Fuzzy extractors for continuous distributions

Ileana Buhan, Jeroen Doumen, Pieter Hartel, Raymond Veldhuis,
EEMCS Faculty, University of Twente,
{Ileana.Buhan, Jeroen.Doumen, Pieter.Hartel, Raymond.Veldhuis}@utwente.nl

Abstract

We show that there is a direct relation between the maximum length of the keys extracted from biometric data and the error rates of the biometric system. The length of the bio-key depends on the amount of distinguishing information that can be extracted from the source data. This information can be used a-priori to evaluate the potential of the biometric data in the context of a specific cryptographic application. We model the biometric data more naturally as a continuous distribution and we give a new definition for fuzzy extractors that works better for this type of data.

1 Introduction

Databases with biometric information are a serious threat to the privacy of users. The ability to track users across multiple databases is an example of this threat. The usual solution of using different passwords in different systems does not apply for obvious reasons - a person only has a limited number of biometric identification available: ten fingers, two eyes, etc. If one of these is compromised nothing can be done to undo the harm. This means that the template of a user, which stores his biometric information, needs protection.

Template protection can be used to store securely a biometric identity of a user. Tracking is no longer possible if different template protection schemes are used in different databases. A protected template will reveal almost nothing about the biometric data. If by some means a database with secured biometric data is compromised, the attacker cannot learn anything about the biometric data. Moreover if such an intrusion is detected the biometric is not lost, since at any time the protection scheme can be reapplied on the original data.

As one needs measurements to obtain biometric data, another inherent problem with biometrics is noise. One cannot use biometric data directly as a password (or key), since classical cryptography cannot cope with the noisiness of the biometric data. Uniform and reproducible randomness is the main ingredient for a good password. Unfortunately, biometric measurements do not fit this directly. Template protection schemes can be applied as a transformation function on biometric data to make the password reproducible. By this transformation, biometrics can be used as passwords.

In the literature often the source of biometric data is considered to be either continuous or discrete. Therefore template protection schemes can be divided in two classes. Representatives of the first class are continuous source shielding functions [5], the reliable component scheme [7] and the multi-bit scheme [3]. The fuzzy vault [8] and the secure sketch [4] belong to the second class.

It is difficult to compare the performance of these schemes because there is no unified view on the evaluation strategy. All authors estimate the error rate of their system in terms of FAR and FRR , but when it comes to evaluating the security of the resulting binary sequence different authors have different opinions. Monrose et al. [6] compute the guessing entropy while Zhang et al. [9] try to estimate the number of effective bits in the resulting key and propose a weighting system for choosing the best combination. Chang et al. [3] analyze the security of a sketch by investigating the remaining entropy of the biometric data, given that the sketch is made public. The same approach is taken by [2].

Contribution. Fuzzy extractors [4] where proposed as a general model capable of describing any template protection scheme that assumes a discrete source initial data. In this paper we extend the scope of the classical fuzzy extractors to continuous source data. We propose CS-fuzzy extractors as a unifying view on template protection schemes. This give us new insights. We show that the length and the quality of the bio-key depends on the amount of distinguishing information that can be extracted from the initial data. This gives a bound on the number of uniformly distributed bits that can be extracted from a given set of data. This information can be used a-priori to evaluate the potential of the biometric data in the context of a specific cryptographic application.

2 Preliminaries

Before we delve into the differences between discrete and continuous source biometrics, we need to establish some background first. We start by giving our notations, as well as some basic definitions. Secondly, we introduce the fuzzy extractor for a discrete source as given by [2, 4]. Thirdly, we briefly discuss the chosen model of the continuous source and its implications. Lastly, we remind the reader of the definitions of biometric error rates common in the literature.

Notation and Definitions. We will use $\mathcal{U}_l$ to denote the set of uniformly distributed binary sequences of length l. When referring to keys extracted from biometric data we are interested in the probability that an adversary can guess the value of the key on the first try. The *min-entropy* or the *predictability* of a random variable X denoted by $H_\infty(X)$ is defined as the logarithm of the most probable element in the distribution: $H_\infty(X) = -\log_2(max_x P(X = x))$. The min-entropy tells us the number of nearly uniform bits that can be extracted from the variable X.

The Kolmogorov distance or *statistical distance* between two probability distributions A and B is defined as: $SD(A, B) = sup_v|Pr(A = v) - Pr(B = v)|$.

For modelling the process of randomness extraction from fuzzy data Dodis et al. [4] define the notion of a fuzzy extractor.

The purpose of a fuzzy extractor is to extract robustly a binary sequence $\mathtt{s}$ from a noisy measurement $\mathtt{w'}$ with the help of some public string $\mathtt{Q}$. This process is presented in figure 1. Enrollment is performed by a function *Gen*, that on input of the noise free biometric w and the binary string $\mathtt{s}$,will compute a public string $\mathtt{Q}$. The binary string $\mathtt{s}$ can be extracted from the biometric data itself as in the reliable component scheme, presented in more detail in section 3.5, or $\mathtt{s}$ can be generated independently as in [5]. The dotted lines in figure 1 illustrate these alternatives.

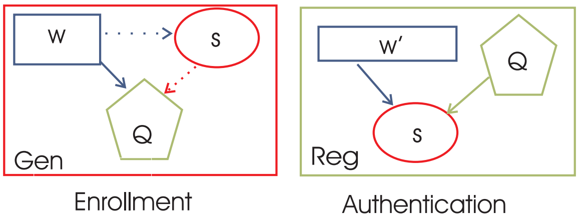

Figure 1: *A fuzzy extractor.*

For a discrete source $\mathcal{M}$ endowed with a metric d, the formal definition of a fuzzy extractor [2, 4] is:

Definition 1 (Fuzzy extractor) *An $(\mathcal{M}, m, l, t, \epsilon)$ fuzzy extractor is a pair of randomized procedures, $\langle Gen, Reg\rangle$, where:*

Gen *is a (necessarily randomized) generation function that on input $w \in \mathcal{M}$ extracts a* private string $\mathtt{s}\in \{0, 1\}^l$ *and a* public string $\mathtt{Q}$*, such that for all random variables W over $\mathcal{M}$ such that $H_\infty[W] \geq m$ and dependent variables $\langle \mathtt{s}, \mathtt{Q}\rangle \leftarrow Gen[w]$, it holds that $SD[\langle \mathtt{s},\mathtt{Q}\rangle, \langle U_l, \mathtt{Q}\rangle] \leq \epsilon$*

Reg *is a regeneration function that given a word $w' \in \mathcal{M}$ and a public string $\mathtt{Q}$ outputs a string $\mathtt{s} \in \{0, 1\}^l$, such that for any words $w, w' \in \mathcal{M}$ satisfying $d(w, w') \leq t$ and any possible pair $\langle \mathtt{s},\mathtt{Q}\rangle \leftarrow Gen[w]$, it holds that $\mathtt{s} = Reg[w', \mathtt{Q}]$*

During authentication, function *Reg* takes as input a noisy measurement w' and the public string $\mathtt{Q}$ and it will output the binary string $\mathtt{s}$ if w and w' come from the same user.

Distribution modelling. The biometric identity of a user is described by multiple features. We assume that the features are independent. For simplicity, in this paper all examples and definitions are presented for one feature only. The extension to higher dimensions is natural.

Let S_a (the subscript a meaning authentic) be the probability distribution that describes a user in the system. We denote with S_g the *probability distribution of the whole population*; in this case the subscript means global.

We use the Gaussian distribution for both S_a and S_g, since it represents a common model for real world raw data. The imposter distribution can then be written as $S_g = N(\mu_g, \sigma_g)$. Any *user distribution* S_a is described by a standard deviation σ_a and a mean μ_a drawn from $\mu_a \in N(\mu_g, \sigma_g - \sigma_a)$.

To estimate w, which represents the biometric identity of a user, multiple measurements are taken and a mean is estimated. The small perturbations between measurements hold important information. They represent an estimate on how far from the mean other genuine samples will be. We can call this information noise which can be represented as the standard deviation. This is used to establish suitable probabilities of value acceptance and rejection area.

A noise free biometric, in the case of a discrete distribution is denoted by w. When a continuous distribution is assumed, the closest to the notion of w is μ_a, the mean of the authentic distribution S_a. We will use μ_a when a noise free biometric template is computed from continuous source initial data and w when the initial data is discretely distributed.

For consistency, we use the same notation for a noisy measurement, w' both for discrete and continuous source data.

Error rates. The error rates of a biometric system are determined by the accuracy with which the matching engine can determine the similarity between a measured sample w' and the expected value μ_a of distribution S_a [1]. We can construct two possible hypotheses:

H_0 the measured w' is coming from the authentic user;

H_1 the measured w' is not coming from the authentic user;

The matching engine has to decide whether H_0 or H_1 is true. To express the accuracy of a biometric system the terms *false acceptance rate* (FAR) and*false rejection rate* (FRR) are used. The*false acceptance rate* is a type I error and represents the probability that H_0 will be accepted when in fact H_1 is true. The*false rejection rate* is a type II error and represents the probability that the outcome of the matching engine is H_1 but H_0 is true. In the setting of figure 2 we have a false acceptance every time another user, from the distribution S_g is generating a measurement which is in the acceptance region

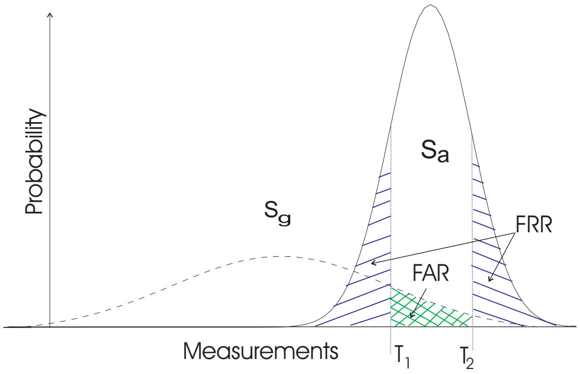

Figure 2: *Threshold $\langle T_1, T_2 \rangle$ determines acceptance and rejection regions*

described by the interval $\langle T_1, T_2 \rangle$. We can then write FAR $=\int_{T_1}^{T_2} pdf(S_g)dx$, where *pdf* stands for *probability density function*. Every time user S_a produces a sample that is in the rejection area, he will be rejected, thus FRR $=1 - \int_{T_1}^{T_2} pdf(S_a)dx$.

Dodis et al. [4] assume that the data source $\mathcal{M}$ is discrete for the definition of fuzzy extractor. However, the class of template protection schemes that uses continuous sources does not fit this model. Instead of trying to fit this class by implicitly discretizing the continuous source, the fuzzy extractor definition should be extended to model both classes. This is the subject of the next section.

3 Fuzzy extractors for continuous distributions

We show in this section that in the fuzzy extractor $(\mathcal{M}, m, l, t, \epsilon)$ there is a natural link between parameter m, the threshold t, the length of the resulting binary sequence l and ϵ the distance between the distribution of the key and the uniform distribution. For the cs-fuzzy extractors we choose slightly different parameters which are more natural for biometric data that are suited for continuous distributions.

3.1 From continuous to discrete sources

Definition 1 relies on a source $\mathcal{M}$ with min-entropy m. How can we construct a source with min-entropy m out of a continuous distribution like S_g? A common solution is to divide the measurement axis into intervals. To each interval d_i a discrete string s_i will be associated.

Example. In the setting of figure 3 the result of this division is the discrete distribution $D_g = \langle d_i \rangle, i = 1..n$. In figure 3, n is equal to 8. The public string $\mathtt{Q}$ contains the representation of the quantization. The

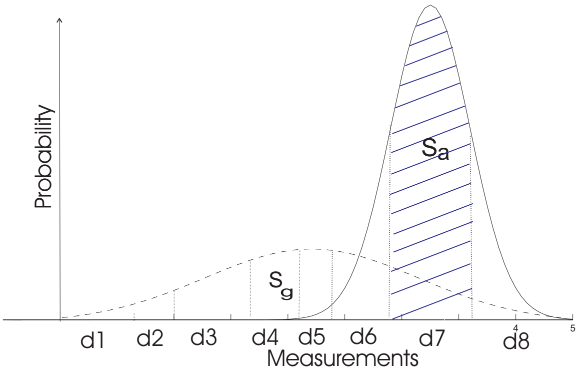

Figure 3: *Discretization of a continuous distribution*

probability of selecting an interval is computed as $p_i = Pr[D_g = d_i] = \int_{d_i} pdf(S_g|\mathsf{Q})dx$ where the integral is taken over the interval d_i. The continuous distribution S_g has been transformed into the discrete distribution $D_g = \langle d_i \rangle$, $i = 1, \ldots, n$ where n=8. A user S_a can be described by only one authentic interval. We denote with p_{auth} the probability associated to the authentic interval. We chose the authentic interval d_i for which the value $p_{auth} = \int_{d_i} pdf(S_a)dx$ is maximized since this describes best our user. In figure 3, d_7 best describes user S_a.

Now we are able to speak of the min-entropy of D_g denoted by m and defined as $m = -log_2 p_{max}$ where $p_{max} = max_i(Pr[D_g = d_i])$.

The effects of the discretization on the error rates, the FAR and the FRR are shown in figure 4. If we associate to user S_a the discrete variable d_i the FAR for this user will be equal to p_{auth}, in figure 4 the crosshatched area. The probability of a false rejection is determined by what is left from the distribution of S_a after removing p_{auth}, in figure 4 the FRR is the dashed area.

3.2 Relating min-entropy m and FAR

The above construction using the biometric data creates a tight relation between the min-entropy m of distribution D_g and the error rates of the biometric system. For the output sequence s to have a small chance of guessing the correct value from the first try we have to maximize the min-entropy by lowering the values of all the probabilities p_i. Unfortunately, by lowering p_i we increase the FRR .

Proposition 1 *For the above defined distribution D_g we have $m \leq -log_2 \mathrm{FAR}$ with equality when $p_{auth} = p_{max}$.*

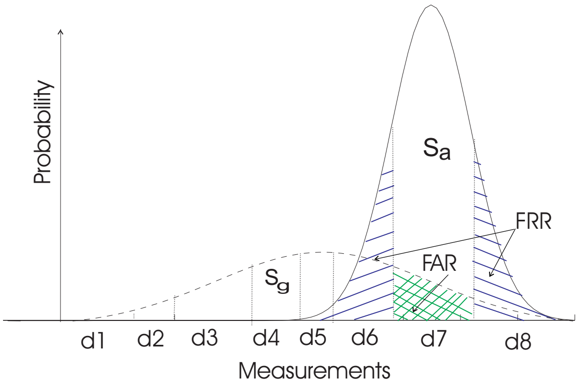

Figure 4: *Effects on the error rates of discretization of a continuous distribution*

Proof: We take $p_{max} = max_i p_i$. Since $p_{max} \geq p_{auth}$, we know that:

$$m = -log_2 p_{max} \leq -log_2 p_{auth} = -log_2 \mathrm{FAR}$$

Corrolary 1 $\mathrm{FAR} \leq 2^m$ *with equality when* $p_{auth} = p_{max}$.

Fact: m is maximized when the probabilities associated with the discrete distribution D_g are uniform.

3.3 Parameters t and FRR

According to definition 1 the *Reg*$[w', \mathbb{Q}]$ procedure will output the same binary sequence s as *Gen*$[w]$ whenever w and w' are close. The idea behind closeness is that w and w' probably belong to the same user. In definition 1 this is written as $d(w, w') < t$, where d is some metric, for example the Euclidian distance or the set difference metric. The value of t is a number. This value as such, does not say anything about the acceptance or the rejection probability of a user which, we feel, is more relevant. Also a suitable metric is not always available in the case of continuous sources.

The probability of correctly identifying that two measurements belong to the same user is the opposite of a type II error, thus the detection probability $P_d = 1 - \mathrm{FRR}$ is a suitable generalization of the threshold t.

3.4 Relating min-entropy and l to ϵ

We show in this section that given the number of bits l that we want to extract, and the min-entropy, $m = H_\infty(D_g)$ for a feature we can estimate ϵ, the distance of the output sequence distribution to the uniform distribution.

We are interested in the statistical distance between the ideal distribution of s where the generated key is distributed uniformly, i.e. in U_l, and the actual distribution of $\mathtt{s}$ given the helper data $\mathtt{Q}$.

$$
\begin{aligned}
\epsilon &= SD[\langle \mathtt{S}, Q\rangle, \langle U_l, Q\rangle] \\
&= \sup_{\mathtt{s}} |P(\mathtt{s} \in S | \mathtt{Q} \in Q) - P(\mathtt{s} \in U_l | \mathtt{Q} \in Q)|
\end{aligned}
$$

Looking at the last term, since the uniform distribution is independent of the helper data, we can write

$$
P(\mathtt{s} \in U_l | \mathtt{Q} \in Q) = P(\mathtt{s} \in U_l) = 2^{-l}.
$$

Introducing the notation $P(\mathtt{s}, \mathtt{Q}) := P\,(\mathtt{s} \in S | \mathtt{Q} \in Q)$, this gives

$$
\begin{aligned}
\epsilon &= \sup_{\mathtt{s}} \left| P(\mathtt{s}, \mathtt{Q}) - 2^{-l} \right|. \\
&= \max_{\mathtt{s}} \begin{cases} \sup_{\mathtt{s}} (P(\mathtt{s}, \mathtt{Q}) - 2^{-l}) & \text{when} \quad P(\mathtt{s}, \mathtt{Q}) \geq 2^{-l} \\ \sup_{\mathtt{s}} (2^{-l} - P(\mathtt{s}, \mathtt{Q})) & \text{when} \quad P(\mathtt{s}, \mathtt{Q}) < 2^{-l} \end{cases}
\end{aligned}
$$

Note that the true value of ϵ will be the largest of these two cases. Studying the first case, we get

$$
\sup_{\mathtt{s}} \left(P(\mathtt{s}, \mathtt{Q}) - 2^{-l}\right) = \left(\sup_{\mathtt{s}} P(\mathtt{s}, \mathtt{Q})\right) - 2^{-l} = 2^{-m} - 2^{-l},
$$

while in the second case we get

$$
\sup_{\mathtt{s}} \left(2^{-l} - P(s, \mathtt{Q})\right) = 2^{-l} - \inf_{\mathtt{s}} (P(\mathtt{s}, \mathtt{Q})) \leq 2^{-l},
$$

with equality when there exists a key sequence that is never attained. If we compare the two cases, we see that the first case represents the value of ϵ if $2^{-m} - 2^{-l} > 2^{-l}$, i.e. when $m \leq l - 1$.

To conclude, this shows that ϵ can be bounded from above in terms of the min-entropy m and l as follows:

$$
\epsilon \leq \epsilon(m, l) = \begin{cases} 0 & \text{if} \quad m = l, \\ 2^{-l} & \text{if} \quad l - 1 < m < l, \\ 2^{-m} - 2^{-l} & \text{if} \quad m \leq l - 1. \end{cases}
$$

3.5 CS-fuzzy extractors

The above relations lead us to the following definition of the fuzzy extractors for continuous sources.

Definition 2 *An* $(S_g, m, l, \mathrm{FRR})$ cs-*fuzzy extractor (continuous source fuzzy extractor) for the user distribution* S_a *is a pair of randomized procedures, "generate" (Gen) and "regenerate" (Reg) with the following properties:*

Gen *is a (necessarily randomized) generation function that on an input S_a extracts a* private string $\mathtt{s} \in \{0,1\}^l$ *and a* public string $\mathtt{Q}$*, such that for any user distribution S_a if $\langle \mathtt{s}, \mathtt{Q} \rangle \leftarrow Gen[S_a]$ then $SD[\langle \mathtt{s},\mathtt{Q} \rangle, \langle U_l, \mathtt{Q} \rangle] \leq \epsilon(m,l)$, where $\epsilon(m,l)$ is defined in section 3.4.*

Reg *is a regeneration function that given a measurement u' sampled from S_a and a public string $\mathtt{Q}$ outputs a string $\mathtt{s} \in \{0,1\}^l$, $\mathtt{s} = Reg[u', \mathtt{Q}]$, where $\langle \mathtt{s},\mathtt{Q} \rangle \leftarrow Gen[S_a]$, with probability equal to the detection probability, $P_d = 1 - \mathrm{FRR}$.*

Since cs-fuzzy extractors preserve the mechanism of the generate and regenerate functions as was used in the original fuzzy extractors, any fuzzy extractor is also a cs-fuzzy extractor. The link between the used parameters in each model was described in the preceding sections. However, cs-fuzzy extractors are better suited to handle continuous source biometric data.

4 CS-fuzzy extractors in practice

In order to demonstrate the usefulness of the cs-fuzzy extractor, we take three prominent template protection schemes for continuous distributions from the literature and fit them in our model. As we discussed earlier, these template protection schemes cannot be described in terms of classical fuzzy extractors.

4.1 Reliable component scheme

One of the most intuitive schemes in the area of template protection was proposed by Tuyls et al. [7] and is known as the *reliable component scheme*. We briefly describe this scheme for 1 user.

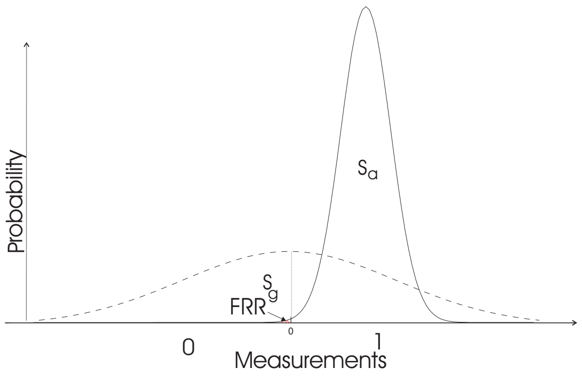

Figure 5: Example 1 Reliable component scheme

Gen During enrollment M samples $\langle w_1, w_2, ..w_M \rangle$ are measured. This is followed by quantization, where a sequence $\langle q_1, q_2, ..q_M \rangle$ is computed.

Here, each measured value w_j, $j = 1..M$ is compared to the imposter mean μ_g, in figure 5, μ_g is equal to 0. If $w_j \leq \mu_g$ then $q_j = 0$ else $q_j = 1$. A feature is called reliable if all q_j, $j = 1..M$ are equal. Only in that case will the feature be used. The public string $\mathtt{Q}$ represents the positions of the reliable components.

Reg During authentication, a noisy version of w, w' is measured. For each reliable component (we look at $\mathtt{Q}$) its value is compared to μ_g. The result of this comparison will represent the key.

This scheme will extract 1 bit from every reliable component, with probability equal to 1-FRR . Robustness is assured by the fact that only the features with very small FRR are chosen, as can be seen in figure 5. We can write the reliable component as a $(S_g, 1, 1, \mathrm{FRR})$ *cs-fuzzy extractor*, where

$$\mathrm{FRR} = \begin{cases} \int_{-\infty}^{\mu_g} e^{\frac{-(x-\mu_a)^2}{2\sigma_a}} dx, & \mu_a > \mu_g \\ \int_{\mu_g}^{\infty} e^{\frac{-(x-\mu_a)^2}{2\sigma_a}} dx, & \mu_a < \mu_g. \end{cases}$$

The output bit is uniformly distributed, because the probability of a bit being equal to 0 is equal to the probability of the same bit being a 1. The main merit of this scheme is its robustness. The disadvantage is that there are many features that are disregarded and depending on the quality of the data used the total length of the output key is rather short.

4.2 Shielding functions

Linnartz et al. [5] were among the first to suggest how to get keys from continuously distributed sources. Their technique is inspired by watermarking. They propose a multiple quantization level system with odd-even bands, see figure 6.

Gen For one feature, the bit $\mathtt{s}$ is embedded by shifting the mean w of the template distribution to the center of the closest even-odd q interval if the value of the key bit $\mathtt{s}$ is a 1, or to the center of the closest odd-even q interval if the value of the key bit $\mathtt{s}$ is a 0.

The public string $\mathtt{Q}$, called helper data is computed:

$$\mathtt{Q} = \begin{cases} (2n + \frac{1}{2})q - w & \text{when} \quad \mathtt{s} = 1 \\ (2n - \frac{1}{2})q - w & \text{when} \quad \mathtt{s} = 0 \end{cases}$$

Where $n \in \mathbb{Z}$ and is chosen such that: $-q < \mathtt{Q} < q$.

Reg is defined as:

$$Reg[w', \mathtt{Q}] = \begin{cases} 1, & \text{when} \quad 2nq \leq w' + \mathtt{Q} < (2n+1)q \\ 0, & \text{when} \quad (2n-1)q \leq w' + \mathtt{Q} < 2nq \end{cases}$$

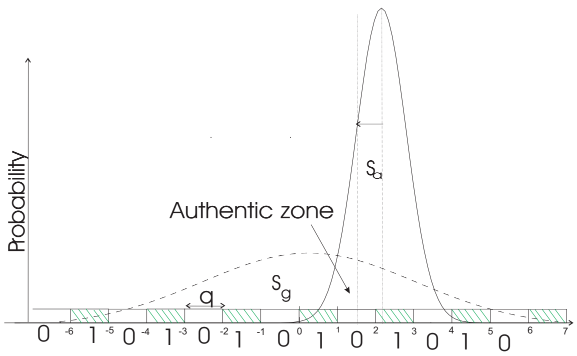

Figure 6: Shielding function discretization, embedding a 0 value key bit.

During authentication a noisy feature w' is extracted. The key bit is 1 if the sum of the noisy feature and the helper data is in an odd-even interval and is 0 otherwise. Whenever the measured value has an error greater than $\frac{q}{2}$ we can get an error in the key computation.

This scheme can be written as a:

$$(S_g, 1, 1, \mathrm{FRR}) \textit{ cs-fuzzy extractor} \text{ where}$$
$$\mathrm{FRR} = \sigma_a \, 2\sqrt{2} \sum_{i=0}^{\infty} \int_{\frac{(1+4i)}{2\sqrt{2}} \frac{q}{\sigma}}^{\frac{(3+4i)}{2\sqrt{2}} \frac{q}{\sigma_a}} e^{-x^2} dx.$$

The FRR depends on the quantization step q. When q is larger, the noise tolerance is higher as well. On the other hand, if q is smaller, the FAR goes down. The output sequence is uniform in this scheme as well.

4.3 Chang multi-bit scheme

Chang et al. [3] extract multiple bits from each feature of a user. They select distinguishable features to generate more key bits from each feature. For each feature the left and the right boundaries, L and R are selected so that with high probability a measurement from any user falls in this interval.

Gen The selected FAR determines for each feature an authentic region, see figure 7, delimited by T_1, T_2. The whole region L, R is divided in segments that have a length equal to the segment determined by T_1 and T_2. A label is associated with each segment. It can happen that some redundant segments are added to the left and to the right of L respectively R to use all labels of a given length. In figure 7 three more segments with the labels 000, 100 and 011 can be added. In the picture the genuine interval has label 101. The public string Q contains the description of the intervals and the associated labels.

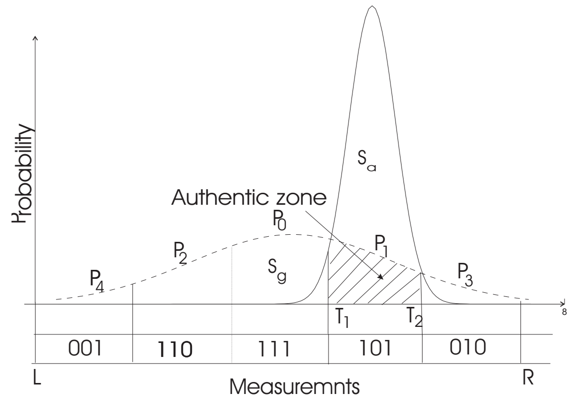

Figure 7: Chang discretization

Reg Every time a user submits his biometric data to the system his feature will fall in one of the published intervals. The label associated with this interval represents the key of this user. An authentic user will be in the authentic area with probability 1-FRR .

This process is repeated for every user, for every feature. Thus they have defined an $(S_g, m, l, \mathrm{FRR})$ where $m = log_2 \int_{\mu_g - \frac{|T_2-T_1|}{2}}^{\mu_g + \frac{|T_2-T_1|}{2}} pdf(S_g)dx$ and $l = log_2 \frac{|L-R|}{|T_2-T_1|}$. The mathematical relation for FRR is $1 - \int_{T_1}^{T_2} pdf(S_g)dx$.

Comments on the distinguishable components. To generate stable cryptographic keys Chang et al. [3] propose to use only the distinguishable features for key generation. We show that in this case choosing the distinguishable feature makes life easier for an intruder and in a particular case the intruder can almost certainly guess the authentic feature on the first try.

A feature is called distinguishable if the distance between the imposter mean and the authentic mean is sufficiently large. In the original paper a feature is distinguishable if $|\mu_g - \mu_a| > k_a \cdot \sigma_a$. In this scheme the authentic mean μ_a, due to the construction is always at the center of the authentic interval.

The goal of an intruder trying to attack this scheme is to find the authentic interval with a minimal number of trials. We model two types of intruders. It is assumed that the first imposter or the *type one imposter* knows the distribution of the population (S_g). The second type intruder called, *type II* also knows the parameters L and R.

Type I The intruder knows that the authentic area of a user is far away from the global mean. In this case he can safely disregard the segment where the mean is situated. This leaves a new probability distribution $\frac{p_1}{1-p_0}, .. \frac{p_n}{1-p0}$. as the central segment falls out.

Type II This attacker knows not only S_g but he also knows the values of L and R. In Chang et al. [3] these limits are computed as follows:

$$\begin{aligned} L &= min(\mu_g - k_g\sigma_g, \mu_a - k_a\sigma_a) \\ R &= min(\mu_g + k_g\sigma_g, \mu_a + k_a\sigma_a) \end{aligned}$$

Here k_g and k_a are natural numbers chosen by the designers of the system. For example the author recommends for k_g the value 5 so that it covers almost the entire user distribution.

If the margin L (and the reasoning is the same for R) is somewhere situated in the right half of a segment we can safely eliminate that segment. According to the definition L will always be smaller then μ_a, which is in the middle of an interval. Thus the attacker can eliminate all intervals for which the middle value is smaller then L.

Example In figure 8 we show how dangerous this combination can be. Assume the imposter distribution is divided in 4 intervals $\langle d_1, d_2, d_3, d_4\rangle$. These intervals are published as helper data. The imposter has to guess which interval is the authentic one. It is assumed that the imposter distribution is known to the attacker.

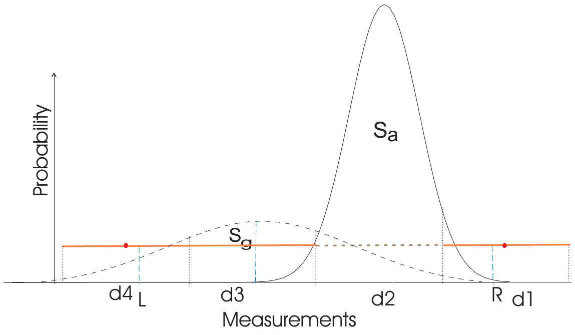

Figure 8: *The genuine interval can be guessed from one try.*

The attacker can eliminate interval number d_3 because it contains the global mean μ_g and he knows that a distinguishable feature should be far away from the global mean.

A type I attacker has 3 candidates for the correct authentic interval. However, the three intervals have different probabilities associated so the order of guessing will be: d_2, d_4, d_1. In this case he is lucky from the first trial.

A type II attacker also knows the value of L and R. The authentic mean is situated at the center of the authentic interval. The interval d_4 cannot be the authentic one since its middle value is smaller then L. Thus the attacker can eliminate d_4. The same reasoning holds for R which eliminates d_1.

As a result the intruder now has only one candidate for the authentic interval, namely d_2.

5 Conclusion and Future Work

Fuzzy extractors are a theoretical tool for modelling and comparing template protection schemes which use a discrete source. We generalize the definition to cs-fuzzy extractors, which can also handle the continuous source cases. Our model can cope with both classes of template protection schemes. Biometric authentication systems are evaluated using the false acceptance rate and the false rejection rate. The link between the two was hitherto not obvious even though they refer to the same data.

In this paper we show, for the first time that there is a natural connection between the false acceptance rate, false rejection rate and the parameters used to evaluate a template protection scheme implemented on the same data.

We also show that the error rates have a direct influence on the length and robustness of the key extracted from the features of a user.

In this paper we only consider the one dimensional case. However, biometric data contains multiple features for each user. For generalizing to multiple independent features, one approach is to analyze each dimension independently. In this case, the relationship between the min-entropy and the FAR is as expected: the more dimensions we have, the lower the FAR is and the number of bits that can be extracted increases. However, the FRR increases with the number of dimensions that are used. Therefore, this may not be the best approach for aggregating multiple features. Zhang et al [9] propose a better approach which can reduce both the FAR and the FRR by simultaneously analyzing all dimensions.

As future work we want to investigate the influence of feature aggregation on the length and robustness of the key.

References

[1] Ruud Bolle, Jonathan Connell, Sharanthchandra Pankanti, Nalini Ratha, and Andrew Senior. *Guide to Biometrics*. SpringerVerlag, 2003.

[2] Xavier Boyen. Reusable cryptographic fuzzy extractors. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors, *ACM Conference on Computer and Communications Security*, pages 82–91. ACM, 2004.

[3] Yao-Jen Chang, Wende Zhang, and Tsuhan Chen. Biometrics-based cryptographic key generation. In *ICME*, pages 2203–2206. IEEE, 2004.

[4] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer, 2004.

[5] Jean-Paul M. G. Linnartz and Pim Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In Josef Kittler and Mark S. Nixon, editors, *AVBPA*, volume 2688 of *Lecture Notes in Computer Science*, pages 393–402. Springer, 2003.

[6] Fabian Monrose, Michael K. Reiter, Qi Li, and Susanne Wetzel. Cryptographic key generation from voice. In *IEEE Symposium on Security and Privacy*, pages 202–213, 2001.

[7] Pim Tuyls, Anton H. M. Akkermans, Tom A. M. Kevenaar, Geert Jan Schrijen, Asker M. Bazen, and Raymond N. J. Veldhuis. Practical biometric authentication with template protection. In Takeo Kanade, Anil K. Jain, and Nalini K. Ratha, editors, *AVBPA*, volume 3546 of *Lecture Notes in Computer Science*, pages 436–446. Springer, 2005.

[8] Umut Uludag, Sharath Pankanti, and Anil K. Jain. Fuzzy vault for fingerprints. In Takeo Kanade, Anil K. Jain, and Nalini K. Ratha, editors, *AVBPA*, volume 3546 of *Lecture Notes in Computer Science*, pages 310–319. Springer, 2005.

[9] Wende Zhang, Yao-Jen Chang, and Tsuhan Chen. Optimal thresholding for key generation based on biometrics. In *ICIP*, pages 3451–3454, 2004.