Int. J. Inf. Secur. (2010) 9:193–208
DOI 10.1007/s10207-010-0103-4

SPECIAL ISSUE PAPER

Embedding renewable cryptographic keys into noisy data

Ileana Buhan · Jeroen Doumen · Pieter Hartel · Qian Tang · Raymond Veldhuis

Published online: 16 March 2010

Abstract A fuzzy extractor is a powerful but theoretical tool that can be used to extract uniform strings from (discrete) noisy sources. However, when using a fuzzy extractor in practice, extra features are needed, such as the renewability of the extracted strings and the ability to use the fuzzy extractor directly on continuous input data instead of discrete data. Our contribution is threefold. Firstly, we propose a *fuzzy embedder* as a generalization of the fuzzy extractor. A fuzzy embedder naturally supports renewability, as it allows a string to be embedded instead of extracted. It also supports direct analysis of quantization effects, as it makes no limiting assumptions about the nature of the input source. Secondly, we give a general construction for fuzzy embedders based on the technique of quantization index modulation (QIM). We show that the performance measures of a QIM, as proposed by the watermarking community, translate directly to the security properties of the corresponding fuzzy embedder. Finally, we show that from the perspective of the length of the embedded string, quantization in two dimensions is optimal. We present two practical constructions for a fuzzy embedder in two-dimensional space. The first construction is optimal from reliability perspective, and the second construction is optimal in the length of the embedded string.

Keywords Biometrics · Cryptographic keys · Sphere packing

I. Buhan (✉)
Information and System Security, Philips Research Laboratories, High Tech Campus 34, Room 6.33, Eindhoven, The Netherlands
e-mail: ileana.buhan@philips.com

J. Doumen
Irdeto Research, Flight Forum 89, 5657 DC, Eindhoven, The Netherlands

P. Hartel · Q. Tang
Distributed and Embedded Security, Faculty of EEMCS, University of Twente, P.O. Box 217, 7500 AE, Enschede, The Netherlands

R. Veldhuis
Signals and Systems Group, Electrical Engineering, University of Twente, P.O. Box 217, 7500 AE, Enschede, The Netherlands

1 Introduction

Cryptographic protocols rely on exactly reproducible key material. In fact, these protocols are designed to have a wildly different output if the key is only perturbed slightly. Unfortunately, exactly reproducible keys are hard to come by, especially when they also need to have sufficient entropy. For example, one can hardly expect an average user to remember a password that consists of a string of 128 random bits. Luckily, it is relatively easy to find "fuzzy" sources, such as physically uncloneable functions (PUFs) [23] and biometrics [12]. However, such sources are inherently noisy and rarely uniformly distributed. The first, main difficulty in using the output of a fuzzy source as key material is the noise, which has to be corrected to produce the same key every time. To solve this problem, the notion of a secure sketch [18] has been proposed. The second difficulty lies in the fact that this output may have a non-uniform distribution, while it should be as close to uniform as possible to serve as a cryptographic key. A strong randomness extractor could be used to turn the reproducible output into a nearly uniform string. This naturally leads to the notion of a fuzzy extractor [12], which gives a reproducible, nearly uniform string as output. A common way of constructing fuzzy extractors is to combine a secure sketch with a strong randomness extractor.

However, when deploying a fuzzy extractor in practice, more difficulties arise. Firstly, even with the same input, it should be possible to generate many different keys. This is paramount when considering biometrics, where the number of possible inputs is limited (two eyes, 10 fingers etc.). To achieve renewability of the cryptographic key, the (fixed) output of the fuzzy extractor must be randomized, for instance by using a common reference string. Unfortunately, this falls outside the scope of the fuzzy extractor, even though it is recognized as an important and sensitive issue [3].

Secondly, the definition of a fuzzy extractor only accepts discrete sources as input. Existing performance measures for secure sketches, such as entropy loss or min-entropy, lose their relevance when applied to continuous sources [18]. This limitation can be overcome by quantizing the continuous input. Li et al. [18] propose to define relevant performance measures with respect to the chosen quantization method. We argue that, instead of defining performance only after quantization, it is better to integrate the quantization into the definition, so that the intricacies of a continuous input can be studied.

CONTRIBUTIONS Our contribution is threefold. Firstly, we propose a new primitive called a *fuzzy embedder*, which is a natural extension of a fuzzy extractor. A fuzzy embedder provides a randomized output and handles arbitrary input sources.

The survey of template protection schemes presented by Uludag et al. [29] divides known template protection schemes into two categories. The first category consists of constructions that extract a cryptographic key from a noisy input. Such constructions are elegantly formalized by the notion of a fuzzy extractor. The second category consists of constructions that "bind" a cryptographic key to a noisy input. For this category, only practical constructions are known, whereas formal models do not exist. The notion of a fuzzy embedder fills this important gap. A fuzzy embedder can be regarded as an extension of a fuzzy extractor, since it can embed a fixed string (for instance one obtained by applying a strong extractor to the input source) into a discrete source and thus achieve the same functionality, namely a randomized cryptographic key.

Interestingly, the fuzzy commitment [16] has a direct relation to a fuzzy embedder as well: removing the binding property from a fuzzy commitment scheme yields a fuzzy embedder, which suggests that a fuzzy commitment is more general than a fuzzy embedder.

Secondly, we propose a general construction for a fuzzy embedder, using data hiding techniques from the watermarking community. Our construction is based on quantization index modulation (`QIM`), which is a watermarking method that can achieve efficient trade-offs between the information embedding rate, the sensitivity to noise, and the distortion [10]. The construction of a fuzzy embedder is intuitive as most of the properties of a fuzzy embedder can be reduced directly to the properties of the underlying `QIM`. The trade-offs of the used `QIM` give rise to similar trade-offs in fuzzy embedder performance measures. In this setting, shielding functions [19] can be regarded as a particular construction of a fuzzy embedder, as they focus on one particular type of quantizer. However, they only consider one-dimensional inputs.

Thirdly, we investigate different quantization strategies for high dimensional data, and we show that quantization in two dimensions gives an optimal length of the embedded uniform string. Finally, we focus on the two-dimensional case and give two practical constructions, one being optimal from the perspective of sensitivity to noise and the other being optimal from the key length perspective.

2 Related work

Reproducible randomness is the main ingredient of a good cryptographic system. Good-quality uniform random sources are rare compared to the more common non-uniform sources. For example, biometric data is easily obtainable, high entropy data. However, biometric data is not uniformly distributed, and its randomness cannot be exactly reproduced. Depending on the source properties, several constructions have been proposed for obtaining cryptographic keys from noisy sources.

Dodis et al. [12] consider discrete distributed noise and propose fuzzy extractors and secure sketches for different error models. These models are not directly applicable to continuously distributed sources. Linnartz et al. [19] construct shielding functions for continuously distributed data and propose a practical construction which can be considered a one-dimensional `QIM` scheme. The same approach is taken by Li et al. [18] who propose quantization functions for extending the scope of secure sketches to continuously distributed data. Buhan et al. [6] analyze the achievable performance of such constructions given the quality of the source in terms of the false acceptance rate and false rejection rate of a biometric system.

The process of transforming a continuous distribution into a discrete distribution influences the performance of the overall system, which uses fuzzy extractors and secure sketches. Quantization is the process of replacing analog samples with approximate values taken from a finite set of allowed values. The basic theory of one-dimensional quantization is reviewed by Gersho [14]. The same author investigates the influence of high dimensional quantization on the performance of digital coding for analog sources [15]. `QIM` constructions are used by Chen and Wornell [10] in the context of watermarking. The same authors introduce dithered quantizers [9]. Moulin and Koetter [22] give an excellent overview of `QIM` in the

general context of data hiding. Barron et al. [2] develop a geometric interpretation of conflicting requirements between information embedding and source coding with side information.

The concept of a fuzzy embedder might seem related to concepts developed in the context of information theoretic key agreement [20] more precisely to secure message transmission schemes based on correlated randomness [21]. However, the settings of the problem are different compared to ours. While in secure message transmission based on correlated randomness, the adversary and the legitimate participants have a noisy share of the same source data, in the fuzzy embedder setting the adversary does not have access to the data source.

ROADMAP The rest of this paper is organized as follows. In Sect. 4, we present the definition of a fuzzy embedder and highlight the differences with fuzzy extractors and fuzzy commitment. In Sect. 5, we propose a general construction of a fuzzy embedder from any QIM and express the performance in terms of the geometric properties of the underlying quantizers. In Sect. 6, we present two practical constructions for the quantization of two-dimensional space and compare the properties of these constructions with the existing square lattice packing. The last section concludes this paper.

3 Preliminaries

Before we delve into the differences between discrete and continuous source noisy data, we need to establish some background. We start by giving our notation, as well as some basic definitions. Second, we summarize the fuzzy extractor for a discrete source as given by Dodis et al. [12] and Boyen et al. [3]. Third, we briefly discuss the chosen model of the continuous source and its implications. Finally, we remind the reader of the definitions of error rates commonly used in the literature.

NOTATION Let M be an n-dimensional discrete, finite set, which together with a distance function $d_M : M \times M \to \mathbb{R}^+$ is a metric space. Similarly, let U be an n-dimensional continuous domain, which together with the distance $d_U : U \times U \to \mathbb{R}^+$ forms a metric space. When the domain is clear from the context, we use d and drop the subscript.

By capital letters we denote random variables, while small letters are used to denote observations of a random variable. Continuous random variables are defined over the metric space U, while a discrete random variable is defined over the metric space M. A random variable A is endowed with a probability density function $f_A(a)$.

We use the random variable P when referring to public sketch data and R for random binary strings, which can be used as cryptographic keys.

ENTROPY When referring to cryptographic keys, the strength of the key is measured as the min-entropy, i.e., the probability that an adversary predicts the value of the secret key from one attempt. The adversary's best strategy is to guess the most likely value. The *min-entropy* or the *predictability* of a random variable A denoted by $H_\infty(A)$ is defined as:

$$H_\infty(A) = -\log_2 \left(\max_{a \leftarrow A} \Pr(A = a) \right).$$

Min-entropy can be viewed as the "worst-case" entropy [12]. For two (possibly correlated) random variables A and B, the *average min-entropy* is defined as

$$\begin{aligned} \tilde{\mathbf{H}}_\infty(A|B) &= -\log \left(\mathbb{E}_{b \leftarrow B} \left[\max_{a \leftarrow A} Pr(A = a|B = b) \right] \right) \\ &= -\log \left(\mathbb{E}_{b \leftarrow B} \left(2^{-H_\infty(A|B=b)} \right) \right), \end{aligned}$$

which represents the remaining uncertainty about A given B or the amount of uncertainty left about variable A when variable B is made public [12] (both A and B are discrete random variables).

MUTUAL INFORMATION By $I(A; B)$, we note the Shannon mutual information between the two random variables A and B, which is a measure of the mutual dependence between two random variable, in the following sense: $I(A; B) = 0$ if and only if A and B are independent random distributed variables.

STATISTICAL DISTANCE The Kolmogorov distance or *statistical distance* between two probability distributions A and B with the same domain is defined as:

$$SD(A, B) = \sup_v |\Pr(A = v) - Pr(B = v)|.$$

Informally, this is the largest possible difference between the probabilities that the two probability distributions can assign to the same event.

FUZZY EXTRACTORS For modeling the process of randomness extraction from noisy data Dodis et al. [12] define the notion of a fuzzy extractor, see Fig. 1. A fuzzy extractor extracts a uniformly random string r from a value x of random variable X in a noise tolerant way with the help of some public sketch p.

The Generate procedure takes a non-uniformly random, noisy input x and produces two outputs: a public string p and a key r. The key r is uniformly random given p, and according to the definition of p, reveals no information about the input x. However, one can reproduce r exactly when both p and x' (close to x) are presented to the Reproduce procedure.

For a discrete metric space M with a distance measure d, the formal definition of a fuzzy extractor [3,12] is:

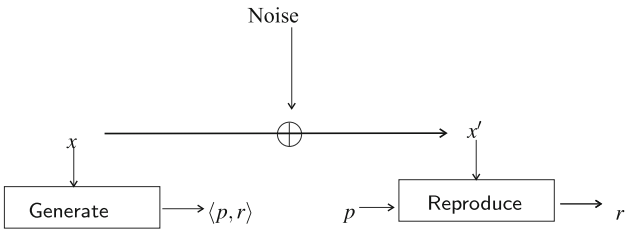

Fig. 1 A fuzzy extractor is a pair of two procedures Generate and Reproduce. The Generate procedure, which takes as input a noisy input x, is executed first. The result is a random sequence r and a public sketch p, which is made public. The Reproduce procedure, which takes as input x' that is corrupted by noise and the public sketch p, will output r if x and x' are close

Definition 1 (*Fuzzy extractor*) An $(M, m, l, t, \varepsilon)$ fuzzy extractor is a pair of randomized procedures, Generate and Reproduce, with the following properties:

1. The generation procedure on input of $x \in M$ outputs an extracted string $r \in R = \{0, 1\}^l$ and a public helper string $p \in P = \{0, 1\}^*$.
2. The reproduction procedure takes an element $x' \in M$ and the public string $p \in \{0, 1\}^*$ as inputs. The *reliability* property of the fuzzy extractor guarantees that if $d(x, x') \leq t$ and r, p were generated by $(r, p) \leftarrow \mathsf{Generate}(x)$, then $\mathsf{Reproduce}(x', p) = r$. If $d(x, x') > t$, then no guarantee is provided about the output of the reproduction procedure.
3. The *security* property guarantees that for any random variable X with distribution $f_X(x)$ of min-entropy m, the string r is nearly uniform even for those who observe p: if $(r, p) \leftarrow \mathsf{Generate}(X)$, then $SD((R, P), (N, P)) \leq \varepsilon$ where N is a random variable with uniform probability.

A fuzzy extractor is *efficient* if Generate and Reproduce run in polynomial time.

In other words, a fuzzy extractor allows to generate the random string r from a value x. The reproduction procedure that uses the public string p produced by the generation procedure will output the string r as long as the measurement x' is close enough. The *security* property guarantees that r looks uniformly random to an adversary, and her chance to guess its value from the first trial is approximately 2^{-m}. Security encompasses both *min-entropy* and *uniformity* of the random string r when p are known to an adversary. There are two shortcomings related to the above definition. Firstly, in the above definition $R = \{0, 1\}^l$ thus a random binary string of length l. The public string $P = \{0, 1\}^*$ which can be for example the syndrome of an error correcting code. However, there are template protection schemes that fit the model of the fuzzy extractors for which P is drawn from $\mathbb{R}$ [19] or $\mathbb{Z}$ [26]. Secondly, one can say that X has min-entropy only if it is a discrete probability density function otherwise its min-entropy depends on the precision or quantization used to represent the variable [18].

FUZZY COMMITMENT The definition of a fuzzy commitment scheme as by introduced Juels et al. [16] is:

Definition 2 (*Fuzzy commitment*) A *fuzzy commitment* scheme consists of a pair of two procedures Commit and Decommit defined as follows:

1. $\mathsf{Commit} : S \times X \rightarrow Y$ (run by the committer). The committer takes $s \in S$ and $x \in X$ as input and generates a committed string $y \in Y$.
2. $\mathsf{Decommit} : Y \times X \rightarrow S$ (run by the verifier). The verifier takes $y \in Y$ and $x \in X$ as input, and outputs a string $s \in S$ or an error $\perp$.
3. t-fuzziness: Suppose $c = \mathsf{Commit}(s, x)$ then $s = \mathsf{Decommit}(c, x')$ given $d(x, x') \leq t$.

In a commitment scheme, there are two parties: the committer and the verifier. The committer commits to a string s by sending some data c to the verifier. The committer can enable the verifier to check the committed value by sending (s, c) and other helper data. A commitment scheme is said to be *hiding* if it is infeasible for the verifier to obtain c without the help of the committer. A commitment scheme is said to be *binding* if it is infeasible for the committer to change the committed value. Note that, besides the t-fuzziness property, the verifier does not need s to run Decommit in a fuzzy commitment scheme.

QUANTIZATION A continuous random variable A can be transformed into a discrete random variable by means of quantization, which we write $Q(A)$. Formally, a quantizer is a function $Q : U \rightarrow M$ that maps each $a \in U$ into the closest *reconstruction point* in the set $M = \{c_1, c_2, \ldots\}$ by

$$Q(a) = \arg\min_{c_i \in M} d(a, c_i).$$

where d is the distance measure defined on U.

The *Voronoi region* or the *decision region* of a reconstruction point c_i is the subset of all points in U, which are closer, with respect to a specific distance measure, to that particular reconstruction point than to any other reconstruction point. We denote with V_{c_i} the Voronoi region of reconstruction point c_i. When A is one dimensional, Q is called a *scalar* quantizer. If all Voronoi regions of a quantizer are equal in both size and shape, the quantizer is *uniform*. In the scalar case, the length of the Voronoi region is then called the *step size*. If the reconstruction points form a lattice, the Voronoi regions of all reconstruction points are congruent.

By quantization, the probability density function of the continuous random variable A, $f_A(a)$, which is continuous, is transformed into the probability density function $f_{Q(A)}(a)$, which is discrete (see Fig. 2).

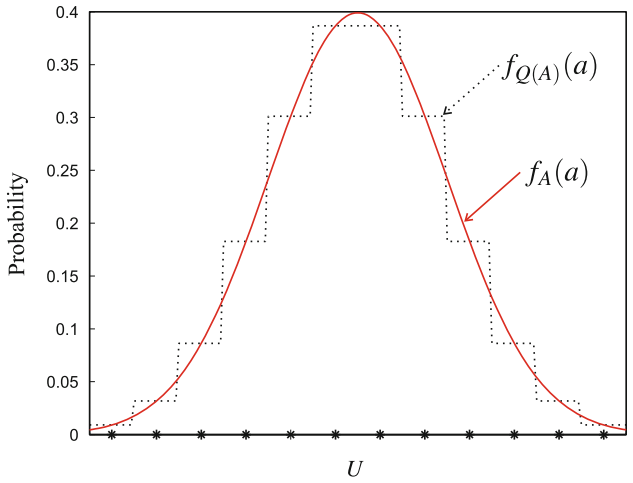

Fig. 2 By quantization, $f_A(a)$ (*continuous line*) is transformed into $f_{Q(A)}(a)$ (*dotted line*). We can write $Q(f_A(a)) = f_{Q(A)}(a)$.

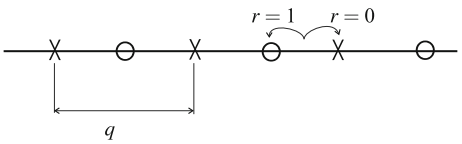

Fig. 3 Quantization of X with two scalar quantizers Q_0 and Q_1 both with step size q

QUANTIZATION-BASED DATA HIDING CODES Quantization-based data hiding codes as introduced by Chen et al. [10] (also known as quantization index modulation) can embed secret information into a real-valued quantity. We start with an example of the simplest case.

Example 1 We want to embed one bit of information, thus $r \in \{0, 1\}$ into a real value x. For this purpose, we use a scalar uniform quantizer with step size q, given by rounding $\frac{x}{q}$:

$$Q(x) = q\left\lfloor \frac{x}{q} \right\rceil.$$

The quantizer Q is used to generate a set of two new quantizers $\{Q_0, Q_1\}$ defined as:

$$Q_0(x) = Q(x + v_0) - v_0 \quad \text{and} \quad Q_1(x) = Q(x + v_1) - v_1$$

where

$$v_0 = \frac{q}{4} \quad \text{and} \quad v_1 = -\frac{q}{4}.$$

In Fig. 3, the reconstruction points for the quantizer Q_1 are shown as circles, and the reconstruction points for the quantizer Q_0 are shown as crosses. The embedding is done by mapping the point x to one of the elements of these two quantizers.

For example, if $r = 1$, x is mapped to the closest $\circ$ point. The result of the embedding is the distance vector to the nearest $\times$ or $\circ$ as chosen by r. When during reproduction procedure x is perturbed by noise, the quantizer will assign the received data to the closest $\times$ or $\circ$ point, and output 0 or 1 respectively. The set of the two quantizers $\{Q_0, Q_1\}$ is called a `QIM`.

The amount of tolerated noise or the reliability is determined by the minimum distance between two neighboring reconstruction points. The size and shape (for high dimensional quantization) of the Voronoi region determines the tolerance for error. The number of quantizers in the `QIM` set determines the amount of information that can be embedded. By setting the number of quantizers and by choosing the shape and size of the decision region, the performance properties can be finely tuned.

Formally, a *Quantization Index Modulation* data hiding scheme can be seen as $\texttt{QIM} : U \times R \to M$ a set of individual quantizers $\{Q_1, Q_2, \ldots Q_{2^l}\}$, where $l = |R|$ and each quantizer maps $x \in U$ into a reconstruction point. The quantizer is chosen by the input value $r \in R$ such that $\texttt{QIM}(x, r) = Q_r(x)$. The set of all reconstruction points is $M = \bigcup_{r \in R} M_r$ where $M_r \subset M$ is the set of reconstruction points of the quantizer Q_r.

We define the *minimum distance* $\sigma_{\min}$ of a `QIM`, as the minimum distance between reconstructions points of all quantizers in the `QIM`:

$$\sigma_{\min} = \min_{r_1, r_2 \in R} \quad \min_{c^i_{r_1} \in M_{r_1}, c^j_{r_2} \in M_{r_2}} d(c^i_{r_1}, c^j_{r_2})$$

where $M_{r_1} = \{c^1_{r_1}, c^2_{r_1}, \ldots\}$ and $M_{r_2} = \{c^1_{r_2}, c^2_{r_2}, \ldots\}$. Hence, balls with radius $\frac{\sigma_{\min}}{2}$ and centers in M are disjoint.

Let ζ_r be the smallest radius ball such that balls centered in the reconstruction point of quantizer Q_r with radius ζ_r cover the universe U. We define the *covering distance* $\lambda_{\max}$ as:

$$\lambda_{\max} = \max_{r \in R} \zeta_r.$$

Any ball $B(c, \zeta_r)$ contains at least one ball $B(c_r, \sigma_{\min}/2)$ for $c_r \in M_r, \forall r \in R$. Hence, balls with radius $\lambda_{\max}$ and centers in M_r cover the universe U.

A *dithered* `QIM` [9] is a special type of `QIM` for which all Voronoi regions of all individual quantizers are congruent polytopes (generalization of a polygon to higher dimensions). Each quantizer in the ensemble $\{Q_1, Q_2, \ldots Q_{2^l}\}$ can be obtained by shifting the reconstruction points of any other quantizer in the ensemble. The shifts correspond to dither vectors $\{v_1, v_2, \ldots v_{2^l}\}$. The number of dither vectors is equal to the number of quantizers in the ensemble.

Now that we have presented the necessary preliminaries, we are ready to present the notion of a fuzzy embedder in the next section.

4 Fuzzy embedder

In this section, we propose a general approach to embed cryptographic keys into noisy, continuous data. In addition, we show the relation between our new fuzzy embedder primitive and two related concepts, the fuzzy extractor and fuzzy commitment. It is worth stressing that the random key r is not extracted from the random x but is generated independently, see Fig. 4.

Definition 3 (*Fuzzy embedder*) A $(U, \ell, \rho, \varepsilon, \delta)$-fuzzy embedder scheme consists of two polynomial-time algorithms $\langle \mathsf{Embed}, \mathsf{Reproduce} \rangle$, which are defined as follows:

- Embed: $U \times R \rightarrow P$, where $R = \{0, 1\}^l$. This algorithm takes $x \in U$ and $r \in R$ as input and returns a public string $p \in P$.
- $\mathsf{Reproduce}$: $U \times P \rightarrow R$. This algorithm takes $x' \in U$ and $p \in P$ as input, and returns a string from R or an error $\perp$.

Given any random variable X over U and a random variable R of size ℓ, the parameters $\rho, \varepsilon, \delta$ are defined as follows:

- The parameter ρ represents the probability that the fuzzy embedder can successfully reproduce the embedded key, and it is defined as

$$\rho = \min_{r \in R} \max_{x \in U} \Pr(\mathsf{Reproduce}(x', \mathsf{Embed}(x, r)) \; r | x' \in X).$$

  In the above definition, the maximum over $x \in U$ ensures that we choose the best possible representative x for the random variable X. In most cases, this will be the mean of X.
- The security parameter ε is equal to the mutual information between the embedded key, and the public sketch P, and it is defined as $\varepsilon = I(R; \mathsf{Embed}(X, R))$.

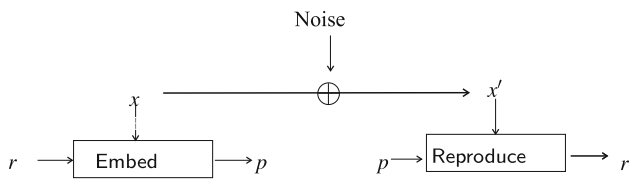

Fig. 4 A fuzzy embedder is a pair of two procedures Embed and $\mathsf{Reproduce}$. The Embed procedure, which takes as input a noisy input x and a binary sequence r generated independently, is executed first. The resulting sketch p is made public. The $\mathsf{Reproduce}$ procedure, which takes as input x' which is (possibly) corrupted by noise and the public sketch p, will output r if x and x' are close

- The security parameter δ is equal to the mutual information of the noisy data and the public sketch and is defined as $\delta = I(X; \mathsf{Embed}(X, R))$.

A few notes are needed to motivate our choice of the security measures of a fuzzy embedder construction. Since the public sketch is computed both on X and R, ε measures the amount of information revealed about X (biometric or PUF), and δ measures the amount of information P reveals about the cryptographic key R.

When evaluating security of algorithms, which derive secret information from noisy data, entropy measures like min-entropy and average min-entropy or entropy loss are appealing since these measures have clear security applicability. However, these measures can only be applied to a variable that has a discrete probability density function. In the case of a continuous random variable, these entropy measures depend on the precision used to represent the values of a random variable, as shown in the next example for min-entropy.

Example Assume that all points X are real numbers between $[0, 1]$ and are uniformly distributed. Assume further that points in X are represented with 2-digit precision, which leads to a min-entropy $H_\infty(X) = \log_2 100$. If we choose to represent points with 4-digit precision, the min-entropy of X becomes $H_\infty(X) = \log_2 10000$, which is higher than $H_\infty(X) = \log_2 100$, although in both cases X is uniformly distributed on the interval $[0, 1]$.

More examples related to average min-entropy and entropy loss can be found in Li et al. [18]. We chose mutual information measure, i.e $I(X; P)$ and $I(R; P)$ because it captures the measure of dependence between two random variables regardless of their type of distribution discrete or continuous. A similar measure for the dependence of two variables is the statistical distance between their distribution. In this case, our choice is motivated by the generality given by the information theoretical measures.

FUZZY EXTRACTOR AND FUZZY EMBEDDER From *Definitions* 1 and 3, we argue that a fuzzy embedder is more general than a fuzzy extractor, due to the following reasons:

1. The fuzzy embedder scheme accepts continuous data as input and can embed different keys, while in a practical deployment, a fuzzy extractor scheme must be combined with quantization and re-randomization to achieve the same goals as a fuzzy embedder.
2. Given a $(U, \ell, \rho, \varepsilon, \delta)$-fuzzy embedder, we can construct a fuzzy extractor as follows:
   - $\mathsf{Generate}'$: $U \rightarrow P \times R$. This algorithm takes $x \in U$ as input, chooses $r \in R$, and returns $p = \mathsf{Embed}(x, r)$ and r.

– Reproduce′: $U \times P \rightarrow R$. This algorithm takes $x' \in U$ and $p \in P$ as input and returns the value Reproduce(x', p).

FUZZY COMMITMENT AND FUZZY EMBEDDER A fuzzy embedder construction leads to a fuzzy commitment [16] construction, since a fuzzy commitment scheme can be constructed from a fuzzy embedder by adding a checksum to the output of the embed procedure. Specifically, given a $(U, \ell, \rho, \epsilon, \delta)$-fuzzy embedder scheme, we can construct a fuzzy commitment scheme as follows:

– Commit′: $U \times R \rightarrow P$. This algorithm takes $x \in U$ and $r \in R$ as input and returns $p =$ Embed(x, r) and $c = \mathsf{h}(r)$ where h is an appropriate hash function.
– Decommit′: $U \times P \rightarrow R$. This algorithm takes $x' \in U, p \in P$, and c as input, and computes $r' =$ Reproduce(x', p). If $r' = \perp$ or $\mathsf{h}(r') \neq c$, output $\perp$; otherwise output r'.

Given a random variable X over U, an honest committer can successfully convince the verifier with probability ρ, to accept the committed string. The parameter ε is an indicator of the hiding property, and the binding property is achieved if $\mathsf{h}(\cdot)$ is collision resistant.

In the next section, we take a closer look at the necessary building blocks when realizing a practical fuzzy embedder and explain the intricacies of a realizing a practical fuzzy embedder.

4.1 Fuzzy embedder building blocks

A fuzzy extractor can transform a noisy, non-uniform discrete source of data, which is easily accessible into a reproducible, uniformly random string, which is suitable to be used as a cryptographic key. Basically, the fuzzy extractor performs two functions: the first is error correction, which compensates for the noise in the source data, and the second is smoothing the non-uniform distribution of the source into a uniformly random distribution of the output.

When considering a fuzzy extractor construction in a practical scenario, the two functions provided are not enough. Firstly, a fuzzy extractor is too limited because it accepts only discrete input data. Thus, a procedure that transforms continuous data into discrete data is necessary. The *cs*-fuzzy extractor, proposed by Buhan et al. [6], is an extension of the fuzzy extractor construction in this sense. Secondly, a fuzzy extractor as pointed out by Boyen [3] needs to re-randomize its output such that one noisy source can be used in more than one application.

A typical fuzzy extractor implementation can be modeled as in Fig. 5. In our view, there are four main building blocks: *quantization, error correction, randomness extraction*, and *randomization*, which can be used in a typical fuzzy extractor implementation.

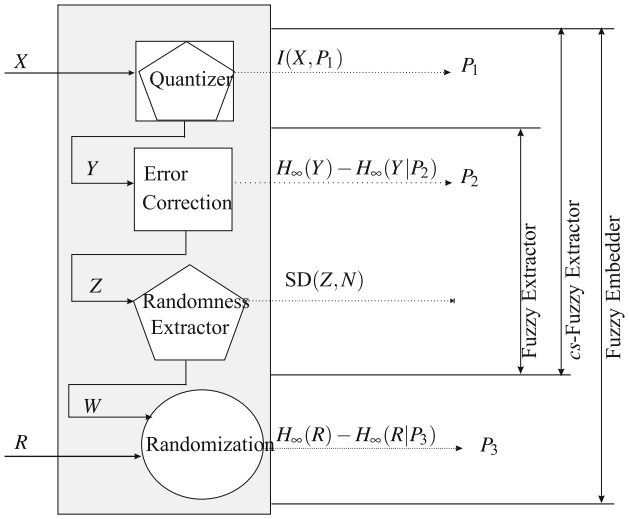

Fig. 5 Typical implementation of a fuzzy extractor. The shape of a *block* is a code for its purpose. *Square blocks* perform error correction, *pentagonal blocks* shape the distribution of the data, while the *circle blocks* are used to randomize the data. A fuzzy extractor can be constructed from an error correcting block and a randomness extractor. On the *left-hand side* of the figure the input variable (with capital letters, above the *arrow*). On the *right-hand side* of each block, the security measure used to evaluate the performance of the block is presented

Each block in Fig. 5 solves a specific problem, and in the following we take a closer look at the purpose and requirements for each of the four blocks.

QUANTIZATION The quantization block is used to transform continuously distributed data X with probability density function $f_X(x)$ into discretely distributed data Y with discrete probability density $f_Y(y)$. This block can shape the probability density function distribution $f_X(x)$ into $f_Y(Y)$ and changes the continually distributed data into discretely distributed data. Examples of quantization schemes can be found in Chen et al. [11] and Zhang et al. [31] and in Fig. 2. During quantization, the public sketch denoted with P_1 in Fig. 5 is computed and made public. The information leaked by the public sketch about the noisy source data is measured in terms of mutual information $I(X; P_1)$ between the source data X and the public sketch P_1.

ERROR CORRECTION The error correction block adds redundant information to the input variable Y to increase the probability that its values are correctly reproduced. The input variable $Y = (Y_1, Y_2, \ldots Y_n)$ is represented as a n-dimensional vector, and its elements Y_i are called feature vectors.

There are two types of noise that can occur in Y. The first is *additive noise* where elements of Y_i are perturbed by noise, and the second is *replacement noise* where some features of Y

can disappear, and new features can appear between two consecutive measurements. Error correction schemes that correct additive noise were proposed by several authors [11,19,31] while error correction schemes for replacement noise can be found in [7,28].

To perform error correction, a *public sketch* (also called *helper data*) is computed for Y. If the helper data is made public, which is the case in most scenarios, it reveals information about the variable Y.

The performance of an error correction scheme is measured in terms of how many *errors* it can correct and the amount of *leaked information*.

When the source data is continuous, the leakage is measured in terms of mutual information, in Fig. 5 $I(X; P)$, where P can be either P_1 or P_2. When the source data is discrete as in the error correction block in Fig. 5, the amount of leaked information is measured in terms of min-average entropy $H_\infty(Y; P_2)$ and min-entropy $H_\infty(Y)$. The difference between the two is called the *entropy loss*.

RANDOMNESS EXTRACTOR This block is used to transform *any* probability density function $f_Y(y)$ into a *uniform* probability function $f_Z(z)$, which is desirable for a cryptographic algorithm. A randomness extractor is used to "purify" the randomness coming from an imperfect source of randomness, it can efficiently convert a distribution that contains some entropy (but is also biased and far from uniform) Y into an almost uniform random variable Z.

The performance of a randomness extractor is measured in terms of the statistical distance between the distribution of the output variable Z and the distribution of a uniform random variable N, denoted in Fig. 5 by $\mathrm{SD}(Z, N)$.

In the process of randomness extraction, an external source of randomness must be present. Reducing the amount of required randomness in the external source and producing outputs, which are as close as possible to a uniform distribution is the main research topic in this area [1,24,25].

There are constructions known as *strong randomness extractors* [12] for which the output of the randomness extractor looks uniform even when the external source of randomness is made public, which are more convenient for the purpose of the scenario depicted in Fig. 5.

RANDOMIZATION This block is used to randomize the string which can be extracted from the noisy source. When biometrics is used as a noisy source, the purpose of randomization is protection of privacy for the biometric data. For example, from one fingerprint only one reproducible, uniform string can be extracted. The randomization ensures that from one fingerprint multiple random sequences, which can be used as cryptographic keys for more applications, can be produced.

We argue that the model described in Fig. 5 covers most of the work done in the area of construction of cryptographic keys from noisy data. Theoretical work in the area usually covers the error correction block and randomness extraction [12,13], whereas others look at more practical aspects like quantization [6,8,11,18,31] or randomization [3,4,7].

The fuzzy embedder construction is intended as an all-encompassing theoretical model given the functionality of a fuzzy extractor. Thus, a fuzzy embedder is able to hide a key in any type of source data. Most of the work in the area of cryptographic use of noisy data focuses on optimizing one aspect, e.g quantization and randomness extraction. Security measures used to quantify the performance in each block are different as they are studied in different research areas. In a practical scenario, when all these blocks are needed it is important to have an overall view of all the information that is leaked or the amount of errors that are corrected. The main purpose of the fuzzy embedder is to put things in perspective and define the overall security measures.

When realizing a fuzzy embedder in a practical scenario, the chosen building blocks depend mostly on the input data and the desired properties of the output data. However, using any given block adds a penalty, expressed in terms of security performance thus adding a minimal number of building blocks is desirable. In the next section, we will present a general construction for a fuzzy embedder that does not rely directly on a fuzzy extractor and uses only a quantizer and a randomizer.

5 Practical construction of a fuzzy embedder

In this section, the following three practical issues are presented. Firstly, we construct a fuzzy embedder using a `QIM`. Secondly, we analyze the performance of this construction in terms of reliability and security. Thirdly, we investigate optimization issues when U is n-dimensional.

`QIM`-*FUZZY EMBEDDER* A fuzzy embedder can be constructed from *any* `QIM` by defining the embed procedure as:

$$\mathsf{Embed}(x, r) = \texttt{QIM}(x, r) - x,$$

and the reproduction procedure as the minimum distance Euclidean decoder:

$$\mathsf{Reproduce}(x', p) = \widetilde{Q}(x' + p),$$

where $\widetilde{Q} : U \to R$ is defined as

$$\widetilde{Q}(y) = \mathsf{argmin}_{r \in R} d(y, M_r).$$

Example Our construction is a generalization of the scheme of Linnartz et al. [19]. Figures 6 and 7 illustrate the Embed, respectively the Reproduce procedures for a `QIM` ensemble of three quantizers $\{Q_\circ, Q_+, Q_\star\}$. During embedding, the secret $r \in \{\circ, \star, +\}$ selects a quantizer, say $Q_\circ$. The selected quantizer finds the reconstruction point $Q_\circ(x)$ closest to x,

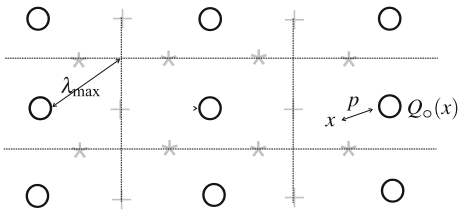

Fig. 6 Embed procedure of QIM fuzzy embedder

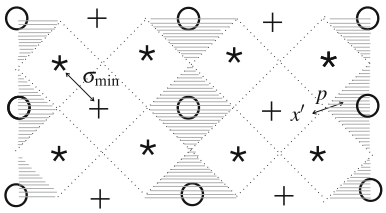

Fig. 7 Reproduce procedure of a QIM fuzzy embedder

and the embedder returns the difference between the two as p, with $p \leq \lambda_{\max}$. Reproduction of p and x' should return $\circ$ if x' is close to x; however, this happens only if $x' + p$ is close to $Q_\circ(x)$ or in other words, if $x' + p$ is in one of the Voronoi regions of $Q_\circ$ (hatched area in Fig. 7). Errors occur if $(x' + p)$ is not in any of the Voronoi regions of $Q_\circ$, thus the size and shape (for $n \geq 2$) of the Voronoi region parameterized by the radius of the inscribed ball $\sigma_{\min}/2$ determines the probability of errors.

5.1 Reliability

In the following lemma, we link the reliability of a QIM-fuzzy embedder to the size and shape of the Voronoi regions of the employed QIM.

Lemma 1 (Reliability) *Let* $\langle \mathsf{Embed}, \mathsf{Reproduce} \rangle$ *be a* $(U, \ell, \rho, \epsilon, \delta)$ QIM*-fuzzy embedder, and let* X *be a random variable over* U *with joint density function* $f_X(x)$*. For any* $r \in R$*, we define*

$$\rho(r) = \int_{V_r} f_X(y - \mathsf{Embed}(X, r))dy,$$

where $V_r = \bigcup_{c \in M_r} V_c$ *is the union of the Voronoi regions of all reconstruction points in* M_r*. Then the reliability is equal to*

$$\rho = \min_{r \in R} \rho(r).$$

Proof Since $\rho(r)$ is exactly the probability that an embedded key r will be reconstructed correctly, the statement follows from the definition. □

In most practical applications, noise has two main properties: larger distances between x and the measurement x' are increasingly unlikely, and the noise is not directional. Thus, the primary consideration for reliability is the size of the inscribed ball of the Voronoi regions, which has radius $\sigma_{\min}/2$.

Corollary 1 (Bounding ρ) *In the settings of Lemma 1, the reliability* ρ *can be bounded by*

$$\min_{r \in R} \sum_{c \in M_r} \int_{B(c, \frac{\sigma_{\min}}{2})} f_X(y)dy \leq \rho$$

where $B(c, r)$ *is the ball centered in* c *with radius* r*.*

Proof The above relation follows from the definition of reliability, since $B(c, \frac{\sigma}{2}) \subset V_c$ and $y = x + \mathsf{Embed}(X, r)$ is always a reconstruction point. □

Corollary 1 shows that reliability is at least the sum of all probabilities computed over balls of radius $\frac{\sigma_{\min}}{2}$ inscribed in the Voronoi regions. Thus the size of the inscribed ball is an important parameter, which determines the reliability to noise.

Example In two-dimensional space there are three regular polytopes, which tile the space: triangle, square and hexagon. If the size of the inscribed circle is equal for all three, in case of a spherically symmetric distribution like the normal distribution, the hexagon has superior reliability performance compared to the other two polytopes because its shape is more close to a ball. The shape of the decision region that inscribes the ball is important as well as we show in Sect. 6.

5.2 Security

In this section, we link the security of a fuzzy embedder to the covering radius, $\lambda_{\max}$ of the employed QIM.

We start this paragraph with one observation. If an adversary learns the value x, she can reproduce the value r with the help of the public value p. However, if an adversary learns the secret key r, she could potentially circumvent the security altogether but cannot reproduce x. We illustrate this observation in the next example.

Example In the fuzzy embedder example given in Fig. 7, the adversary can choose between three different key values$\{\circ, +, \star\}$. Assume she learns the correct key, in our example $\circ$. To find the correct value for x, she still has to decide which of the reconstruction points of the quantizer $Q_\circ$ is closest to x. Without any other information, this is an impossible task since the quantizer $Q_\circ$ has an infinite number of reconstruction points.

The public sketch p leaks information about both the random string r (the amount of information revealed is δ) and the value x (the amount of information revealed is denoted

with ε). We note that full disclosure of the string r is not enough to recover x.

We now consider how large δ, the leakage on the key can be in terms of P, which due to our construction is a continuous variable. We know that any $p \in P$ has the property that $p \leq \lambda_{\max}$. A technical difficulty in characterizing the size of P arises as P is not necessarily discrete. Tuyls et al. [27] show the following result, establishing a link between the continuous and the quantized version of P denoted here with P_d.

Lemma 2 (Tuyls et al. [27]) *For continuous random variables X, Y, and $\xi > 0$, there exists a sequence of quantized random variables X_d, Y_d that converge pointwise to X, Y (when $d \to \infty$) such that for sufficiently large d, $I(X; Y) \geq I(X_d; Y_d) \geq I(X; Y) - \xi$.*

From the lemma above we have:

$$I(R; P_d) \leq H(P_d) \leq |P_d|,$$

P_d is a quantized representation, of the public sketch P, using a uniform scalar quantizer with step d. The reason for quantizing P is to make it suitable for a digital representation. $|P_d|$ represents the size, in bits, of the sketch.

To limit the information loss of the construction, which is the result of publishing the sketch P_d, it is best to have $|P_d|$ as small as possible. However, a small representation of P_d implies that the cardinality of the set of values of P_d is small as well. There are two ways in which we can achieve a small representation for P_d. The first is to limit the support on which P is defined, while the second is to choose a higher value for the quantization step d. The second approach is not convenient since the quantization that is used for P has to be used for the noisy data X, thus we concentrate on the first option: limit the support on which P is defined.

In our construction, we have $|P_d| \leq \lambda_{\max}$. Thus by bounding the size of p, we bound the value of δ. In the rest of this paper, for simplicity reasons we use P when referring the P_d.

5.3 Optimization

In this paragraph, we analyze the key length allowed by the restrictions placed by our performance criteria on the embed and reproduce procedures. Firstly, we take a look at the reproduce procedure which ties in directly with the reliability. The minimum size of an error to produce a wrong decoding is $\sigma_{\min}/2$. Thus, the collection of balls centered in the reconstruction point of all quantizers with radius $\sigma_{\min}/2$ should be disjoint.

Secondly, the result of the embed procedure for any arbitrary point x and any key $r \in R$ has to be smaller than the covering distance $\lambda_{\max}$. Hence, for each key r the collection of balls centered in the reconstruction points of Q_k and with radius $\lambda_{\max}$ should cover the entire space U.

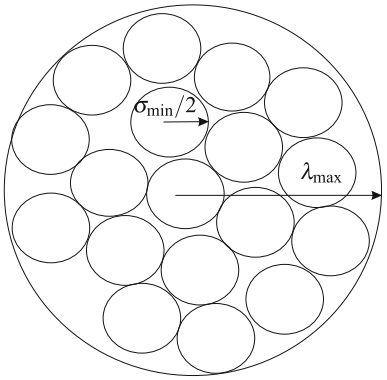

Fig. 8 Optimization of reliability versus security. Reliability is determined by the size of the ball with radius $\sigma_{\min}/2$. Each *small ball* has associated with its center a different key $r \in R$. The number of *small balls* inside the *large ball* with radius $\lambda_{\max}$ is equal to l, the number of elements in R. To have as many keys as possible, we want to increase the number of *small balls*, thus we want *dense (sphere) packing*. The size of the public sketch $p \in P$ is at most $\lambda_{\max}$. Since for any $x \in U$, we want to be within λ_{max} distance to a specific $r \in R$, large balls should *cover optimally* the space U. When the point x falls in a region, which does not belong to any ball, the Reproduce procedure gives the closest center of a *small ball*, thus *we want polytopes which tile the space*

These two radii can be linked as follows:

Lemma 3 *The covering distance of a* QIM, $\lambda_{\max}$ *is bounded from below by:*

$$\lambda_{\max} \geq \sqrt[n]{N}\frac{\sigma_{\min}}{2}$$

where n represents the dimension of the universe U and N is the number of different quantizers.

Proof As noted above, all balls with radius $\sigma_{\min}/2$ centered in the centroids of the whole ensemble are disjoint. Each collection of balls with radius $\lambda_{\max}$ centered in the centroids of an individual quantizer gives a covering of the space U, see Fig. 8. □

Therefore, a ball with radius $\lambda_{\max}$, regardless of its center, contains at least the volume of N disjoint balls of radius $\sigma_{\min}/2$, one for each quantizer in the ensemble. Comparing the volumes, we have

$$s_n \lambda_{\max}^n \geq s_n N \left(\frac{\sigma_{\min}}{2}\right)^n$$

where s_n is a constant only depending on the dimension.

The main conclusion of Lemma 3 is that for a QIM-fuzzy embedder to produce a long random string r, thus the length of r depends on the number of small balls which can be placed into a large ball.

Consider the case when an adversary has partial knowledge about the random variable X. For example, she could

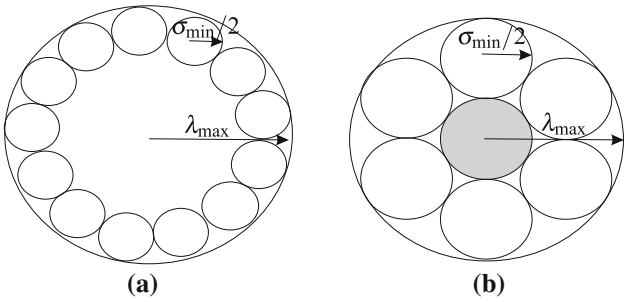

Fig. 9 **a** Construction that yields equiprobable keys in case the background distribution is spherical symmetrical in the two-dimensional space. **b** Optimal construction that results in minimal public sketch size and has equiprobable keys in the two-dimensional space

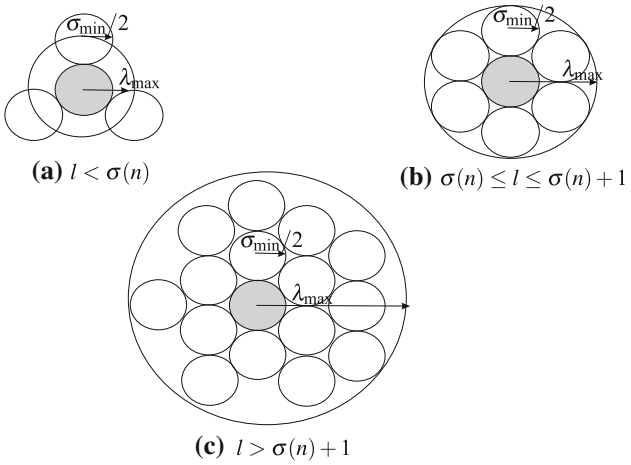

Fig. 10 Different choices for the number of quantizers in relation to $\lambda_{\max}$ in a QIM-fuzzy embedder construction. **a** There is only one quantizer in the QIM set. This construction cannot be used for data hiding. **b** Number of quantizers in the QIM set is equal to $\sigma_n + 1$, when the middle ball has a different codeword then the neighboring balls (e.g., the 7-hexagonal construction) or precisely equal to the σ_n, when the middle ball has no codeword associated (e.g., the 6-hexagonal construction). **c** The QIM set has more quantizers then the kissing number

know the average distribution of all (fingerprint) biometrics, or the average distribution of the PUFs. This average distribution is known in the literature as the *background distribution*. While any QIM-fuzzy embedder achieves equiprobable keys if the background distribution on U is uniform, the equiprobability can break down when this background distribution is non-uniform and known to the adversary. A legitimate question is: *how can a* QIM*-fuzzy embedder achieve equiprobable keys when the background distribution is not uniform?*

In the literature [8,11,19], it is often assumed that the background distribution is a multivariate Gaussian distribution. We make a weaker assumption, namely that the background distribution is not uniform but spherically symmetrical and decreasing. In other words, we assume that measurement errors only depend on the distance, and not on the direction, and that larger errors are less likely.

Thus, to achieve equiprobable keys given this background distribution, the reconstruction points must be equidistant as for example the construction in Fig. 9 a. Note that putting more "small" balls inside the "large" ball is not possible since they are not equiprobable. The problem with the construction in Fig. 9a is the size of the sketch which becomes large.

The natural question, which arises is: *what is the minimum sketch size attainable such that all keys are equiprobable for a given desired reliability?*

This question leads us to consider the kissing number $\tau(n)$, which is defined to be the maximum number of white n-dimensional spheres touching a black sphere of equal radius, see Fig. 9b. The radius of the "small" balls determines reliability and the minimum $\lambda_{\max}$, such that a QIM-fuzzy embedder can be built is equal to the radius of the circumscribed ball as shown in Fig. 9b.

The next question we ask is: *for a minimum sketch size and a given reliability, are there dimensions which are better than others?* For example, why not pack spheres in the three-dimensional space where the kissing number is 12. For the same reliability: is it possible to obtain more keys? For most dimensions, only bounds on the kissing number are known [17,30]. Assuming a spherically symmetrical and decreasing background distribution, there are only so many different equiprobable keys one can achieve:

Theorem 1 (Optimal high dimensional packing) *Assume the background distribution to be spherically symmetrical and decreasing. For a* $(U, \ell, \rho, \varepsilon, \delta)$ QIM*-fuzzy embedder with* $\dim(U) = n$ *with equiprobable keys and minimal sketch size, we have that* $\ell \leq \tau(n)$.

Proof The target reliability ρ will translate to a certain radius σ. In other words, we need to stack balls of radius σ optimally.

In Fig. 10, we have three possible constructions for the QIM-fuzzy embedder, with different choices of number of quantizers in the set versus the size of the public sketch.

The construction in Fig. 10a cannot be used for data hiding since there is only one quantizer in the set. To achieve the maximum number of equiprobable keys without the sketch size getting too big, the best construction is to center the background distribution in one such ball and to assign a different key to each touching ball as in Fig. 10b. Construction in Fig. 10c yields a higher value for $\lambda_{\max}$ and is not optimal from the perspective of the size of the sketch.

The trade-off between the number of quantizers (and thus the length of the output sequence) and the size of the sketch can be seen by comparing constructions in Fig. 10b, c. As the number of quantizers increases so does the size of the sketch.

Thus, the number of possible equiprobable keys, when the background distribution is spherically symmetric and decreasing, is upper bounded by the kissing number $\tau(n)$.

Combined with known bounds on the kissing number [17, 30], we arrive at the following, somewhat surprising conclusion:

Corollary 2 *Assuming a spherically symmetrical and decreasing background distribution on U and equiprobable keys, for a $(U, \ell, \rho, \varepsilon, \delta)$* QIM*-fuzzy embedder, the most equiprobable keys are attained by quantizing two dimensions at a time, leading to*

$$N(n) = 6^{\lfloor \frac{n}{2} \rfloor} 2^{(n-2\lfloor \frac{n}{2} \rfloor)}$$

different keys.

Proof Known upper bounds [17] on the kissing number in n dimensions state that $\tau(n) \leq 2^{0.401n(1+o(1))}$. This means that $N(n) \geq \tau(n)$ in all dimensions, since $N(n) \approx 2^{1.3n}$ and small dimensions can easily be verified by hand. Also note that $N(n_1 + n_2) \leq N(n_1)N(n_2)$. Thus, quantizing dimensions pairwise gives the largest number of equiprobable keys for any spherically symmetric distribution.

Example Given a vector $X = (X_1, X_2, \ldots X_{10})$, there are several choices when considering quantization. One possibility is to quantize each of the elements $X_i, i \in \{1, 10\}$ independently. A second choice is to quantize pairs of elements (X_i, X_j) where $i \neq j$ and $i, j \in \{1, 10\}$. Another option is to quantize three elements at a time (X_i, X_j, X_s) where $i \neq j \neq s$ and $i, j, s \in \{1, 10\}$. We illustrate in this example that the two-dimensional quantization is optimal in the sense of Corollary 2. Table 1 shows the effect of quantization on the key space for different dimension choices.

- For *two-dimensional* quantization (*Table* 1), the kissing number is equal to 6, the 10 elements of vector X are grouped in 5 subsets of 2 elements each. For each subset, we can embed at most 6 keys, and for the 5 pairs we have in total a key space of 6^5 possible keys.
- For *three-dimensional* quantization (*Table* 1), kissing number is 12, the 10 elements of X can be grouped as 3 pairs of 3 elements, and there is one vector element left which can only be quantized in one dimension. The number of possible keys is $12^3 \times 2$.

The result of Corollary 2, confirmed by our example, shows that the best strategy for quantization is the two-dimensional quantization. As this result points us to two dimensions, we will give two practical constructions for the two-dimensional case in the next section.

Table 1 Different choices for quantization and its effect of the key space (maximum number of bits that can be embedded) for a 10-dimensional vector X

Dimension	σ_n	Subsets	Key space
1	2	1×10	$2^{10} = 1024$
2	6	2×5	$6^5 = 7776$
3	12	$3 \times 3 + 1$	$12^3 \times 2 = 3456$
4	24	$4 \times 2 + 2$	$24^2 \times 6 = 3456$
5	40	5×2	$40^2 = 1600$
6	72	$6 + 4$	$72 \times 24 = 1728$
7	126	$7 + 3$	$126 \times 12 = 1512$
8	240	$8 + 2$	$240 \times 6 = 1440$
9	272	$9 + 1$	
10	>336	10	

In the first column, we have the number of dimensions that are quantized at a time, the second column gives the value of the kissing number for the chosen dimension. The third column gives the particular choice for grouping the subsets, and the forth column shows the size of the key space

6 Practical constructions in two dimensions

In this section, we present two optimal constructions for the QIM-fuzzy embedder in the two-dimensional space. The first, 7-hexagonal tiling, is optimal from reliability point of view, while the second is optimal from the number of equiprobable keys it can embed and the sketch size. We choose a hexagonal lattice to represent reconstruction points for the QIM, since this gives both the smallest circle covering (for the Embed procedure) and the densest circle packing (for the Reproduce procedure).

The first construction, the *7-hexagonal tiling*, can embed $n \times \frac{\log_2 7}{2}$ bits, where n is the dimensionality of random variable X. This construction is optimal from the reliability point of view. However, in this construction keys are not equiprobable, when the background distribution is not flat enough. The second construction, the *6-hexagonal tiling*, fixes this problem but achieves a slightly lower key length of $n \times \frac{\log_2 6}{2}$ bits.

In our constructions, the reconstruction points of all quantizers are shifted versions of some base quantizer Q_0. A dither vector $\overrightarrow{v_k}$ is defined for each possible $r \in R$. We define the *tiling polytope* as the repeated structure in the space that is obtained by decoding to the closest reconstruction point. It follows from this definition that the tiling polytope contains exactly one Voronoi region for each quantizer in the ensemble. In Figs. 11 and 12, the tiling polytopes are delimited by the dotted line.

The n-dimensional variable $X = (X_1, X_2, \ldots X_n)$ is partitioned into $\frac{n}{2}$-two-dimensional subspaces (X_1, X_2). Each subspace is considered separately. On the x-axis in Fig. 11 we have the values for X_1, and on the y-axis we have the

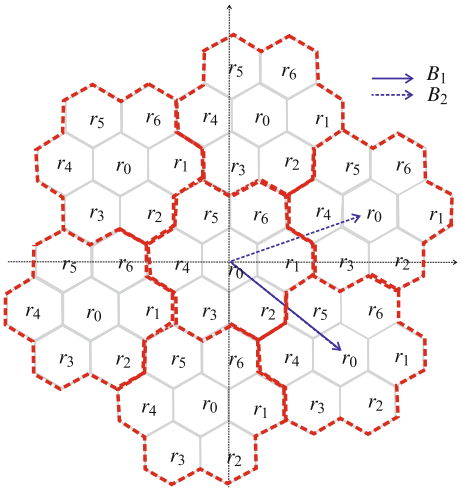

Fig. 11 Reproduce procedure of the 7-hexagonal tiling

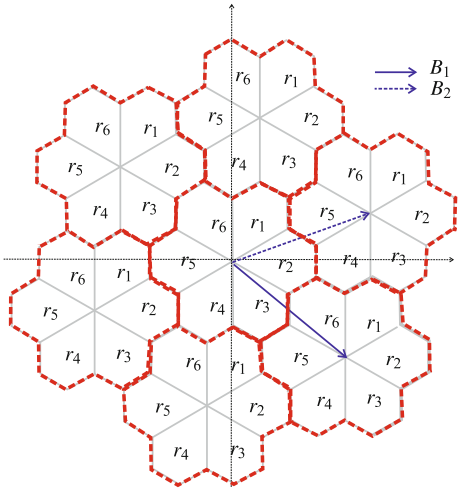

Fig. 12 Reproduce procedure of the 6-hexagonal tiling

values of X_2. Along the z-axis (not shown in the figure), we have the joint probability density $f_{X_1X_2}(x)$.

We start our construction by choosing the densest circle packing existing in the two-dimensional space which is the hexagonal packing. All circles have equal radius, and the center of the circle is the reconstruction point. With each reconstruction point, a key value is associated. However, the circles do not tile the space. As a result when x, the realization of X, falls into the non-covered region, it cannot be associated with any reconstruction point. We need to approximate the circle with some polygons that tile the two-dimensional space. In the two-dimensional space, the Voronoi region for the hexagonal lattice is a hexagon.

In the two-dimensional space, there are only three such regular polygons: triangles, squares, and hexagons. Since we assume a spherical symmetrical distribution for $f_{X_1X_2}$, the hexagon is the best approximation to the circle from reliability point of view. The next step is to associate a key value to each hexagon such that for any value of (X_1, X_2), any key label is at most at the given distance (sphere-covering problem).

6.1 7-Hexagon tiling

Thus, our first construction is a dithered `QIM` defined as an ensemble of 7 quantizers. The reconstruction points of the base quantizer Q_0 are defined by the lattice spanned by the vectors $\overrightarrow{B_1} = (5, \sqrt{3})q, \overrightarrow{B_2} = (4, -2\sqrt{3})q$, where q is the scaling factor of the lattice. In Fig. 11, these points are labeled k_0. The other reconstruction points of quantizers $Q_i, i = 1, \ldots, 6$ are obtained by shifting the base quantizer by the dither vectors $\{\overrightarrow{v_1}, \ldots, \overrightarrow{v_6}\}$ such that $Q_i(x) = Q_0(\overrightarrow{x} + \overrightarrow{v_i})$. The values for these dither vectors are: $\overrightarrow{v_0} = (0, 0)$, $\overrightarrow{v_1} = (2, 0)$, $\overrightarrow{v_2} = (-3, \sqrt{3})$, $\overrightarrow{v_3} = (-1, -\sqrt{3})$, $\overrightarrow{v_4} = (-2, 0)$, $\overrightarrow{v_5} = (3, -\sqrt{3})$, and $\overrightarrow{v_6} = (1, \sqrt{3})$. The embed and reproduce procedures work as in our construction in Sect. 5. The reproduce procedure is shown in Fig. 11.

6.2 6-Hexagon tiling

Assume that the background distribution is a spherical symmetrical distribution with mean centered in the origin of the coordinates. In the construction above, the hexagon centered in the origin will typically have a higher associated probability than the off-center hexagons. This effect grows as we increase the scaling factor q of the lattice.

This construction eliminates the middle hexagon, to make all keys equiprobable (see Theorem 1). The key length is $\frac{\log_2 6}{2}$ bits. The tiling polytope is formed by 6 decision regions, and thus there are only 6 dither vectors, see Fig. 12. The same dither vectors, $\{\overrightarrow{v_1}, \ldots, \overrightarrow{v_6}\}$ are used to construct the quantizers, but the basic quantizer Q_0 itself is not used. The embed and reproduce procedures are defined as in Sect. 5.

6.3 Performance comparison

We compare the two constructions proposed above, i.e., the 7-hexagonal tiling (Fig. 11) and the 6-hexagonal tiling (Fig. 12), in terms of reliability, min-entropy of the key and entropy loss to the scalar quantization scheme introduced by Linnartz et al. [19] on each dimension separately (we refer to this as 4-square tiling).

To perform the comparison, we consider identically and independently distributed (i.i.d.) Gaussian sources. We assume the background distribution has mean (0, 0) and standard deviation $\sigma_{X_1X_2}$. Without loss of generality, we assume that for any random $(X_1, X_2) \in U^2$, the probability distribution of $f_{X_1X_2}(x)$ has mean $\mu = (\mu_1, \mu_2)$ and standard deviation σ_x^2. This model comes from biometrics, where the

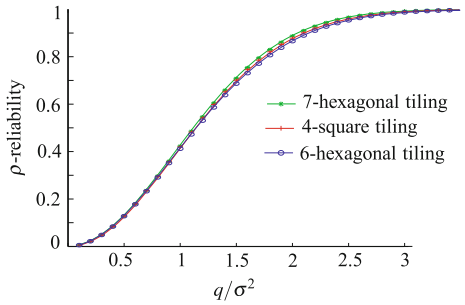

Fig. 13 Key length comparison for the three QIM-fuzzy embedder constructions-scaled to one dimension

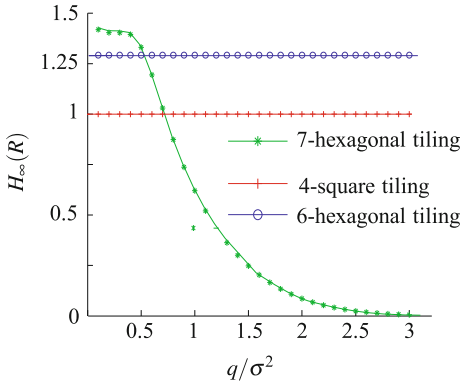

Fig. 14 Mutual information between the key and the public sketch for the three QIM-fuzzy embedders

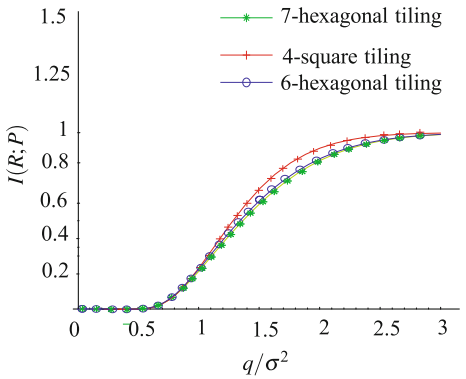

Fig. 15 Reliability of the three QIM-fuzzy embedder constructions

background distribution (also called imposter distribution) describes all users, and the user distribution is the distribution of random variable X.

To evaluate the reliability relative to the quality of the source data (amount of noise, measured in the terms of standard deviation from mean), we compute probabilities associated with equal area decision regions and the reconstruction point centered in the mean μ of the distribution $f_X(x)$. The curves in Fig. 15 are obtained by progressively increasing the area of the Voronoi regions. The size of the Voronoi region is controlled by the scaling factor of the lattice, q. The best performance is obtained by the hexagonal decision regions. This is because the regular hexagon best approximates a circle, the optimal geometrical form for a spherical symmetrical distribution. However, the differences between reliability of the three QIM-fuzzy embedders are small.

The min-entropy in $r \in R$ is compared in Fig. 13 between 7-hexagonal tiling, 6-hexagonal tiling, and 4-square tiling. Maximizing the min-entropy means minimizing the probability for an adversary to guess the key correctly on her first try. The key length for the 7-hexagonal tiling decreases rapidly with the increase in the lattice scaling factor q relative to $\sigma_{X_1X_2}$. While for a small lattice the scaling factor q, one can approximate the background distribution as uniform, with the increase in scaling the center hexagon has a substantially higher probability associated with it, and thus one key value is more likely than all the others.

The 6-hexagonal tiling construction eliminates the middle hexagon, and as a result all keys become equiprobable, at the cost of a somewhat lower reliability, see Fig. 15.

Finally, we evaluate the mutual information for the key when publishing the sketch for the three constructions compared. The results are shown in Fig. 14. The values are scaled to the number of bits lost from each bit that is made public. The results are somewhat surprising in the sense that the 4-square tiling looses more bits compared to our two new constructions. The reason is that while the size of the public sketch p is equal for all three constructions, thus they all lose the same amount of information but the key length differs (Fig. 15).

7 Conclusions

We propose the notion of a *fuzzy embedder* as a generalization of a fuzzy extractor. Fuzzy embedders solve two problems encountered when fuzzy extractors are used in practice: (1) a fuzzy embedder naturally supports renewability and (2) it supports direct analysis of quantization effects. This is made possible by embedding a key instead of extracting one, and by making no limiting assumptions about the nature of the input source.

We give a general construction of a fuzzy embedder, using a QIM to construct the Embed and Reproduce procedures. The QIM performance measures (from watermarking) can be directly linked to the reliability and security properties of the constructed fuzzy embedder.

This construction gives a deep insight into the trade-offs between the parameters of a fuzzy embedder. We describe the key length-entropy loss trade-off as a simultaneous sphere-packing/sphere-covering problem, and we show that when considering equiprobable keys, quantizing dimensions pairwise gives the largest key length.

We also give two explicit, two-dimensional constructions, which can embed a longer key per dimension than existing (one-dimensional) schemes. The 7-hexagonal tiling scheme

achieves the optimal probability of detection but only performs well if the underlying background distribution is flat enough. We show that our 6-hexagonal tiling scheme is optimal from a key length perspective, given that each key is equiprobable. Using the 6-hexagonal construction, we obtain $\frac{log_2 6}{2}$ bits per dimension of the input data, which is superior compared to the single bit obtained by the shielding scheme.

We propose a new, holistic model, the fuzzy embedder, that encompasses both the theoretical clarity and the practical needs of a template protection scheme.

References

1. Barak, B., Impagliazzo, R., Wigderson, A.: Extracting randomness using few independent sources. In: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04), vol. 45, pp. 384–393. Roma, Italy, Oct (2004)
2. Barron, R.J., Chen, B., Wornell, G.W.: The duality between information embedding and source coding with side information and some applications. IEEE Trans. Inf. Theory **49**(5), 1159–1180 (2003)
3. Boyen, X.: Reusable cryptographic fuzzy extractors. In: Atluri, V., Pfitzmann, B., McDaniel, P.D. (eds.) Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS 2004), Washington DC, USA pp. 82–91. ACM, Oct (2004)
4. Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., Smith, A.: Secure remote authentication using biometric data. In: Ronald, C., (ed.) Advances in Cryptology, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2005), Aarhus, Denmark volume 3494 of Lecture Notes in Computer Science, pp. 147–163. Springer, May (2005)
5. Buhan, I.R., Doumen, J., Hartel, P.H.: Controling leakage of biometric information using dithering. In: Proceedings of the 16th European Signal Processing Conference (EUSIPCO), Lausanne, Switzerland EUSIPCO. European Association for Signal, Speech and Image Processing, EURASIP, Aug (2008)
6. Buhan, I.R., Doumen, J., Hartel, P.H., Veldhuis, R.N.J.: Fuzzy extractors for continuous distributions. In: Deng, R., Samarati, P. (eds.) Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS), Singapore, pp. 353–355, New York, March (2007). ACM. (Subsumed by Chapter 3 of this thesis, except examples)
7. Chang, E.C., Li, Q.: Hiding secret points amidst chaff. In: Serge, V., (ed.) 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Saint Petersburg, Russia volume 4004 of Lecture Notes on Computer Science, pp. 59–72. Springer, May (2006)
8. Chang, Y.J., Zhang, W., Chen, T.: Biometrics-based cryptographic key generation. In IEEE International Conference on Multimedia and Expo (ICME'04), Taipei, Taiwan, pp. 2203–2206. IEEE Computer Society, June (2004)
9. Chen, B., Wornell, G.W.: Dither modulation: a new approach to digital watermarking and information embedding. Proc. SPIE Secur. Watermarking Multimed. Contents **3657**, 342–353 (1999)
10. Chen, B., Wornell, G.W.: Quantization index modulation methods for digital watermarking and information embedding of multimedia. J. VLSI Signal Process., Springer, Netherlands **27**(1–2), 7–33 (2001)
11. Chen, C., Veldhuis, R.N.J., Kevenaar, T.A.M., Akkermans, A.H.M.: Multi-bits biometric string generation based on the likelyhood ratio. In: IEEE Conference on Biometrics: Theory, Applications and Systems (BTAS'07), Washinton, DC, pp. 1–6. IEEE Computer Society, Sept (2007)
12. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology, Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2004), Interlaken, Switzerland volume 3027 of Lecture Notes in Computer Science, pp. 523–540. Springer, May (2004)
13. Dodis, Y. and Smith, A.: Correcting errors without leaking partial information. In: Gabow H.N., Fagin, R. (eds.) Proceedings of the 37th Annual ACM Symposium on Theory of Computing(STOC), pp. 654–663. ACM, Baltimore, MD, USA, May (2005)
14. Gersho, A.: Principles of quantization. IEEE Trans. Circuits Syst. **25**(7), 427–436 (1978)
15. Gersho, A.: Asymptotically optimal block quantization. IEEE Trans. Inf. Theory **25**(4), 373–380 (1979)
16. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS). Singapore, pp. 28–36. ACM SIGSAC, Nov (1999)
17. Kabatiansky, G.A., Levenshtein, V.I.: Bounds for packings on a sphere and in space. Probl. Peredachi Informatsii **1**, 3–25 (1978)
18. Li, Q., Sutcu, Y., Memon, N.: Secure sketch for biometric templates. In: Lai, X., Chen, K., (eds.) Advances in Cryptology 12th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2006), Shanghai, China volume 4284 of Lecture Notes in Computer Science, pp. 99–113. Springer, Dec (2006)
19. Linnartz, J.P., Tuyls, P.: New shielding functions to enhance privacy and prevent misuse of biometric templates. In: Kittler, J., Nixon, M.S., (eds.) 4th International Conference on Audio-and Video-Based Biometrie Person Authentication (AVBPA 2003), Guildford, UK volume 2688 of Lecture Notes in Computer Science, pp. 393–402. Springer, June (2003)
20. Maurer, U.: Perfect cryptographic security from partially independent channels. In: Proceedings of the 23rd ACM Symposium on Theory of Computing (STOC), New Orleans, Louisiana, USA, pp. 561–572. ACM Press, Aug (1991)
21. Maurer, U.: Secret key agreement by public discussion. IEEE Trans. Inf. Theory **39**(3), 733–742 (1993)
22. Moulin, P., Koetter, R.: Data-hiding codes. Proc. IEEE **93**(12), 2083–2126 (2005)
23. Skoric, B., Tuyls, P., Ophey, W.: Robust key extraction from physical uncloneable functions. In: Ioannidis, J., Keromytis, A.D., Yung, M., (eds.) Applied Cryptography and Network Security (ACNS 2005), New York, NY, USA volume 3531 of Lecture Notes in Computer Science, pp. 407–422. Springer, June (2005)
24. Ta-Shma, A.: On extracting randomness from weak random sources (extended abstract). Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (STOC 1996), Philadephia, Pennsylvania, USA **28**, 276–285, May 1996
25. Trevisan, L., Vadhan, S.: Extracting randomness from samplable distributions. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science,Redondo Beach, CA, USA, volume 41, pp. 32–42. IEEE Computer Society, (2000)
26. Tuyls, P., Akkermans, A., Kevenaar, T., Schrijen, G., Bazen, A., Veldhuis, R.: Practical biometric authentication with template protection. In: Takeo, K., Anil K.J., Nalini K.R. (eds.) Proceedings of the 5th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA 2005), Hilton Rye Town, NY, USA volume 3546 of Lecture Notes in Computer Science, pp. 436–446. Springer, July (2005)
27. Tuyls, P., Goseling, J.: Capacity and examples of template-protecting biometric authentication systems. In: Maltoni, D., Jain, A.K. (eds.) Proceedings of International Workshop on Biometric

Authentication (ECCV 2004), Prague, Czech Republic volume 3087 of Lecture Notes in Computer Science, pp. 158–170. Springer, May (2004)
28. Uludag, U., Pankanti, S., Jain, A.K.: Fuzzy vault for fingerprints. In: Kanade, T., Jain, A.K., Ratha, N.K. (eds.) Proceedings of the 5th International Conference on Audio- and Video-Based Biometric Person Authentication, (AVBPA 2005) Hilton Rye Town, NY, USA volume 3546 of Lecture Notes in Computer Science pp. 310–319. Springer, July (2005)
29. Uludag, U., Pankanti, S., Prabhakar, S., Jain, A.K.: Biometric cryptosystems: issues and challenges. Proc. IEEE **92**(6), 948–960 (2004)
30. Zeger, K., Gersho, A.: Number of nearest neighbors in a euclidean code. IEEE Trans. Inf. Theory **40**(5), 1647–1649 (1994)
31. Zhang, W., Chang, Y.J., Chen, T.: Optimal thresholding for key generation based on biometrics. In: Proceedings of the IEEE 2004 International Conference on Image Processing (ICIP 2004), Singapore pp. 3451–3454. IEEE Computer Society, Oct (2004)