Facial Template Protection via Lattice-based Fuzzy Extractors

Kaiyi Zhang[1], Hongrui Cui[1], and Yu Yu[1]

[1]Department of Computer Science, Shanghai Jiao Tong University
{*kzoacn,rickfreeman*}*@sjtu.edu.cn*
yuyu@yuyu.hk

November 27, 2021

Abstract

With the growing adoption of facial recognition worldwide as a popular authentication method, there is increasing concern about the invasion of personal privacy due to the lifetime irrevocability of facial features. In principle, *Fuzzy Extractors* enable biometric-based authentication while preserving the privacy of biometric templates. Nevertheless, to our best knowledge, most existing fuzzy extractors handle binary vectors with Hamming distance, and no explicit construction is known for facial recognition applications where ℓ_2-distance of real vectors is considered. In this paper, we utilize the dense packing feature of certain lattices (e.g., E_8 and Leech) to design a family of *lattice-based* fuzzy extractors that docks well with existing neural network-based biometric identification schemes. We instantiate and implement the generic construction and conduct experiments on publicly available datasets. Our result confirms the feasibility of facial template protection via fuzzy extractors.

Keywords— Facial Recognition, Fuzzy Extractor, Privacy Protection, Biometric Authentication

1 Introduction

Biometic-based authentication technology has become increasingly popular in the past years. It is used almost everywhere in public spaces such as airports, hotels, shopping centers, and on personal devices such as mobile phones. Despite the convenience and potential benefits (e.g., to prevent and solve crimes) it offered, there are rising concerns about privacy, security, and legislation regarding the use of the technology.

A major security challenge we tackle in this paper is the tension between the ease of use and secure storage of the biometric templates. Unlike password-based authentication where passwords can be stored in a protected form (salted hash), noisy biometric templates have to be stored in the clear to facilitate biometic-based authentication. Once the data storage is compromised then it may lead to security breaches and privacy issues.

Facial templates are typically protected with standard encryption or in trustworthy environments. As far as mobile devices are concerned, iPhone and iPad Pro store

FaceID templates in Secure Enclave [App21], and Android offers a TEE called Trusty to store sensitive information like fingerprint templates [And21]. When it comes to PC, Windows Hello stores biometric templates locally in AES-encrypted form [Mic21]. However biometric templates are protected (by SE, TEE or encryption), they must be decrypted into memory in clear form for subsequent use.

1.1 Biometric-Based Authentication

In a typical password-based authentication, a server receives a username and password from a client via a secure channel, and the server verifies whether they match the record in her own database before granting access. To mitigate the risk of password database compromise and to avoid sending passwords in cleartext[1], the server can store the digest of a password instead of the password itself. Unfortunately, biometric-based authentication does not have such a straightforward (functionally equivalent yet privacy-preserving) counterpart due to the conflict between the noisy characteristics of biometrics and the diffusion nature of cryptographic hashing.

We recall a biometric-based authentication scenario between a client and a server. Biometrics (e.g., fingerprints, face, and iris characteristics) serve as the credentials for authentication due to the unique feature. The client uses the biometric information and interacts with the server to complete the authentication. Server stores (a certain form of) the biometric templates of the clients for authentication and verification.

Let us first consider the following non-privacy-preserving scenario:

Enrollment. Initially, each client registers himself with his ID and a number of his biometric samples, from which some descriptive features are extracted and stored as a template.

Authentication. During authentication, a client claims an identity and gets his biometric information acquired, processed, and uploaded to the server. The server compares the received information against the stored template of the claimed ID based on their closeness under certain metrics and decides whether the authentication succeeds or not.

The disadvantages are obvious: 1) The server sees the sensitive private information including users' facial features[2], which are not strictly necessary for authentication propose; 2) in case that the server is compromised, it will pose further security risks to the owner of the facial features.

To securely store biometrics in a way similar to password hashing, fuzzy extractors are introduced by Dodis et al. [DRS04]. In short, a fuzzy extractor takes as input a sample from a noisy source and generates a digest along with a helper string. With the helper string, one can recover the same digest based on any other sample (from the same source) as long as the two samples are sufficiently close under a certain distance measure. If the source has enough entropy, the extracted digest is guaranteed to be sufficiently (pseudo)random and thus suffices for authentication purposes.

As shown in Figure 1, the original scheme is enhanced into a privacy-preserving one with an appropriate fuzzy extractor:

Enrollment. The client scans its own biometric sample and computes the biometric embedding, denoted as w. Then the client applies a fuzzy extractor to the embedding to obtain a random string and a "helper string", denoted as $(R, P) \leftarrow \mathsf{FE.Gen}(w)$. The random string is then hashed into a digest $H(R)$ and sent to the server together with

[1]For simplicity, we omit other (e.g., rainbow table and replay) attacks, which can be addressed by means of randomization (e.g., salted hashing and nonce-based protocol).

[2]For example, the GDPR categorizes facial recognition data as sensitive personal data that requires additional protection.

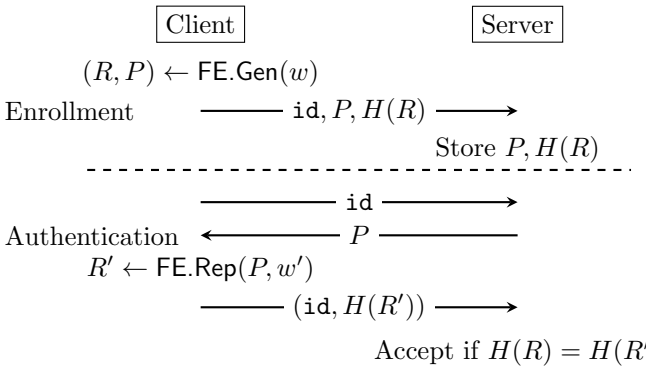


Figure 1: A fuzzy extractor-based authentication process.

the client's identity `id`. Finally, the client stores the helper locally and discards the rest relevant information.

Authentication. Upon a request to authenticate itself, the client re-captures the client's biometric sample, computes its embedding w', and recovers the corresponding digest with the "helper string", i.e., $R' \leftarrow \mathsf{FE.Rep}(P, w')$. Then, it sends $\mathtt{id}, H(R')$ to the server who accepts if $H(R) = H(R')$.

Nevertheless, an important subtlety, mostly neglected in the literature of theoretical fuzzy extractors, is that most of the existing fuzzy extractors can only handle low Hamming weight errors while the errors introduced in the neural network based facial recognition algorithms are of low cosine (or equivalently ℓ_2) distance. In particular, vectors w, w' are considered of the same person if $\|w - w'\|_2$ is small. Our work is dedicated to addressing this mismatch via the dense packing feature of certain lattices in the high-dimensional vector space. For the sake of completeness, we recall the background knowledge of facial recognition, existing fuzzy extractors, and other related privacy preserving technologies before summarizing our contribution of this work.

1.2 Facial Recognition

Face embedding is a mapping from an image to a real vector (we use the term "feature vector" and "embedding" interchangeably). Ideally, neural networks are constructed and trained to make the similarity of embeddings proportional to that between the faces. Therefore, this reduces the facial recognition problem to the measurement of the distances between embeddings under a specific metric (typically in the ℓ_2-norm). The image processing flow that we use in this work is shown in Figure 2.

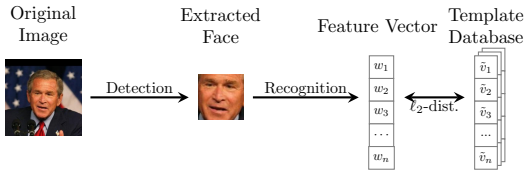


Figure 2: The image processing flow for recognition.

Face Recovery Attacks. Tan and Zhou introduced a method to reconstruct images from templates in [TZ19]. Chi Nhan Duong et al. further introduces a face recovery algorithm, which only makes black-box use of the underlying facial recognition system [DTL+20]. The work of Yang et al. can effectively generate an inversion model for a

deep neural network with only partial prediction value [YZCL19]. This line of research implies that the mapping from facial images to templates is reversible and much useful information about the original image can be recovered from the embedding, which further necessitates the privacy protection of facial templates.

1.3 Classical Fuzzy Extractors and Their Drawbacks

Dodis et al. initiated the study of fuzzy extractors as the generalization of standard randomness extractors to admit noisy sources [DRS04]. Numerous works improved on the original construction by enhancing the security guarantees (namely robustness [BDK+05, DKRS06, KR08, CDF+08] and reusability [CFP+16, Boy04, ABC+18, ACEK17, WL18a, WLH18] or both simultaneously [WL18b, WLG19]) and reducing assumptions [FMR13]. Xavier demonstrates that reusable fuzzy extractors can be applied to biometric authentication with enhanced privacy [Boy04].

Nevertheless, the aforementioned constructions mainly focus on binary sources with *Hamming* noise, i.e., two samplings of the source only differ in a few positions, or some distances like set difference and edit distance. Such characterization does not capture the small ℓ_2-norm noise in the facial recognition setting, since two vectors over a large alphabet close in ℓ_2-distance may differ in a large portion of coordinates.

A number of works in the literature address the problem of designing effective fuzzy extractors for sources with non-Hamming metric and large alphabets. Parente and van de Graaf proposed the use of Low-Density Lattice Code (LDLC) in the construction of fuzzy extractors for continuous sources [Pv16]. Their work inherently assumes Gaussian noise and lacks a concrete instantiation of the proposed system. Jana et al. proposed "Neural Fuzzy Extractors" where they design a neural network for fuzzy extractors based on LDLC [JSE+20]. Their work, nevertheless, only reported experiments on fingerprint data, and security analysis is based on an upper bound (rather than the concrete security it achieves). Buhan et al. studied fuzzy extractors on continuous source by showing an upper bound on the extracted string's length and the error rate of source in [BDHV07]. Verbitskiy et al. studied the effect of quantization methods on continuous sources fuzzy extractors in [VTO+09]. Both works lack analysis on high dimensions and concrete constructions. Zhang, Li, and Zhan used the integer lattice (i.e., $q\mathbb{Z}^n$) to design "fuzzy commitment" in [ZLZ06]. But their implementation is based on simulated Iris plant data, which is irrelevant to our application. Li et al. proposed a fuzzy extractor in the maximum norm [LGM+17]. Their experiment is not based on any actual biometric data, and it lacks concrete security analysis.

1.4 Other Related Works

Locality-Sensitive Hashing. Locality-sensitive hashing (LSH) is a way to solve the approximate nearest neighbor problem in high dimensional space [HPIM12, IM98]. Briefly speaking, it hashes similar inputs into the same "buckets" with high probability. The work of Uzun et al. [UCK+21] utilizes LSH to transform Euclidean distance into Hamming distance in the context of fuzzy-matching private set intersection. Our construction differs from the this in two ways: 1) we aim to directly design fuzzy extractors for Euclidean distance, rather than reducing it to existing construction for Hamming distance; 2) we carefully choose lattices suitable for our purpose (e.g., E8 and Leech) while LSH typically builds on random lattice [Cha02, JLY+12].

Secure Multi-party Computation. Recently Agrawal et al. proposed "External-Facing Biometric Matching" [ABMR20], where "internal-facing" authentication means that it uses the same device to store templates and take measurement of the biometrics (e.g. to unlock mobile phones) while "external-facing" authentication refers to

that the biometrics is captured by an external sensor (e.g., ATM, entrance access control system). In the latter scenario, a user carries an electronic device (e.g., mobile phone) to execute a secure multiparty computation protocol with the external sensor to complete authentication. In contrast, our work does not require the user to carry any device. Moreover, their work does not address the secure storage of biometric templates.

Homomorphic Encryption. We mention the privacy-preserving facial recognition systems based on additive homomorphic encryption and other cryptographic tools [SSW10, EFG$^+$09]. The scenario is however a bit different from ours, where a user interacts with a server (in possession of a list of templates) to find out whether the image belongs to the list. This line of research does not directly address the privacy issues considered in this work since there the templates of the server are not protected. Moreover, the result is returned to the user rather than the server, contrary to the authentication setting.

1.5 Our Contribution

Our contributions are as follows.

- We introduce a new family of secure sketch — lattice-based secure sketch[3]— and present three instantiations based on the $\mathbb{Z}^n$, E_8^m, and Leech lattices. The new constructions are compatible with the state-of-the-art fuzzy extractors [WLG19], yielding a family of fuzzy extractors for Euclidean errors. Also, any lattice with efficient CVP algorithm can be easily adapted to this framework. This motivates the explicit construction of lattices that are both dense and efficiently CVP-decodable, which might be of independent interest.
- We incorporate and implement our fuzzy extractors into the state-of-the-art model of ArcFace [DGXZ19] pretrained by an open-source project [dee18] and evaluate the system on the LFW dataset [HRBLM07, LM14].
- We concretely analyze the security from a real world dataset.
  Admittedly, the level of security provided is not satisfactory. We attribute this to not only our fuzzy extractor but more to the inherent limits of the facial recognition method (false acceptance rate). We refer to Section 4 for more details on our method and results.

A more compatible facial recognition algorithm is likely to improve not only the performance but also the security level substantially, which is left as future work.

2 Preliminaries

In this section, we introduce the notions and definitions used throughout this paper, as well as the fundamentals of lattices, fuzzy extractors, and facial recognition.

2.1 Notations

For an integer n we let $[n] \stackrel{\text{def}}{=} \{1, 2, 3, ..., n\}$. For a real number x, $\lceil x \rfloor$ denotes rounding x to the closest integer, and $\lfloor x \rfloor$ denotes rounding x to the largest integer no greater than x. Let $\log(x)$ denote the binary logarithm $\log_2(x)$. Let $a \bmod b$ denote the residue of integer a divided by integer b. To simplify our notation, for odd b, $a \bmod b$ must be in range $\{-(b-1)/2, ..., 0, ...(b-1)/2\}$. For integer vector $x = (x_1, x_2, ..., x_n)$, let $x \bmod b$ denote $(x_1 \bmod b, x_2 \bmod b, ..., x_n \bmod b)$. As a slight abuse of notation, we

[3]We note our construction does not involve any hard lattice problems as in post-quantum cryptography. In contrast, we require efficient CVP algorithms for the lattices in this paper.

let $|s|$ denote the bit length of the binary string s, let $|x|$ denote the absolute value of the real number x and let $|\mathcal{S}|$ be the size of set $\mathcal{S}$, which will be clear from the context.

2.2 Metric Spaces

A metric space $\mathcal{M}$ is a set equipped with a real numbered distance function $d : \mathcal{M} \times \mathcal{M} \to R^*$ satisfying the axioms of identity of indiscernibles, symmetry, and triangle inequality, i.e., for any x, y, $z \in \mathcal{M}$: $d(x, y) = 0 \Leftrightarrow x = y$; $d(x, y) = d(y, x)$; and $d(x, y) + d(y, z) \geq d(x, z)$.

Definition 1 (Euclidean Metric) *The Euclidean distance between $x, y \in \mathbb{R}^n$ is defined as*

$$d_2(x, y) \stackrel{\text{def}}{=} \sqrt{\sum_{i \in [n]} (x_i - y_i)^2} \ ,$$

where x_i, y_i denote the i-th elements of vectors x, y respectively.

Definition 2 (Maximum Metric) *The maximum distance between vector $x, y \in \mathbb{R}^n$ is defined as*

$$d_\infty(x, y) \stackrel{\text{def}}{=} \max_{i \in [n]} |x_i - y_i| \ ,$$

where x_i, y_i denote the i-th elements of vectors x, y respectively.

2.3 Lattice

A lattice Λ is a discrete additive subgroup of $\mathbb{R}^n$. Λ can be generated from a basis $G \in \mathbb{R}^{n \times k}$ (i.e. $\Lambda = \{Gs : s \in \mathbb{Z}^k\}$).

Closest Vector Problem. Given a lattice Λ and any point $y \in \mathbb{R}^n$, the closest vector problem or maximum likelihood decoding, is the task of finding the closest points $x \in \Lambda$ under a certain metric (e.g., Euclidean). We denote the decoding function with $\mathsf{decode}(\cdot)$.

Direct Product. Since a lattice is a special case of a group, we follow the regular notion on groups to denote $\Lambda_1 \times \Lambda_2$ as the direct product of two lattices Λ_1 and Λ_2. One can verify that $\mathsf{decode}(\Lambda_1 \times \Lambda_2) = (\mathsf{decode}(\Lambda_1), \mathsf{decode}(\Lambda_2))$. We use Λ^n to indicate $\Lambda \times ... \times \Lambda$ where it is repeated n times.

Voronoi Cell. Fixing a lattice point $y \in \Lambda$, we define its *Voronoi cell* $V_y(\Lambda)$ as the set of points in $\mathbb{R}^n$ that are closer to y than any other lattice points, i.e.,

$$V_y(\Lambda) \stackrel{\text{def}}{=} \left\{x \in \mathbb{R}^n : \forall y' \in \Lambda, y' \neq y, d(x, y) \leq d(x, y')\right\} \ ,$$

where $d(\cdot, \cdot)$ is a certain distance function. Alternatively, a Voronoi cell can be defined by decoding function,

$$V_y(\Lambda) \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n : \mathsf{decode}(x) = y\} \ .$$

We further define $V_{y'}(\Lambda)$ for a non-lattice point y' by setting

$$V_{y'}(\Lambda) \stackrel{\text{def}}{=} V_y(\Lambda) + (y' - y)$$

for arbitrary $y \in \Lambda$. We omit the subscript if y is the origin. A Voronoi cell is both centrally symmetric and convex.

2.4 Min Entropy

For a (discrete) random variable X with support $\mathcal{X}$, let $\mathbf{H}_\infty(X)$ denote the min-entropy of X, i.e.,

$$\mathbf{H}_\infty(X) = -\log \max_{x \in \mathcal{X}} p(x) \ ,$$

where $p(x) = \Pr[X = x]$.

2.5 Secure Sketch and Fuzzy Extractor

A fuzzy extractor extends the classical randomness extractor by allowing extraction from noisy sources. In particular, "fuzziness" states that strings extracted from two inputs are guaranteed to be identical if they have a small difference under some metric. Moreover, "randomness extraction" requires the output to be close to uniform randomness as long as the input has sufficient entropy.

A secure sketch is an efficient algorithm that generates auxiliary error-correcting information from noisy inputs without sacrificing too much entropy. It is a crucial building block underlying all existing fuzzy extractor constructions. Formally, we have the following definitions.

Definition 3 *An $(\mathcal{M}, m, m', t)$-secure sketch (SS) consists of a pair of PPT algorithms ($\mathsf{SS.Gen}$, $\mathsf{SS.Rec}$) described below.*

- $\mathsf{SS.Gen}(w) \mapsto s$ *: The sketch generation algorithm that outputs a sketch $s \in \mathcal{S}$ from input $w \in \mathcal{M}$;*
- $\mathsf{SS.Rec}(w', s) \mapsto \tilde{w}$ *: The recovery algorithm outputs $\tilde{w} \in \mathcal{M}$ from input $w' \in \mathcal{M}$ and a sketch $s \in \mathcal{S}$.*

They satisfy the following two properties.

Correctness. *For any $w, w' \in \mathcal{M}$, it holds that*

$$w = \mathsf{SS.Rec}(w', \mathsf{SS.Gen}(w)), \ \textit{if } d(w, w') \leq t \ .$$

Privacy. *For any distribution W over $\mathcal{M}$, it holds that*

$$\mathbf{H}_\infty(W | \mathsf{SS.Gen}(W)) \geq m', \ \textit{if } \mathbf{H}_\infty(W) \geq m \ .$$

Definition 4 *An $(\mathcal{M}, m, \mathcal{R}, t, \epsilon)$-fuzzy extractor consists of three PPT algorithms ($\mathsf{FE.Init}$, $\mathsf{FE.Gen}$, $\mathsf{FE.Rep}$).*

- $\mathsf{FE.Init}(1^\lambda) \mapsto \mathrm{pp}$ *: Generates the public parameter* pp*, which is the implicit input to $\mathsf{FE.Gen}$ and $\mathsf{FE.Rep}$;*
- $\mathsf{FE.Gen}(w) \mapsto (R, P)$ *: From input $w \in \mathcal{M}$, extracts a random string R and public information P that could help to recover R from another sampling of w;*
- $\mathsf{FE.Rep}(w', P) \mapsto R'$ *: Outputs a string R' on input $w' \in \mathcal{M}$ and the public helper P.*

They satisfy the following two properties.

Correctness. *If $d(w, w') \leq t$, for $(P, R) \leftarrow \mathsf{FE.Gen}(w)$, we have $R = \mathsf{FE.Rep}(w', P)$;*

Security. *For any distribution W over $\mathcal{M}$ s.t. $\mathbf{H}_\infty(W) \geq m$, R is indistinguishable from the uniform distribution over $\mathcal{M}$ conditioned on P.*

2.6 From Secure Sketch to Fuzzy Extractor.

Existing works in the fuzzy extractor literature show that from a secure sketch scheme (possibly with some additional properties), one can acquire a fuzzy extractor in the random oracle model, or the plain model with the LWE or DDH assumptions [Boy04, BDK+05, WLG19].

Those frameworks essentially allow us to concentrate on the construction of secure sketches and follow existing works to finish the transformation to fuzzy extractors. For simplicity and efficiency, we use a variant of the construction in [Boy04], described as Algorithm 1 and Algorithm 2, where H is a cryptographic hash function.

Algorithm 1: $\mathsf{FE.Gen}$ from $\mathsf{SS.Gen}$

Function *FE.Gen* (w)
- $r \leftarrow \{0,1\}^\lambda$;
- $s := \mathsf{SS.Gen}(w)$;
- $P = (s, r)$;
- $R := H(w, P)$;
- **return** (R, P)

Algorithm 2: $\mathsf{FE.Rep}$ from $\mathsf{SS.Rec}$

Function *FE.Rep* (w', P)
- Parse P as (s, r) ;
- $\tilde{w} := \mathsf{SS.Rec}(w', s)$;
- **return** $H(\tilde{w}, P)$

2.7 Adapting Embeddings to Fuzzy Extractors

As mentioned in Section 1.4, current facial recognition algorithms map images to *real* vectors such that similar images correspond to close vectors with respect to cosine distance or ℓ_2 metric. While real numbers can be simulated by float-point numbers in clear text, they are arguably less natural to handle in cryptography.

Therefore we apply a simple quantization step on feature vectors to effectively transform all real vectors to fix-point ones, which are equivalent to integer vectors. Thus in the following, we assume without loss of generality that the embedding algorithm returns discrete values represented as integers.

3 The Main Construction

Recall the shortcoming of previous fuzzy extractor schemes is that they support only a few metrics (Hamming distance, set difference), which do not fit with facial recognition methods. In this section, we first describe the construction of a secure sketch for ℓ_∞-metric concretely. Then, we abstract this construction to a "lattice-based" secure sketch paradigm and present an instantiation based on the E_8 lattice.

3.1 Construction for Maximum Metric

Let w be an integer vector $(w_1, w_2, ..., w_n)$. Our goal is to construct two functions $(\mathsf{SS.Gen}, \mathsf{SS.Rec})$ such that for a predefined tolerance parameter t, $\mathsf{SS.Rec}(w', \mathsf{SS.Gen}(w))$

should return exactly w if $d_\infty(w, w') \leq t$. Moreover, we want $|s|$ to be as small as possible to maximize the entropy of w condition on s.

As shown in Algorithm 3 and Algorithm 4, the message space consists of n-dim integer vector, and the public information is essentially the modulo $2t+1$ residue of the input message.

Algorithm 3: Sketch generation for ℓ_∞-metric

Function $\mathsf{SS.Gen}$ (w)
  $s := -w \bmod (2t+1)$;
  return s

Algorithm 4: Input recover for ℓ_∞-metric

Function $\mathsf{SS.Rec}$ (w', s)
  $e := (w' + s) \bmod (2t+1)$;
  $\tilde{w} := w' - e$;
  return $\tilde{w}$

The correctness and privacy of this construction is also straightforward, as shown in Theorem 3.1.

Theorem 3.1 *The* $(\mathsf{SS.Gen}, \mathsf{SS.Rec})$ *in Section 3.1 is a* $(\mathbb{Z}^n, m, m - \lceil n \log(2t+1) \rceil, t)$*-secure sketch in the* ℓ_∞*-metric.*

Proof:
For correctness, $d_\infty(w, w') \leq t$ implies $|w'_i - w_i| \leq t$, $\forall i \in [n]$. This in turn implies,

$$\begin{aligned} e &= (w' + s) \bmod (2t+1) \\ &= (w'_1 - w_1, w'_2 - w_2, \ldots, w'_n - w_n) \bmod (2t+1) \\ &= (w'_1 - w_1, w'_2 - w_2, \ldots, w'_n - w_n) \ . \end{aligned}$$

Thus,

$$\begin{aligned} \tilde{w} &= w' - e \\ &= (w'_1 - (w'_1 - w_1), w'_2 - (w'_2 - w_2), \ldots, w'_n - (w'_n - w_n)) \\ &= (w_1, w_2, \ldots, w_n) \\ &= w \ . \end{aligned}$$

For privacy,

$$\mathbf{H}_\infty(W | \mathsf{SS.Gen}(W)) \geq \mathbf{H}_\infty(W) - |s| \geq m - \lceil n \log(2t+1) \rceil \ .$$

This instantiation of a secure sketch in ℓ_∞-norm already gives rise to a facial template protection. However, as we will show later, such a scheme does not exhibit satisfactory performance when applied to pre-trained facial recognition models, which measure similarity by ℓ_2-norm rather than ℓ_∞-norm. In the following subsection, we generalize this construction to a lattice-based one that offers better approximation.

3.2 On the Generalization of Secure Sketch

Intuitively, the construction for maximum metric can be viewed as a variant of the "Syndrome-based Construction" of [DRS04] as illustrated in Figure 3. In the original construction, the function $\mathsf{decode}(\cdot)$ corresponds to the syndrome decoding algorithm of a certain error-correction code. Our generalization follows a similar route, but corrects errors over lattices instead of using the Hamming metric.

We describe the construction followed by analysis and discussions. Fix an n-dimensional integer lattice Λ. On input w, we first find the closest lattice point c of the t-scaled lattice $t\Lambda$ via invoking the CVP algorithm $\mathsf{decode}(\cdot)$, and then generates sketch $s = c - w$. The parameter $t \in \mathbb{R}$ represents the error correction capability of our construction and essentially reflects the trade-off between security and utility. To recover w from a noisy input w' and a secure sketch s, we apply the CVP algorithm on $\tilde{c} = w' + s$, calculates the difference $e = w' - w = \tilde{c} - c$, and finally outputs $w = w' - e$. Notice that the linearity of lattice Λ ensures that if $e \in V(t\Lambda)$ the difference $e = w' - w$ can be correctly recovered.

Notice that directly applying "decode" to w' would incur an error if w and w' are separated by two adjacent lattice points, as illustrated by the dotted line in Figure 3.

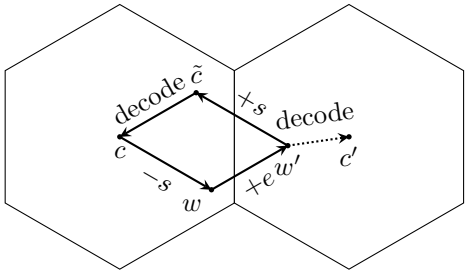


Figure 3: An intuitive illustration of the generalized secure sketch construction in the two dimensional case where decode(x) returns the lattice point closest to a given vector x.

More generally, we can define "closeness" between two points as one point lies in the scaled centrally symmetric and convex body centered at the other one. We introduce a new metric that generalizes ℓ_p-distances and fits well with our application.

Definition 5 (Centrally Symmetric Convex Body Metric) *The distance between $x, y \in \mathbb{R}^n$ is defined as*

$$d_B(x, y) \stackrel{\text{def}}{=} \min\{t \geq 0 : y - x \in t \cdot B\} \ ,$$

where B is a centrally symmetric and closed convex body centered at the origin point.

As a special case of the *Minkowski functional* [B+74], this metric can be interpreted as a family of distance functions. If B is a hypercube, this distance is equivalent to ℓ_∞. If B is a hypersphere, this distance is equivalent to ℓ_2. In particular, given a lattice Λ, its Voronoi cell $V(\Lambda)$ is both centrally symmetric and convex, and $d_V(w, w') \leq t$ implies $w' \in V_w(t\Lambda)$. We leave the proof that the distance function in Definition 5 is a metric to Appendix A.

We argue this new metric could achieve better performance because it can be regarded as an approximation of ℓ_2-metric.

We modify the definition of a traditional secure sketch by replacing $d(w, w') \leq t$ with $d_V(w, w') \leq t$. This gives rise to the generalized construction in Algorithm 5 and Algorithm 6.

Algorithm 5: Generalized Construction for $\mathsf{SS.Gen}$

Function SS.Gen (w)
- $c = \mathsf{decode}(w)$;
- $s = c - w$;
- **return** s

Algorithm 6: Generalized Construction for $\mathsf{SS.Rec}$

Function SS.Rec (w', s)
- $c' = w' + s$;
- $c^* = \mathsf{decode}(c')$;
- $\tilde{w} = c^* - s$;
- **return** $\tilde{w}$

Definition 6 *An $(\mathcal{M}, m, m', \Lambda, t)$-lattice-based secure sketch (SS) consists of a pair of PPT algorithms $(\mathsf{SS.Gen}, \mathsf{SS.Rec})$ which satisfies the following syntax and properties.*

- $\mathsf{SS.Gen}(w) \mapsto s$ *: on input $w \in \mathcal{M}$ generates a sketch $s \in \mathcal{S}$.*
- $\mathsf{SS.Rec}(w', s) \mapsto \tilde{w}$ *: on input $w' \in \mathcal{M}$ and $s \in \mathcal{S}$ outputs $\tilde{w} \in \mathcal{M}$.*
- **Correctness**. *If $d_V(w, w') \leq t$, then $w = \mathsf{SS.Rec}(w', s)$, where $s = \mathsf{SS.Gen}(w)$.*
- **Privacy**. *For any distribution W over $\mathcal{M}$ with min-entropy $\mathbf{H}_\infty(W) \geq m$, we have $\mathbf{H}_\infty(W|\mathsf{SS.Gen}(W)) \geq m'$.*

We prove that this construction meets the new definition.

Theorem 3.2 *The $(\mathsf{SS.Gen}, \mathsf{SS.Rec})$ in Section 3.2 is a $(\mathcal{M}, m, m - \log|V(t\Lambda) \cap \mathbb{Z}^n|, t)$ lattice-based secure sketch.*

Proof: For correctness, if $d_V(w, w') \leq t$ i.e. $w' \in V_w(t\Lambda)$, we have $w' - w = (w'+s) - (w+s) = c' - c$. Thus, $c' \in V_c(t\Lambda), c^* = \text{decode}(c') = c, \tilde{w} = c^* - s = c - s = w$.

For privacy, we have the size of secure sketch is bounded by integer points within a Voronoi cell and thus,

$$\begin{aligned} \mathbf{H}_\infty(W|\mathsf{SS.Gen}(W)) &\geq \mathbf{H}_\infty(W) - |s| \\ &\geq m - \log|V_c(t\Lambda) \cap \mathbb{Z}^n| \\ &= m - \log|V(t\Lambda) \cap \mathbb{Z}^n| \ . \end{aligned}$$

3.3 Instantiation of the Lattice-based Secure Sketch

As mentioned above, all we need to do is to look for a "dense" and efficiently decodable lattice. To our best knowledge, we are not aware of any explicit construction of high-dimensional lattices that are simultaneously dense and decodable in the literature of coding theory and cryptography. Therefore, we resort to good lattices in low dimensions. Nevertheless, we found that the E_8 and Leech lattices provide an improvised makeshift according to our requirements. We plan to investigate other lattices like Barnes-Wall lattice in the future.

E_8 Lattice. Let $\frac{\mathbf{1}}{\mathbf{2}}$ denote the 8-dim vector $(0.5, ..., 0.5)$. The E_8 lattice [CS13] is defined by

$$\mathrm{E}_8 \stackrel{\text{def}}{=} \mathrm{D}_8 \cup (\mathrm{D}_8 + \frac{\mathbf{1}}{\mathbf{2}}) \ ,$$

where

$$\mathrm{D}_8 \stackrel{\text{def}}{=} \{(x_1, ..., x_8) : x_i \in \mathbb{Z}, \sum_i x_i = 0 \bmod 2\} \ .$$

Algorithm 7: Unique Decoding of D_8

```
Function Decode' (x)
    x̃ = ⌈x⌋;
    if Σ_i x̃_i = 0 (mod 2) then
        return x̃;
    else
        i = argmin |x_i − x̃_i| ;
        if x̃_i > x_i then
            return x̃ + e_i;
        else
            return x̃ − e_i;
```

Algorithm 8: Unique Decoding of E_8

```
Function Decode (x)
    x_1 = Decode'(x);
    x_2 = Decode'(x − 1/2);
    return x_i closer to x for i ∈ {1, 2};
```

We use the decoding algorithm for E_8 in [CS82], which involves two decodings of D_8, as shown in Algorithm 7 and Algorithm 8. To fit the high dimension template, we may repeatedly use it m times, i.e., E_8^m.

Leech Lattice. The Leech lattice is a dense 24-dimensional lattice with an efficient decoding procedure. Due to its complexity we refer to existing works for its definition and the decoding algorithm we use in this paper. We implemented the decoding procedure for the Leech lattice in [CS86].

3.4 Fuzzy Extractor for Facial Templates

Recall that we can apply existing transforms (shown in Section 2.6) to the aforementioned secure sketches to obtain fuzzy extractors. These constructions can then be applied to human facial templates.

4 Analysis and Evaluation

In this section, we analyze the concrete security level of our facial template protection scheme by estimating the entropy of the protected biometric templates. More specifically, the unpredictability of the biometric secret (in the context of fuzzy extractor) is measured by the minimum entropy of the template conditioned on the public helper information. A major technical difficulty we encounter is to model the most likely events over a continuous random variable whose distribution is unknown. This demonstrates

the gap between theoretical security analysis and the concrete security level the system can offer.

To tackle these problems, we propose a simplified model to characterize the distribution of biometric embedding vectors. In particular, the abstraction we introduce in this model is derived from the actual functionalities of the machine-learning models that generate the embedding vectors. This reduces the problem of security evaluation of the continuous biometric template to the min-entropy evaluation of a discrete distribution (over a finite set), which is non-uniform as we can observe. Therefore, it remains to extrapolate the distribution of the biometric features from a public dataset of limited size before quantifying its min-entropy. We accomplish this task by fitting a parameterized distribution from real biometric templates in a public dataset. As a result, our analysis shows that the protected facial templates can offer roughly 45 bits of min-entropy.

4.1 Modelling the Facial Distribution

In the following, we focus on facial features as they are ubiquitous and representative. Recall that the machine-learning model returns biometric embedding vectors in a high-dimensional real space satisfying the following constraints with high probability:

- Embedding vectors of the same person allows deviations but their pair-wise distance is upper-bounded by a threshold t_1;
- Embedding vectors of different persons are lower-bounded by a threshold t_2;
- It should hold that $t_1 < t_2$ to effectively separate the cases of no-match and match.

Therefore, a machine learning model should transform the facial images of a population D to templates in the form of high-dimensional embedding vectors such that:

- Vectors of a person p should concentrate in a small region V_p. This represents the error-tolerance feature of machine-learning models.
- Regions of different persons should be pairwise disjoint for the most majority of the population.

Notice that almost all machine-learning models use the ℓ_2-norm as the distance measure of their feature vectors. In view of that existing fuzzy extractors typically adopt the Hamming metric and are thus not compatible with facial recognition applications, our construction in Section 3 can be regarded as efforts to use the lattice-based metric in Definition 5 to approximate ℓ_2 distance. The intuition is that a region $V \subseteq \mathbb{R}^n$ where every two elements are ℓ_2-distance bounded can be "enclosed" by a hypersphere. Voronoi cells of a lattice are a good approximation of non-overlapping hyperspheres in a high-dimensional space.

Based on the above observation, we formulate the following idealized assumption on the distribution of embedding vectors returned by a machine-learning model.

Assumption 1 (The model) *Given an n-dimensional rank-k lattice Λ with basis $G \in \mathbb{R}^{n\times k}$ and a facial recognition machine learning model M, the embedding vector w of a facial image returned by M can be decomposed as $w = G \cdot s + \Delta + e$ and follows the distribution:*

$$W \stackrel{\text{def}}{=} G \cdot \mathcal{S} + \mathcal{D} + \mathcal{E} \ ,$$

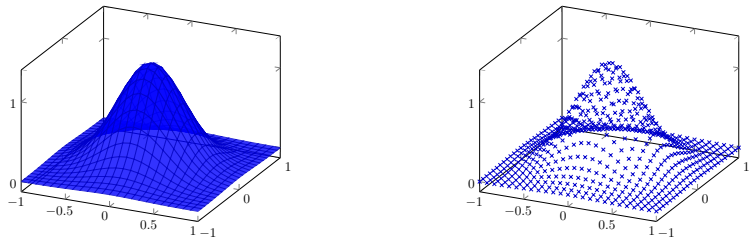
where $\mathcal{S}$ has support $\mathbb{Z}^k$, $G \cdot \mathcal{S}$ is the distribution of lattice points, $\mathcal{D}$ of support $\mathbb{R}^n$ is a offset from the center of the embedding vectors to $G \cdot s$, and $\mathcal{E}$ of support $\mathbb{R}^n$ describes the noise during measurement. We further assume that the offset distribution $\mathcal{D}$ is independent of the distribution $\mathcal{S}$ with finite support.

For the embedding $w = G \cdot s_0 + \Delta_0 + e$, one can consider (s_0, Δ_0) as the unique identifier of an individual subject to measurement noise e. We assume that the noise during the registration phase can be omitted for simplicity[4] *(i.e., $e \approx 0$). The intuition is that the biometric enrollment typically uses many measurements to reduce noise in the facial template and improve the accuracy of subsequent recognition.*

Therefore, due to the construction in Definition 6 the public information P corresponds to the offset vector Δ. As a slight abuse of notations, given $w = G \cdot s + \Delta$ and $P = \Delta$, the distribution of w conditioned on P should have its probability density all concentrated on lattice points, as illustrated in Figure 4 in the 2-dimensional case. Thus, if we denote as $\mathbf{H}_\infty(W|P)$ the inverse logarithm of the expected number of trials to find a collision given P, it holds that

$$\mathbf{H}_\infty(W|P) = \mathbf{H}_\infty(\mathcal{S}) \ .$$

To conclude, the model effectively transforms the task of estimating the unpredictability of a facial template in the form of a high dimensional real vector (given public information), into that of estimating the min-entropy of its closest lattice point. As we shall see in the following subsection, the min-entropy can be effectively estimated from existing public datasets.



(a) Continues distribution of W (b) W condition on P

Figure 4: A 2-dimensional illustration of W and W condition on P.

4.2 The Distribution of Facial Features

It remains to estimate the min-entropy from a dataset $\tilde{\mathcal{D}}$ of real-world facial samples/templates. Let $\tilde{\mathcal{S}}$ be the corresponding set of the closest lattice points, i.e., $\tilde{\mathcal{S}} = \mathsf{decode}(\tilde{\mathcal{D}})$.

First Attempt. A natural estimation is by the definition of min-entropy:

$$\tilde{\mathbf{H}}_\infty(W|P) = \tilde{\mathbf{H}}_\infty(\mathcal{S}) = -\log \max_{s \in \tilde{\mathcal{S}}} p(s) \ .$$

The limitation of this method is that the size of $\tilde{\mathcal{D}}$ becomes the bottleneck limiting the efficiency and practicality of the estimation. In particular, by measuring probabilities from the uniform distribution on $\tilde{\mathcal{D}}$, the derived min-entropy never exceeds $\log|\tilde{\mathcal{D}}|$. In other words, to estimate $\mathbf{H}_\infty(w|P)$ for w with h-bits of min-entropy, one must prepare a dataset of size at least 2^h, which is unrealistic.

Our Method. Recall that the decode algorithm is parameterized by t. We denote $\max_{s \in \tilde{\mathcal{S}}} p(s)$ by p_t, where larger values for t enable better error tolerance at the expense of potentially more leakage. Given dataset $\tilde{\mathcal{D}}$, we can set parameter t at a low enough

[4]The assumption that $e = 0$ during measurement phase is not strictly necessary but it simplifies entropy estimation. As we will see, adding more noise can only increase the unpredictability/entropy about w.

level t_0 with an acceptable false rejection rate (e.g., @70%) but templates of different people from the dataset are never mapped to the same lattice point. Intuitively, the observed distribution over lattice points, in this case, may not faithfully reflect the actual distribution of facial templates because the observed probabilities are the just inverse of the dataset size. This implies that we cannot get a meaningful estimation for p_{t_0} when the false acceptance rate in the dataset is zero at a low error tolerance level t_0.

We tackle this problem via extrapolation. In particular, we collect samples of (t, p_t) for large values of t's and fit p_t to a curve. In this way, we can estimate p_{t_0} for parameter t_0 which can not be reflected by a small database. Through experiments on several candidate distributions, we put forward our core assumption below along with justifications and empirical evidence.

Assumption 2 *p_t follows the cumulative distribution function of Johnson's S_U distribution [Joh49b, Joh49a], i.e.,*

$$\begin{aligned} p_t &= J_{\gamma,\delta,\xi,\lambda}(t) \\ &= \Phi\left(\gamma + \delta \sinh^{-1}\left(\frac{t-\xi}{\lambda}\right)\right) \end{aligned}$$

Here $J(x)$ denotes the cumulative distribution function of a Johnson's S_U distribution, Φ denotes the cumulative distribution function of the Gaussian distribution $\mathcal{N}(0,1)$ and $\gamma, \xi, \delta, \lambda$ are parameters of Johnson's S_U distribution.

This assumption is justified for the following reasons:

- Recall that the function p_t is the maximum probability of the nearest lattice points of biometric templates over a lattice scaled by a factor t. If the scale factor t is so large that all templates are included in one Voronoi cell, then all probability density concentrates on this single lattice point and we have $p_t = 1$. In the other direction, if t is small enough such that no two people are mapped to the same lattice point then p_t becomes the probability of a most likely feature vector returned by a machine learning model, which is arguably very small (i.e. $p_t \to 0$). This implies p_t satisfies the requirements of a cumulative density function.
- We choose the Johnson's S_U distribution as it best fits the data over other candidate distributions. We also use the Kolmogorov-Smirnov test [MJ51] to evaluate its suitability and find its p-value > 0.05, which empirically proves our hypothesis (i.e., p_t follows the Johnson's S_U distribution). We show an example in Figure 5 to demonstrate that the data and the CDF of the distribution fits well with each other.

Under this assumption, we can estimate parameters $\hat{\gamma}, \hat{\delta}, \hat{\xi}, \hat{\lambda}$ from the observed distribution $\tilde{\mathcal{D}}$ over a real-world dataset, and then obtain the min-entropy as:

$$\hat{\mathbf{H}}_\infty(S) = -\log J_{\hat{\gamma},\hat{\delta},\hat{\xi},\hat{\lambda}}(t) \ .$$

4.3 Experimental Setup

Hardware. We conduct our experiments on a Ubuntu 20.04 machine with Ryzen™ 5 3600 CPU and 16GB RAM.

Face Embedding. We use ArcFace [DGXZ19] to transform facial images into 512-dimensional real vectors. The models are pretrained by an open-source project [dee18].

Dataset. We conduct experiments on the Labeled Faces in the Wild (LFW) [HRBLM07,

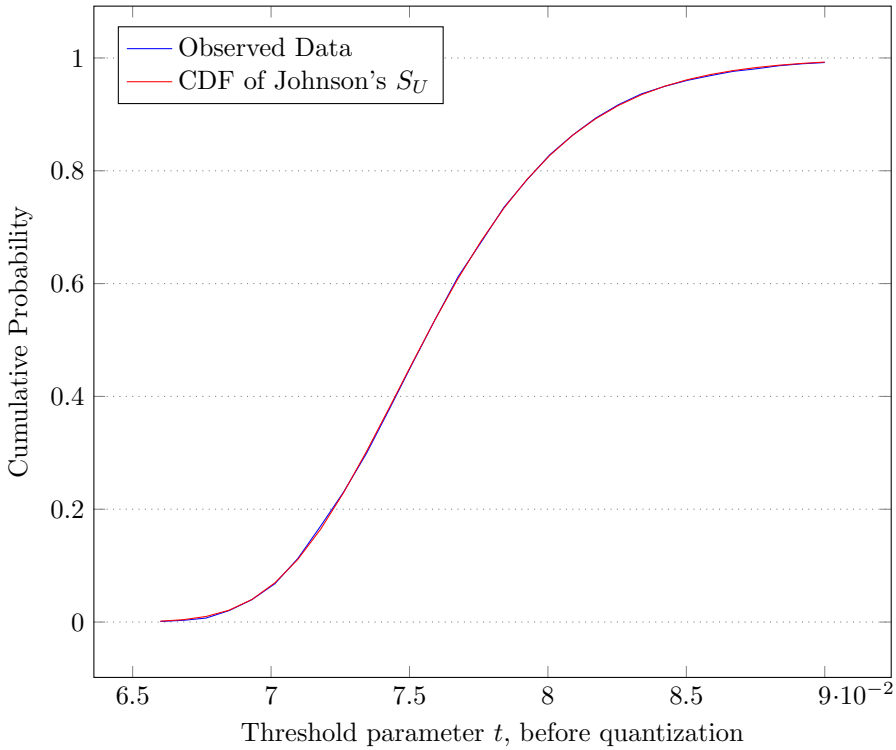


Figure 5: The curve of observed p_t and the cumulative distribution function of a Johnson's S_U distribution, where the parameters are fitted.

LM14] dataset, which is a popular benchmark for the facial recognition problem. The LFW dataset contains more than 13,000 facial images. Some examples are shown in Figure 6. We corrected several known errors in LFW [HRBLM] in order to reflect the actual performance under the extremely low false acceptance rate.

Figure 6: A few samples from the updated LFW dataset [HRBLM07, LM14].

4.4 Experimental Results and Remarks

The running time of FE.Gen and FE.Rep is both less than 100ms, which does not constitute a bottleneck compared to the inference operation of a neural network.

We report the experimental results on the LFW dataset in Table 1. The row "$\mathbb{Z}^n$" represents the construction based on the regular integer grid lattice in the max norm, while "E_8^m" and "Leech" represents the construction based on those two lattices, where $n = 512$ and $m = n/8$. The two parameter sets under each category correspond to different false rejection rate levels. The security level $\mathbf{H}_\infty$ is the number of bits of min-entropy evaluated according to the method in Section 4.2.

From Table 1 we observe the following facts:

Table 1: The experimental results of evaluation on the LFW dataset.

Method	FRR	FAR	$\mathbf{H}_\infty$
$\mathbb{Z}^n$	70%	1.8×10^{-6}	29.1
E_8^m	70%	1.8×10^{-7}	37.8
Leech	70%	2.1×10^{-7}	**45.5**
No Protection	70%	1.6×10^{-7}	N/A

- In general, our system offers a security level of $30 \sim 45$ bits, namely, any (even computationally unbounded) adversary has probability $2^{-30} \sim 2^{-45}$ in successfully guessing the embedding in one trial. Although 45-bit security may not be sufficiently strong in the cryptographic sense, it arguably offers realistic security if used in combination with computationally heavy hash functions like PBKDF2 [MKR17], bcrypt [PM99] or scrypt [PJ16]. For instance, if one trial takes one second on a PC, then the recovery of facial embedding needs more than one million years in average.
- We fix the false reject rate at 70% for system usability, i.e., the authentic individual needs three times on average to be authenticated. Increasing FRR will improve the security level at the cost of less convenience for the users.
- The security level offered by the Leech lattice norm-based construction is better than those based on E_8^m and $\mathbb{Z}^n$, which attributes to that the Leech lattice-based norm is a better approximation of ℓ_2 than E_8^m and $\mathbb{Z}^n$.

Our experiment demonstrates the feasibility of facial biometric template protection. Although the face recognition technology has been well-developed, the most advanced ones are based on proprietary models. Due to these constraints, we have to use open-source models and publicly available datasets. Nevertheless, we believe that better models and data can contribute to performance and security simultaneously.

Moreover, this new lattice-based construction reduces the task of finding a more compatible fuzzy extractor to the search for a dense lattice that admits efficient decoding algorithms in high dimensions. While the mathematics literature is rich in the constructions of dense lattices, relatively less attention is paid to efficient decoding algorithms. A lattice that satisfies both constraints implies a better fuzzy extractor and hence a better facial template protection scheme.

5 Conclusion

In this paper, we construct facial-recognition compatible fuzzy extractors from lattice-based secure sketches. We then instantiate the constructions with three concrete lattices and implement the corresponding facial template protection scheme. We perform our experiments on the publicly available LFW dataset and analyze the security level.

6 Acknowledge

- This study was funded by the National Key Research and Development Program of China (Grant Nos. 2020YFA0309705 and 2018YFA0704701) and the National Natural Science Foundation of China (Grant Nos. 62125204, 61872236, and 61971192).

- All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

[ABC+18] Quentin Alamélou, Paul-Edmond Berthier, Chloé Cachet, Stéphane Cauchie, Benjamin Fuller, Philippe Gaborit, and Sailesh Simhadri. Pseudoentropic isometries: A new framework for fuzzy extractor reusability. In Jong Kim, Gail-Joon Ahn, Seungjoo Kim, Yongdae Kim, Javier López, and Taesoo Kim, editors, *ASIACCS 18: 13th ACM Symposium on Information, Computer and Communications Security*, pages 673–684, Incheon, Republic of Korea, April 2–6, 2018. ACM Press.

[ABMR20] Shashank Agrawal, Saikrishna Badrinarayanan, Pratyay Mukherjee, and Peter Rindal. Game-set-MATCH: Using mobile devices for seamless external-facing biometric matching. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 20: 27th Conference on Computer and Communications Security*, pages 1351–1370, Virtual Event, USA, November 9–13, 2020. ACM Press.

[ACEK17] Daniel Apon, Chongwon Cho, Karim Eldefrawy, and Jonathan Katz. Efficient, reusable fuzzy extractors from LWE. Cryptology ePrint Archive, Report 2017/755, 2017. `http://eprint.iacr.org/2017/755`.

[And21] Android. Trusty tee. `https://source.android.com/security/trusty`, 2021.

[App21] Apple. About face id advanced technology. `https://support.apple.com/en-us/HT208108`, 2021.

[B+74] Sterling K Berberian et al. *Lectures in functional analysis and operator theory*, volume 15. Springer, New York, 1974.

[BDHV07] Ileana Buhan, Jeroen Doumen, Pieter H. Hartel, and Raymond N. J. Veldhuis. Fuzzy extractors for continuous distributions. In Feng Bao and Steven Miller, editors, *ASIACCS 07: 2nd ACM Symposium on Information, Computer and Communications Security*, pages 353–355, Singapore, March 20–22, 2007. ACM Press.

[BDK+05] Xavier Boyen, Yevgeniy Dodis, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Secure remote authentication using biometric data. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 147–163, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany.

[Boy04] Xavier Boyen. Reusable cryptographic fuzzy extractors. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 2004: 11th Conference on Computer and Communications Security*, pages 82–91, Washington, DC, USA, October 25–29, 2004. ACM Press.

[CDF+08] Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padró, and Daniel Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 471–488, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg, Germany.

[CFP+16] Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam D. Smith. Reusable fuzzy extractors for low-entropy distributions. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 117–146, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[Cha02] Moses Charikar. Similarity estimation techniques from rounding algorithms. In *34th Annual ACM Symposium on Theory of Computing*, pages 380–388, Montréal, Québec, Canada, May 19–21, 2002. ACM Press.

[CS82] J Conway and N Sloane. Fast quantizing and decoding and algorithms for lattice quantizers and codes. *IEEE Transactions on Information Theory*, 28(2):227–232, 1982.

[CS86] John Conway and NJAA Sloane. Soft decoding techniques for codes and lattices, including the golay code and the leech lattice. *IEEE Transactions on Information Theory*, 32(1):41–50, 1986.

[CS13] John Horton Conway and Neil James Alexander Sloane. *Sphere packings, lattices and groups*, volume 290. Springer Science & Business Media, New York, 2013.

[dee18] deepinsight. insightface. `https://github.com/deepinsight/insightface`, 2018.

[DGXZ19] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[DKRS06] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 232–250, Santa Barbara, CA, USA, August 20–24, 2006. Springer, Heidelberg, Germany.

[DRS04] 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, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

[DTL+20] C. N. Duong, T. D. Truong, K. Luu, K. G. Quach, H. Bui, and K. Roy. Vec2face: Unveil human faces from their blackbox features in face recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6131–6140, 2020.

[EFG+09] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In Ian Goldberg and Mikhail J. Atallah, editors, *PETS 2009: 9th International Symposium on Privacy Enhancing Technologies*, volume 5672 of *Lecture Notes in Computer Science*, pages 235–253, Seattle, WA, USA, August 5–7, 2009. Springer, Heidelberg, Germany.

[FMR13] Benjamin Fuller, Xianrui Meng, and Leonid Reyzin. Computational fuzzy extractors. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 174–193, Bengalore, India, December 1–5, 2013. Springer, Heidelberg, Germany.

[HPIM12] Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of computing*, 8(1):321–350, 2012.

[HRBLM] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild Home.

[HRBLM07] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[IM98] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.

[JLY+12] Jianqiu Ji, Jianmin Li, Shuicheng Yan, Bo Zhang, and Qi Tian. Super-bit locality-sensitive hashing. In *Advances in neural information processing systems*, pages 108–116, 2012.

[Joh49a] Norman L Johnson. Bivariate distributions based on simple translation systems. *Biometrika*, 36(3/4):297–304, 1949.

[Joh49b] Norman L Johnson. Systems of frequency curves generated by methods of translation. *Biometrika*, 36(1/2):149–176, 1949.

[JSE+20] Abhishek Jana, Md. Kamruzzaman Sarker, Monireh Ebrahimi, Pascal Hitzler, and George T. Amariucai. Neural fuzzy extractors: A secure way to use artificial neural networks for biometric user authentication. *CoRR*, abs/2003.08433, 2020.

[KR08] Bhavana Kanukurthi and Leonid Reyzin. An improved robust fuzzy extractor. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *SCN 08: 6th International Conference on Security in Communication Networks*, volume 5229 of *Lecture Notes in Computer Science*, pages 156–171, Amalfi, Italy, September 10–12, 2008. Springer, Heidelberg, Germany.

[LGM+17] Nan Li, Fuchun Guo, Yi Mu, Willy Susilo, and Surya Nepal. Fuzzy extractors for biometric identification. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 667–677. IEEE, 2017.

[LM14] Gary B. Huang Erik Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. Technical Report UM-CS-2014-003, University of Massachusetts, Amherst, May 2014.

[Mic21] Microsoft. Windows hello biometrics in the enterprise. `https://docs.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/hello-biometrics-in-enterprise`, 2021.

[MJ51] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[MKR17] Kathleen Moriarty, Burt Kaliski, and Andreas Rusch. PKCS #5: Password-Based Cryptography Specification Version 2.1. RFC 8018, January 2017.

[PJ16] Colin Percival and Simon Josefsson. The scrypt password-based key derivation function. *IETF Draft URL: http://tools. ietf. org/html/josefsson-scrypt-kdf-00. txt (accessed: 30.11. 2012)*, 2016.

[PM99] Niels Provos and David Mazieres. Bcrypt algorithm. In *USENIX*, 1999.

[Pv16] Vladimir P. Parente and Jeroen van de Graaf. A practical fuzzy extractor for continuous features. In Anderson C. A. Nascimento and Paulo Barreto, editors, *ICITS 16: 9th International Conference on Information Theoretic Security*, volume 10015 of *Lecture Notes in Computer Science*, pages 241–258, Tacoma, WA, USA, August 9–12, 2016. Springer, Heidelberg, Germany.

[SSW10] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. Efficient privacy-preserving face recognition. In Donghoon Lee and Seokhie Hong, editors, *ICISC 09: 12th International Conference on Information Security and Cryptology*, volume 5984 of *Lecture Notes in Computer Science*, pages 229–244, Seoul, Korea, December 2–4, 2010. Springer, Heidelberg, Germany.

[TZ19] Mingtian Tan and Zhe Zhou. Do not return similarity: Face recovery with distance, 2019.

[UCK+21] Erkam Uzun, Simon P Chung, Vladimir Kolesnikov, Alexandra Boldyreva, and Wenke Lee. Fuzzy labeled private set intersection with applications to private real-time biometric search. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, pages 911–928, 2021.

[VTO+09] E. Verbitskiy, P. Tuyls, C. Obi, B. Schoenmakers, and B. Skoric. Key extraction from general non-discrete signals. Cryptology ePrint Archive, Report 2009/303, 2009. `http://eprint.iacr.org/2009/303`.

[WL18a] Yunhua Wen and Shengli Liu. Reusable fuzzy extractor from LWE. In Willy Susilo and Guomin Yang, editors, *ACISP 18: 23rd Australasian Conference on Information Security and Privacy*, volume 10946 of *Lecture Notes in Computer Science*, pages 13–27, Wollongong, NSW, Australia, July 11–13, 2018. Springer, Heidelberg, Germany.

[WL18b] Yunhua Wen and Shengli Liu. Robustly reusable fuzzy extractor from standard assumptions. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 459–489, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Heidelberg, Germany.

[WLG19] Yunhua Wen, Shengli Liu, and Dawu Gu. Generic constructions of robustly reusable fuzzy extractor. In Dongdai Lin and Kazue Sako, editors, *PKC 2019: 22nd International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 349–378, Beijing, China, April 14–17, 2019. Springer, Heidelberg, Germany.

[WLH18] Yunhua Wen, Shengli Liu, and Shuai Han. Reusable fuzzy extractor from the decisional diffie–hellman assumption. *Designs, Codes and Cryptography*, 86(11):2495–2512, 2018.

[YZCL19] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 225–240. ACM Press, November 11–15, 2019.

[ZLZ06] Gang Zheng, Wanqing Li, and Ce Zhan. Cryptographic key generation from biometric data using lattice mapping. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 4, pages 513–516. IEEE, 2006.

A Centrally Symmetric Convex Body Distance

Theorem A.1 *The distance function in Definition 5 is a metric.*

Proof: Recall that we only need to prove that the three properties in the definition of a metric are satisfied:

Identity of indiscernibles. Obviously, $d_B(x, y) = 0 \Leftrightarrow x = y$.

Symmetry. Because the Voronoi cell B is centrally symmetric, we have $d_B(x, y) = d_B(y, x)$.

Triangle inequality. Denote $d_B(x, y), d_B(x, z), d_B(z, y)$ as t_0, t_1, t_2 respectively. And let $\vec{xy}, \vec{xz}, \vec{zy}$ go from the origin point and hit the border of P at a, b, c respectively. Therefore $\vec{xy} = \vec{xz} + \vec{zy}$ implies $t_0\vec{a} = t_1\vec{b} + t_2\vec{c}$, implies $\vec{a} = \frac{t_1}{t_0}\vec{b} + \frac{t_2}{t_0}\vec{c}$

Suppose $t_0 > t_1 + t_2$, thus $0 \leq \frac{t_1}{t_0} + \frac{t_2}{t_0} < 1$.

However, this contradicts the convex property because border *bac* is not convex.