
Reed-Solomon	$[2^m - 1, 2^m - 1 - 2t, 2t + 1]$	Non binary	N/A	N/A

hamming codes with rate less than 0.2781 do not exist. If $|g| \geq 23$, we have the option of choosing between the Repetition, the BCH and the Reed-Muller codes. Our exact choice in this situation will be guided by how close the code's information rate is to the channel capacity and by the quantity of bit error reduction.

Discussion

We conclude that when the bit error probabilities are low, extracting a bit string using an error correcting code of the same length as original noisy string provides better results, because the performance of standard error correction codes increases with increasing code length.

When the bit error probabilities of the individual bits are high, it is most effective to first reduce the bit error probability, before attempting extraction. This ensures that the length of the extracted string is a long as possible while minimising the probability of erroneous decoding.

Our results are consistent with the results of an earlier work by Bosch [30]. His work described the complexities and design choices made to implement fuzzy extractors on hardware, more specifically on Field Programmable Gate Arrays (FPGAs). He modelled the noise present in the FPGA responses as a BSC and aimed at finding a good code that achieved a pre-specified $P_{ed} \leq 10^{-6}$ for different bit error probabilities. Fixing the number of error free bits that should be obtained, he computed number of source bits using any specified $[n, k, d]$ code. We explain this by the means of the following example. Given that the bit error probability is p_b and the number of error free bits desirable is 171. For each $[n, k, d]$ code C, he computed the probability of erroneous decoding compared it with the pre-specified $P_{ed} \leq 10^{-6}$ and computed number of source bits $\lceil \frac{171}{k} \times n \rceil$ required. His results showed that using the concatenation code technique produces the best performance, in terms of minimising the number of source bits for any pre-specified P_{ed}.

7.3 Chapter summary

In this Chapter we investigated the problem of optimising the length of the extracted bit string while minimising the inability to correct error by choosing a appropriate error correcting technique. Our problem is particularly challenging because after the binarisation of the NUPO's feature vectors, we obtain a bit string whose bits have varying bit error probabilities. These error probabilities range from low to high. The challenge was to extract from this noisy string, a consistent bit string that can be easily reproduced on input of future measurements of the same NUPO. We considered a simplified version of the problem, we considered the situation where the bits on the noisy binarised feature vector string have two different bit error probabilities. We simplified the problem this way because it is subject to complete analysis and envisage that it can offer insight to the solution of the general problem.

Our findings are summarised in the following;

When the bit error probabilities of the bits from the binarisation of feature vectors are low, applying error correcting codes of the same length as the original noisy string provides better results, because the functionality of standard error correcting codes increase as code length increases.

When the bit error probabilities of the individual bits are high, it is most effective to first reduce the bit error probability before attempting to extract the unique identifier bit string. This ensures that the length of the extracted string is maximised and that the probability of erroneous decoding is low. The code concatenation technique (Technique *C* in Section 7.1) may be used to achieve this result.

Chapter 8

Conclusions

8.1 Results

Our motivation for this work was to improve the functionality and security of helper data systems. As was mentioned in the introductory chapter, these two notions are interwoven because helper data systems are security systems and their function is to provide adequate security to systems and information. Therefore we may surmise that any concern which affects functionality affects security. Enforcing security in helper data systems is a non-trivial task, primarily because of the noisiness of the NUPO.

We started by giving a comprehensive discussion of helper data systems, what they are, their primary objectives and the challenges to meeting these objectives.

Using a generic construction of a fuzzy extractor in the helper data system, we highlighted the following salient construction concerns; privacy, entropy after discretisation, length of extracted string taking into account the system designer's lack of knowledge about the true NUPO distribution and the shape of noise during the extraction procedure. Using the generic construction, we investigated the shape of the NUPO noise on the unit interval which can be protracted to the effect of NUPO noise on the extracted string space. This result is useful because it gives useful insight to determining which error patterns are more/less likely and can aid the designer in the choice of a suitable error correcting technique. We also computed the length of the extracted string for our construction taking into account the system designer's lack of knowledge about the true distribution of the NUPO.

A large portion of this work was focused on the threats and vulnerabilities in the helper data system. We identified the some of the major threats and vulnerabilities that exist in these systems and have provided useful insights to the solutions of some of these problems. We considered two major causes of failure in helper data systems, failures resulting from attacks by an adversary and failures resulting from intrinsic vulnerabilities which lead to system malfunctioning and provide an opening for attack by the adversary. We further classified these threats and vulnerabilities according to the system component/module which they affect.

Starting from the design stage of the NUPO system, we considered the intrinsic vulnerability of improperly estimating the distribution of the expected input, the NUPO distribution. As this estimate is obtained from a sample population, we derived relationships between the sample size and the distance between the real and estimated distributions. In addition we computed probabilistic bounds between the sample size and certain security parameters. Our results can be applied in determining the security level achieved by any given sample size, or conversely, given a certain desirable level of security, we can compute the sample size that achieves the security target.

One of the most prominent and potentially damaging attack on the helper data system is on the helper data itself, which is usually considered public. We authenticate the helper data to detect unauthorised modification. We gave an in-depth and exhaustive discussion on helper data authentication mecha-

nisms. We highlighted solutions which can be proved secure using the PKI infrastructure, and in the standard, random oracle and common reference string models.

To accommodate natural drifts in the NUPO measurement over time, it may be necessary to have the helper data modified (by an authorised party). We provided a secure means of allowing for the authorised modification of the helper data by an authorised censor, using Chameleon hashes in Sanitizable signatures. The scheme presented in here is particularly useful because it allows for delegation of duty while ensuring continual use of the NUPO system. This is relevant in the situation that the enrolment authority is not easily accessible.

It is common knowledge that the bit string obtained from the binarisation of biometric feature vectors have bits with high and varying bit error probabilities. The challenge in this situation is to find an efficient means of extracting a consistent string from the binarised feature vector in a noise tolerant manner (i.e. the string should be easily reproducible on input of a noisy version of the binarised feature vector). This is achieved using error correcting codes. We investigated the problem of maximizing the length of the string extracted from the binarised feature vector while minimising probability that an error pattern occurs than cannot be corrected by a specific error correcting technique. We considered the simplified situation where the bits of the binarised feature vector have two different bit error probabilities. We compared various error correcting techniques to determine which effectively corrects noise while optimising that the length of the extracted bit string. Our results showed that when the bit error probability is high, reducing the error probability of the bits before attempting to extract a noise free bit string proves to be most effective. This buttresses the results of previous works [30], and earlier references cited therein.

In discussing the security/functionality issues above, we have answered some of the salient questions of every security system such as what exactly we need to protect, against what/whom? what we hope to achieve, how we achieve it, how security affects the functioning of the system and the cost of security. While the answers to some of these questions are trivial for helper data systems, others are not. Our work in this thesis touches all aspects. Functionality and security in helper data systems are intertwined, and so improvements such choosing the right sample size to estimate the distribution of the NUPO (Chapter 5), the shape of the NUPO noise during the extraction procedure (Section 4.4 of Chapter 4) and choosing effective error correcting techniques (Chapter 7) improve both the performance and security of the helper data system. Providing a secure means of securely modifying helper data in Section 6.2 provides added functionality to the helper data system and allows for its continual usage.

In terms of cost of security the results of Chapter 5 can enable the designer compute how much samples he should use to estimate the distribution of the NUPO to achieve a pre-specified amount of security (security here is measured parameters ϵ and κ, see Section 5.2). By doing this he can ensure that an adversary does not have substantially better information about the extracted bit string.

In addition most of our results have a wider application. The results in Chapter 7, can be applied to the general scenario of optimising error correcting techniques, with the aim of minimising the probability of failing to correct noise, when the bit error probabilities of the bits involved are relatively high. The results of Section 4.4 can be employed in understanding the behaviour of noise as it propagated across different metric spaces by some pre-specified function. The estimates and bounds obtained for the relationship between the sample size and distance between the real and estimated distributions find application whenever the distribution of an estimate is used instead of the true distribution.

8.2 Recommendations and open problems

Our work focused on identifying, preventing and proffering solutions to various threats and vulnerabilities in the helper data systems. We quantified the cost (in terms of resources and techniques) of preventing these threats and vulnerabilities. However we did not quantify the consequences of the mentioned threats and vulnerabilities nor test their impact on live constructions of helper data systems. In particular it would be interesting to find out the exact impact of concerns such as choice of

error correcting techniques and sample size on a live helper data system. We find this interesting because intrinsic vulnerabilities such as these do not usually lead to an immediate collapse of the system but keep it functioning at a substantially sub-optimal level.

In addition, we have not considered side channel attacks such as differential fault analysis[1] and radiation monitoring attacks[2]. We believe that insight into this category of attacks is beneficial for the overall robust security of helper data systems.

[1] Inducing faults (unexpected environmental conditions) into cryptographic implementations, to reveal their internal states.
[2] Attacks based on leaked electromagnetic radiation which can directly provide information about secret data.

Bibliography

[1] Y. Dodis, L. Reyzin, and A. Smith, *Fuzzy extractors: How to generate strong keys from biometrics and other noisy data*, Advances in cryptology—EUROCRYPT 2004, Lecture Notes in Computer Science, vol. 3027, Springer, Berlin, 2004, pp. 523–540.

[2] X. Boyen, *Reusable Cryptographic Fuzzy Extractors*, ACM Conference on Computer and Communications Security—CCS 2004, ACM Press, New York, 2004, pp. 82-91.

[3] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith, *Secure Remote Authentication Using Biometric Data*, revised version available at: http://www.cs.stanford.edu/ xb/eurocrypt05b/, 2005.

[4] R. Cramer, Y. Dodis, S. Fehr, C. Padro, and D. Wichs, *Detection of Algebraic Manipulation with Applications to Robust Secret Sharing and Fuzzy Extractors*, Advances in Cryptology—EUROCRYPT 2008, Lecture Notes in Computer Science, Springer-Verlag, 2008.

[5] Y. Dodis, J. Katz, L. Reyzin, and A. Smith, *Robust Fuzzy Extractors and Authenticated Key Agreement from Close Secrets*, Advances in Cryptology—CRYPTO 2006, Lecture Notes in Computer Science, vol. 4117, Springer, Berlin, 2006, pp. 232–250.

[6] V.V.T. Tong, H. Sibert, J. Lecoeur, and M. Girault, *Biometric Fuzzy Extractors Made Practical: A Proposal Based on Finger Codes*, Advances in Biometrics—International Conference on Biometrics 2007, Lecture Notes in Computer Science, vol. 4642, Springer, Berlin, 2007, pp. 604–613.

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

[8] Q. Li, Y. Scutcu, and N. Memon, *Secure Sketch for Biometric Templates*, Advances in cryptology—ASIACRYPT 2006, Lecture Notes in Computer Science, vol. 4284, Springer, Berlin, 2006.

[9] E.A. Verbitskiy, P. Tuyls, D. Denteneer, and J.P. Linnartz, *Reliable Biometric Authentication with Privacy Protection*, XXIV Benelux Symposium on Information Theory, 2003, pp. 125-132.

[10] R. Renner and S. Wolf, *Simple and Tight Bounds for Information Reconciliation and Privacy Amplification*, Advances in Cryptology—ASIACRYPT 2005 (Bimal Roy, ed.), Lecture Notes in Computer Science, vol. 3788, Springer, India, 2005, pp. 199–216.

[11] G. Anteniese, D. Chou, D. B Medeiros, and G. Tsudik, *Sanitizable Signatures*, European Symposium on Research in Computer Security—ESORICS 2005, Lecture Notes in Computer Science, vol. 3679, Springer-Verlag, 2005, pp. 159–177.

[12] N. Ratha, J. Connell, R. M Bolle, and S. Chikkerur, *Cancelable Biometrics: A Case Study in Fingerprints*, 18th International Conference on Pattern Recognition—ICPR 2006, Vol. 4, IEEE Computer Society, 2006, pp. 370–373.

[13] P. Tuyls, B. Škorić, and T. Kevenaar, *Security with noisy data: Private Biometrics, secure key storage and anti-counterfeiting*, Springer, London, 2007.

[14] I. Buhan and P. Hartel, *The State of the Art in Abuse of Biometrics*, Technical Report: Center for Telematics and Information Technology,University of Twente, Vol. 2008, Hindawi Publishing Corporation, 2005.

[15] C. Roberts, *Biometric Attack Vectors and Defences*, Computers and Security, Vol. 1, 2007, pp. 14-25.

[16] A.K. Jain, K. Nandakumar, and A. Nagar, *Biometric Template Security*, EURASIP Journal on Advances in Signal Processing, Vol. 2008, Hindawi Publishing Corporation, 2007.

[17] Search Security, *False Acceptance and False Rejection Rates*, http://searchsecurity.techtarget.com.

[18] P.Tuyls and J. P. Linnartz, *New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates*, Audio- and Video-Based Biometric Person Authentication, Lecture Notes in Computer Science, vol. 2688, Springer, Berlin/Heidelberg, 2003.

[19] UR Biometirc, *Biometric Features*, http://www.fp.net.nz/Biometrics.htm.

[20] U.M. Maurer, *Secret key agreement by public discussion from common information*, IEEE Transactions on Information Theory, Vol. 39(3), 1993, pp. 733-742.

[21] C.S. Petrie and J.A. Connelly, *A noise-based IC random number generator for applications in cryptography*, IEEE Transactions on [Circuits and Systems I:]Fundamental Theory and Applications, Vol. 47(5), 2000, pp. 615-621.

[22] A. Juels and M. Sudan, *A Fuzzy Vault Scheme*, IEEE International Symposium on Information Theory (2002 proceedings), 2002, pp. 408.

[23] C.S. Petrie and J.A. Connelly, *A noise-based IC random number generator for applications in cryptography*, IEEE Transactions on [Circuits and Systems I:]Fundamental Theory and Applications, Vol. 47(5), 2000, pp. 615-621.

[24] E.A. Verbitskiy, P. Tuyls, B. Škorić, L.A.M. Schoenmakers, and C.O. Obi, *Continuous Space Fuzzy Extractors*, Manuscript, 2008.

[25] V. Shoup, *A Computational Introduction to Number Theory and Algebra (Version 2)*, http://shoup.net/, 2008.

[26] Y. Dodis and J. Spencer, *On the (non)universality of the one-time pad*, The 43rd Annual IEEE Symposium on the Foundations of Computer Science, 2002. Proceedings, 2002, pp. 376 - 385.

http://cnx.org/content/m0073/latest/

[27] D.Johnson, *Noisy Channel Coding Theorem*, http://cnx.org/content/m0073/latest/, 2007.

[28] A.K. Jain, A.Ross, and S. Prabhakar, *An Introduction to Biometric Recognition*, IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics, Vol. 14(1), 2004.

[29] F. J. MacWilliams, N. J. A. Sloane, and S. Prabhakar, *The Theory of Error-Correcting Codes*, Vol. I and II, North-Holland, Amsterdam, 1977.

[30] C.Bosch, *Efficient Fuzzy Extractors for Reconfigurable Hardware*, Masters Thesis, 2008.

[31] *3D Face Biometric Research*, http://www.3dface.org/, 2008.

[32] R. Pellikaan, X. Wu, and S. Bulygin, *Codes and Cryptography on Algebraic Curves*, Book in preparation, 2008.

[33] R. Rubinfeld, *Randomness and Computation: The Leftover Hash Lemma and Explicit Extractors*, Lecture notes, http://people.csail.mit.edu/ronitt/COURSE/S08/drafts/22.pdf, 2008.

[34] Wikipedia, http://www.wikipedia.org/, 2008.

[35] E.W. Weisstein, *Change of Variables Theorem*, From MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/ChangeofVariablesTheorem.html, 2008.

Appendix A

Appendices

A.1 Appendix A

Proof of Lemma 4.4.2

Let $Z = F(X)$.

$$\begin{aligned} z &= F(x) = \int_{-\infty}^{x} f_X(t)dt \\ f_X(x)dx &= f_Z(z)dz \\ f_Z(z) &= f_X(x)\frac{1}{\frac{dz}{dx}} \end{aligned} \tag{A.1}$$

Using Eqn.(A.1)

$$f_Z(z) = \frac{f_X(x)}{f_X(x)} = 1. \quad \square$$

Proof of Theorem 4.4.4

1. The proof that $f_Z = 1$ follows from Lemma 4.4.2.

   We prove that $f_{Z',Z} = f_{Z'|Z}$.

$$\begin{aligned} f_{Z'|Z}(z', z) &= \frac{f_{Z',Z}(z', z)}{f_Z(z)} \\ &= f_{Z',Z}(z', z) \text{ Follows from Lemma 4.4.2 .} \end{aligned}$$

2. Next we derive an expression for the density function $f_{Z'}$.

$$f_{X'}(x')dx' = f_{Z'}(z')dz' \tag{A.2}$$

$$z' = F(x') = \int_{-\infty}^{x'} f_X(t)dt \tag{A.3}$$

   From Eqn.(A.3)

$$\frac{dx'}{dz'} = \frac{1}{f_X(x')} \tag{A.4}$$

$$\begin{aligned} f_{Z'}(z') &= \frac{f_{X'}(x')}{f_X(x')} \\ &= \frac{f_{X'}(F^{\text{inv}}(z'))}{f_X(F^{\text{inv}}(z'))}. \end{aligned}$$

3. Finally, we prove that $f_{Z',Z} = \frac{f_Y(F^{\text{inv}}(z') - F^{\text{inv}}(z))}{f_X(F^{\text{inv}}(z'))}$.

   Before giving the proof, we state the following Change of variable theorem, which will prove helpful.

   Theorem A.1.1 [?Mathworld]*[Change of variable theorem]*

$$\int_R f(x,y)dx\,dy = \int_{R^*} f(x(u,v), y(u,v)) \left|\frac{\partial(x,y)}{\partial(u,v)}\right| du\,dv,$$

   where $\left|\frac{\partial(x,y)}{\partial(u,v)}\right|$ *stands for the Jacobian,* $\left|\frac{\partial(x,y)}{\partial(u,v)}\right| = \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{vmatrix}$ *and* $R = f(R^*)$ *is the image of the original region* R^*.

   Recall

$$\begin{aligned} x &= F^{\text{inv}}(z) \text{ and} \\ x' &= F^{\text{inv}}(z'). \end{aligned}$$

   In the following we will make use of the assumption that X and Y are independent.

$$\begin{aligned} f_{ZZ'}(z,z')dz\,dz' &= f_X(x)dx f_Y(x'-x)dy \\ f_{ZZ'}(z,z') &= f_X(x) f_Y(x'-x) \left|\frac{\partial(x,y)}{\partial(z,z')}\right| \quad \text{By the Change of variable theorem} \\ &= f_X(x) f_Y(x'-x) \begin{vmatrix} \frac{\partial z}{\partial x} & \frac{\partial z}{\partial y} \\ \frac{\partial z'}{\partial x} & \frac{\partial z'}{\partial y} \end{vmatrix}^{-1} \\ &= \frac{f_X(x) f_Y(x'-x)}{f_X(x) f_X(x')} \quad \text{Using } \frac{\partial z}{\partial x} = f_X(x),\ \frac{\partial z}{\partial y} = 0, \text{ and } \frac{\partial z'}{\partial y} = \frac{\partial z'}{\partial x'} \cdot \frac{\partial x'}{\partial y} = f_X(x'). \\ &= \frac{f_Y(x'-x)}{f_X(x')} \\ &= \frac{f_Y(F^{\text{inv}}(z') - F^{\text{inv}}(z))}{f_X(F^{\text{inv}}(z'))} \quad \text{Substituting } x = F^{\text{inv}}(z) \text{ and } x' = F^{\text{inv}}(z'). \quad \square \end{aligned}$$

Proof of Corollary 4.4.5

1. Let f_X and f'_X be the density function of X and X' respectively.

$$X \sim \mathcal{N}(\mu_x, \sigma_x^2) \text{ and } Y \sim \mathcal{N}(0, \sigma_y^2) \text{ so, } X' = X + Y \sim \mathcal{N}(\mu_x, \sigma_x^2 + \sigma_y^2).$$

$$z = F(x) = \int_{-\infty}^{x} f_X(t)dt = \frac{1}{2}\left(1 + \text{erf}\frac{x-\mu_x}{\sqrt{2\sigma_x^2}}\right) \tag{A.5}$$

$$x = \mu_x - \sigma_x\sqrt{2}\text{erfc}^{\text{inv}}(2z) \tag{A.6}$$

$$z' = F(x') = \int_{-\infty}^{x'} f_X(t)dt = \frac{1}{2}\left(1 + \text{erf}\frac{x'-\mu_x}{\sqrt{2\sigma_x^2}}\right) \tag{A.7}$$

$$x' = \mu_x - \sigma_x\sqrt{2}\text{erfc}^{\text{inv}}(2z') \tag{A.8}$$

$$f_{X'}(x')dx' = f_{Z'}(z')dz' \tag{A.9}$$

$$\begin{aligned} f_{Z'}(z') &= \frac{f_{X'}(x')}{f_X(x')} \\ &= \frac{\sigma_x\sqrt{2\pi}}{\sqrt{2\pi(\sigma_x^2+\sigma_y^2)}} e^{-\frac{(x'-\mu_x)^2}{2(\sigma_x^2+\sigma_y^2)} + \frac{(x'-\mu_x)^2}{2\sigma_x^2}} \\ &= \frac{\sigma_x}{\sqrt{(\sigma_x^2+\sigma_y^2)}} e^{\frac{\sigma_y^2(x'-\mu_x)^2}{2\sigma_x^2(\sigma_x^2+\sigma_y^2)}} \\ &= \frac{c}{\sqrt{1+c^2}} e^{\frac{(x'-\mu_x)^2}{2\sigma_x^2(1+c^2)}} \text{ where } \frac{\sigma_x}{\sigma_y} = c. \end{aligned} \tag{A.10}$$

Substituting Eqn.(A.8) into Eqn.(A.10),

$$\begin{aligned} f_{Z'}(z') &= \frac{c}{\sqrt{1+c^2}} e^{\frac{(x'-\mu_x)^2}{2\sigma_x^2(1+c^2)}} \\ &= \frac{c}{\sqrt{1+c^2}} e^{\frac{(\text{erf}^{\text{inv}}(2z'-1))^2}{(1+c^2)}} \\ &= \frac{c}{\sqrt{1+c^2}} e^{\frac{(\text{erfc}^{\text{inv}}(2z'))^2}{(1+c^2)}}. \end{aligned}$$

2. Next we derive a closed formula for the quantity $f_{ZZ'}$.

$$f_{ZZ'}(z,z') = \frac{f_Y(F^{\text{inv}}(z') - F^{\text{inv}}(z))}{f_X(F^{\text{inv}}(z'))} \text{ By Theorem 4.4.4} \tag{A.11}$$

Substituting the formula for F^{inv} into Eqn.(A.11) and simplifying,

$$\begin{aligned} f_{ZZ'}(z,z') &= \frac{\sigma_x}{\sigma_y} e^{\frac{-\sigma_x^2\left(\text{erfc}^{\text{inv}}(2z) - \text{erfc}^{\text{inv}}(2z')\right)^2}{\sigma_y^2} + (\text{erfc}^{\text{inv}}(2z))^2} \\ &= ce^{-c^2\left(\text{erfc}^{\text{inv}}(2z) - \text{erfc}^{\text{inv}}(2z')\right)^2 + (\text{erfc}^{\text{inv}}(2z))^2}. \quad \square \end{aligned}$$

Relationship between ℓ_1 and ℓ_2 norm

- For $v \in \mathbb{R}^k$, $\sum_{i=1}^k |v_i| \leq \sqrt{k} \left(\sum_{i=1}^k v_i^2\right)^{\frac{1}{2}}$[33].
  Hence,
$$\sum_{i=1}^k |p_i - \hat{p}_i| \leq \sqrt{k} \left(\sum_{i=1}^k |p_i - \hat{p}_i|^2\right)^{\frac{1}{2}}.$$

- Let $a_1, \ldots, a_n$, be a sequence of non negative numbers, then
$$\sum_{i=1}^n a_i^2 \leq \left(\sum_{i=1}^n a_i\right)^2$$
$$\sum_{i=1}^k (p_i - \hat{p}_i)^2 = \sum_{i=1}^k |p_i - \hat{p}_i|^2 \leq \left(\sum_{i=1}^k |p_i - \hat{p}_i|\right)^2$$
  Hence,
$$\sum_{i=1}^k |p_i - \hat{p}_i| \quad \geq \quad \left(\sum_{i=1}^k |p_i - \hat{p}_i|^2\right)^{\frac{1}{2}}. \qquad \square$$

Proof of Theorem 5.1.3

We derive expressions for the mean and variance of $D_{P,\hat{P}}$.

1. We compute $\mathbb{E}[D_{P,\hat{P}}]$
$$\begin{aligned}\mathbb{E}[D_{P,\hat{P}}] = \mathbb{E}\left[\sum_i (p_i - \hat{p}_i)^2\right] &= \mathbb{E}\left[\sum_i p_i^2 - \sum_i 2 p_i \hat{p}_i + \sum_i \hat{p}_i^2\right] \\ &= \sum_i p_i^2 - 2\sum_i p_i \mathbb{E}[\hat{p}_i] + \sum_i \mathbb{E}[\hat{p}_i^2] \\ &= \frac{1}{N}(1 - \sum_i p_i^2), \text{ Using } \mathbb{E}[\hat{p}_i] = p_i \text{ and } \mathbb{E}[\hat{p}_i^2] = p_i^2 + \frac{p_i(1-p_i)}{N}.\end{aligned}$$

2. Recall that, $\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]$.

   Hence,
$$\mathrm{Var}\left[\sum_i (p_i - \hat{p}_i)^2\right] = \mathbb{E}\left[\left(\sum_i (p_i - \hat{p}_i)^2\right)^2\right] - \left(\mathbb{E}\left[\sum_i (p_i - \hat{p}_i)^2\right]\right)^2 \qquad \text{(A.12)}$$

$$
\begin{aligned}
\mathbb{E}\left[\left(\sum_i (p_i - \hat{p}_i)^2\right)^2\right] &= \mathbb{E}[\sum_i (p_i - \hat{p}_i)^4 + 2\sum_{i,j:i<j} (p_i - \hat{p}_i)^2 (p_j - \hat{p}_j)^2] \\
&= \mathbb{E}[\sum_i ({p_i}^4 - 4{p_i}^3\hat{p}_i + 6{p_i}^2\hat{p}_i^2 - 4p_i\hat{p}_i^3 + \hat{p}_i^4) + 2\sum_{i,j:i<j} (p_i - \hat{p}_i)^2 (p_j - \hat{p}_j)^2] \\
&= \sum_i ({p_i}^4 - 4{p_i}^3\mathbb{E}[\hat{p}_i] + 6{p_i}^2\mathbb{E}[\hat{p}_i^2] - 4p_i\mathbb{E}[\hat{p}_i^3] + \mathbb{E}[\hat{p}_i^4]) \\
&+ 2\sum_{i,j:i<j} \mathbb{E}[(p_i - \hat{p}_i)^2 (p_j - \hat{p}_j)^2] \\
&= \sum_i ({p_i}^4 - 4{p_i}^3\mathbb{E}[\hat{p}_i] + 6{p_i}^2\mathbb{E}[\hat{p}_i^2] - 4p_i\mathbb{E}[\hat{p}_i^3] + \mathbb{E}[\hat{p}_i^4]) \\
&+ 2\sum_{i,j:i<j} ({p_i}^2{p_j}^2 - 2{p_i}^2 p_j\mathbb{E}[\hat{p}_j] + {p_i}^2\mathbb{E}[\hat{p}_j^2] - 2p_i{p_j}^2\mathbb{E}[\hat{p}_j] + 4p_i p_j\mathbb{E}[\hat{p}_i\hat{p}_j] \\
&+ -2p_i\mathbb{E}[\hat{p}_i^2\hat{p}_j^2]{p_j}^2\mathbb{E}[\hat{p}_i^2] - 2p_j\mathbb{E}[\hat{p}_j\hat{p}_i^2] + \mathbb{E}[\hat{p}_j^2\hat{p}_i^2])
\end{aligned}
$$

The moment generating function, $M_N(\bar{t}) = \left(\sum_{i=1}^k p_i e^{t_i}\right)^N$, where $\bar{t} = (t_1, \ldots, t_k)$, is used to derive the expected value for higher orders of $\hat{p}_i$.

$$
\begin{aligned}
\mathbb{E}[\hat{p}_i] &= \mathbb{E}\left[\frac{N_i}{N}\right] = p_i && \text{(A.13)} \\
\mathbb{E}[\hat{p}_i^2] &= \frac{1}{N^2}\frac{\partial^2 M_N(\bar{t})}{\partial t_i^2}\bigg|_{\bar{t}=0} = p_i^2 + \frac{p_i(1-p_i)}{N} && \text{(A.14)} \\
\mathbb{E}[\hat{p}_i^3] &= \frac{1}{N^3}\frac{\partial^3 M_N(\bar{t})}{\partial t_i^3}\bigg|_{\bar{t}=0} = p_i^3 + \frac{(2-3N)p_i^3 + 3(N-1)p_i^2 + p_i}{N^2} && \text{(A.15)} \\
\mathbb{E}[\hat{p}_i^4] &= \frac{1}{N^4}\frac{\partial^4 M_N(\bar{t})}{\partial t_i^4}\bigg|_{\bar{t}=0} && \text{(A.16)} \\
&= p_i^4 + \frac{(-6N^2 + 11N - 6)p_i^4 + 6(N-1)(N-2)p_i^3 + 7(N-1)p_i^2 + p_i}{N^3} \\
\mathbb{E}[\hat{p}_i\hat{p}_j] &= p_i p_j - \frac{p_i p_j}{N} && \text{(A.17)} \\
\mathbb{E}[\hat{p}_i^2\hat{p}_j] &= \frac{1}{N^3}\frac{\partial}{\partial t_j}\frac{\partial^2 M_N(\bar{t})}{\partial t_i^2}\bigg|_{\bar{t}=0} = p_i^2 p_j + \frac{(2-3N)p_i^2 p_j + (N-1)(N-2)p_i p_j}{N^2} && \text{(A.18)} \\
\mathbb{E}[\hat{p}_i^2\hat{p}_j^2] &= \frac{1}{N^4}\frac{\partial^2}{\partial t_i^2}\frac{\partial^2 M_N(\bar{t})}{\partial t_j^2}\bigg|_{\bar{t}=0} && \text{(A.19)} \\
&= p_i^2 p_j^2 + \frac{(-6N^2 + 11N - 6)p_i^2 p_j^2 + (N-1)(N-2)(p_i^2 p_j + p_i p_j^2) + (N-1)p_i p_j}{N^3}
\end{aligned}
$$

Substituting the equations above into Eqn.(A.12) and simplifying, we have

$$
\mathrm{Var}\left[D_{P,\hat{P}}\right] = \mathrm{Var}\left[\sum_i (p_i - \hat{p}_i)^2\right] = \frac{2}{N^3}\left(\sum_i (Np_i^2 - p_i^2 + 4p_i^3 - 2Np_i^3) + (N-3)\left(\sum_i p_i^2\right)^2\right). \quad \square
$$

Proof of Corollary 5.2.1

Assuming that $N \geq 2$, we derive an upper bound for the quantity $\mathrm{Var}\left[D_{P,\hat{P}}\right]$.

$$\begin{aligned}
\mathrm{Var}\left[D_{P,\hat{P}}\right] &= \frac{2}{N^3}\left(\sum_i (Np_i^2 - p_i^2 + 4p_i^3 - 2Np_i^3) + (N-3)\left(\sum_i p_i^2\right)^2\right) \\
&\leq \frac{2}{N^3}\left(\sum_i (Np_i^2 - p_i^2) + (N-3)\left(\sum_i p_i^2\right)^2\right) \\
&\leq \frac{2}{N^3}\left(\sum_i (Np_i^2 - p_i^2) + N\left(\sum_i p_i^2\right)^2\right) \\
&\leq \frac{2}{N^3}\left(\sum_i Np_i^2 + N\left(\sum_i p_i^2\right)^2\right) \\
&\leq \frac{2}{N^3}\sum_i 2Np_i^2 \\
&= \frac{4}{N^2}\sum_i p_i^2. \qquad \square
\end{aligned}$$

Proof of Corollary 5.2.6

Let $N \geq \frac{2\sqrt{\sum_i p_i^2}}{\kappa\sqrt{\epsilon}} + \frac{1-\sum_i p_i^2}{\kappa}$, we prove that $\Pr[D_{P,\hat{P}} < \kappa] \geq 1-\epsilon$.
Let $\kappa > \mathbb{E}[D_{P,\hat{P}}]$.

$$\begin{aligned}
\Pr[D_{P,\hat{P}} \geq \kappa] &= \Pr[D_{P,\hat{P}} - \mathbb{E}[D_{P,\hat{P}}] \geq \kappa - \mathbb{E}[D_{P,\hat{P}}]] \\
&\leq \Pr[|D_{P,\hat{P}} - \mathbb{E}[D_{P,\hat{P}}]| \geq \kappa - \mathbb{E}[D_{P,\hat{P}}]] \\
&\leq \frac{\mathrm{Var}\left[D_{P,\hat{P}}\right]}{(\kappa - \mathbb{E}[D_{P,\hat{P}}])^2} \\
&\leq \frac{4\sum_i p_i^2}{N^2(\kappa - \mathbb{E}[D_{P,\hat{P}}])^2} \\
&\leq \epsilon, \text{ Because } N \geq \frac{2\sqrt{\sum_i p_i^2}}{\kappa\sqrt{\epsilon}} + \frac{1-\sum_i p_i^2}{\kappa}.
\end{aligned}$$

Hence we can conclude that if $N \geq \frac{2\sqrt{\sum_i p_i^2}}{\kappa\sqrt{\epsilon}} + \frac{1-\sum_i p_i^2}{\kappa}$, then $\Pr[D_{P,\hat{P}} < \kappa] \geq 1-\epsilon$. $\square$

A.2 Appendix B

Error correcting codes

In coding theory, a linear code C of length n and rank k is a linear subspace of the vector space $\mathbb{F}_q^n$ with dimension k.

The major parameters of a linear code C are

- The alphabet[1] size q.
- The block length n - number of symbols in the code word.
- The message length k - number of symbols in the message.
- minimum distance d - minimum distance between any two different code words.

The code C as defined above is called a q-ary $[n, k, d]$ linear code with cardinality q^k. The most common distance measure used in coding theory is the hamming distance[2].

Theorem B.1 [29] *A code with minimum distance d can correct at least $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors. If d is even, the code can simultaneously correct $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors and detect $\frac{d}{2}$ errors.*

Theorem B.2 Hamming bound [32] For any $[n, k, d]$ linear code over $\mathbb{F}_q$,

$$q^k \leq \frac{q^n}{\sum_{i=0}^{n} \binom{n}{i}(q-1)^i}, \quad \text{where } t = \lfloor \frac{d-1}{2} \rfloor.$$

Definition B.3 An $[n, k, d]$ code, which satisfies the Hamming bound with equality is referred to as a perfect code.

Definition B.4 [32] *A $k \times n$ matrix G with entries in $\mathbb{F}_q$ is called a generator matrix of an $\mathbb{F}_q$-linear code C if the rows of G are a basis of C.*

Some standard binary error correcting codes

In this section, $m, t \in \mathbb{Z}$.

- **Repetition code:** The repetition code exists for any length n and any alphabet. The most common repetition code is the binary repetition code. The binary repetition code of length n has two code words, $c_1 = c_{1_0}, \ldots, c_{1_{n-1}} = 0, \ldots, 0$ and $c_2 = c_{2_0}, \ldots, c_{2_{n-1}} = 1, \ldots, 1$. When n is odd the code is perfect.
- **Hamming code:** The Hamming code is a perfect code with parameters $[2m-1, 2m-m-1, 3]$. However, it can correct only one error.
- **Hadamard codes:** An $[n, k, d]$ Hadamard code has parameters $n = 2^m$, $k = m+1$ and $t = 2^{m-1}$. They are constructed from binary Hadamard matrices [3].
- **Reed-Muller codes:** The $RM(r, m)$ Reed-Muller is a code of length 2^m and order r. The 0^{th} order Reed-Muller code $RM(0, m)$ corresponds to the $[2^m, 1, 2^m]$ repetition code. The 1^{st} order Reed-Muller code $RM(1, m)$ corresponds to the $[2^m, m+1, 2^{m-1}]$ Hadamard code.
- **BCH codes:** The BCH codes are one of the popular codes used in coding theory. An $[n, k, d]$ BCH code has parameters, $n = 2^m - 1$, $k \geq 2^m - mt - 1$ and $d \geq 2t + 1$.
- **Golay codes:** The Golay codes are perfect linear error-correcting codes. There is a binary and a ternary version. The binary Golay code G_{23} has parameters [23, 12, 7] while the ternary code G_{11} has parameters [11, 6, 5].

[1] An alphabet is a finite set of symbols that is used to represent a code word.

[2] The Hamming distance $D(x, y)$ between $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ is given by $D(x, y) = |\{1 \leq i \leq n | x_i \neq y_i\}|$.

[3] A binary Hadamard matrix is a normalised Hadamard matrix. A Hadamard matrix H_n of order n is an $(n \times n)$ matrix with elements 1 and -1 such that: $H_n \cdot H_n^T = n \cdot I_n$, where I_n is an $n \times n$ identity matrix.