Revisiting Fuzzy Signatures: Towards a More Risk-Free Cryptographic Authentication System based on Biometrics

Shuichi Katsumata AIST Tokyo, Japan shuichi.katsumata@aist.go.jp Takahiro Matsuda AIST Tokyo, Japan t-matsuda@aist.go.jp Wataru Nakamura Hitachi, Ltd Tokyo, Japan wataru.nakamura.va@hitachi.com

Kazuma Ohara AIST Tokyo, Japan ohara.kazuma@aist.go.jp Kenta Takahashi Hitachi, Ltd Tokyo, Japan kenta.takahashi.bw@hitachi.com

ABSTRACT

Biometric authentication is one of the promising alternatives to standard password-based authentication offering better usability and security. In this work, we revisit the biometric authentication based on *fuzzy signatures* introduced by Takahashi et al. (ACNS'15, IJIS'19). These are special types of digital signatures where the secret signing key can be a "fuzzy" data such as user's biometrics. Compared to other cryptographically secure biometric authentications as those relying on fuzzy extractors, the fuzzy signature-based scheme provides a more attractive security guarantee. However, despite their potential values, fuzzy signatures have not attracted much attention owing to their theory-oriented presentations in all prior works. For instance, the discussion on the practical feasibility of the assumptions (such as the entropy of user biometrics), which the security of fuzzy signatures hinges on, is completely missing.

In this work, we revisit fuzzy signatures and show that we can indeed efficiently and securely implement them in practice. At a high level, our contribution is threefold: (i) we provide a much simpler, more efficient, and direct construction of fuzzy signature compared to prior works; (ii) we establish novel statistical techniques to experimentally evaluate the conditions on biometrics that are required to securely instantiate fuzzy signatures; and (iii) we provide experimental results using a real-world finger-vein dataset to show that finger-veins from a single hand are sufficient to construct efficient and secure fuzzy signatures. Our performance analysis shows that in a practical scenario with 112-bits of security, the size of the signature is 1256 bytes, and the running time for signing/verification is only a few milliseconds.

CCS CONCEPTS

• Security and privacy → Digital signatures; Biometrics.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-8454-4/21/11...\$15.00 https://doi.org/10.1145/3460120.3484586

KEYWORDS

cryptographically secure biometric authentication; fuzzy signature; biometric entropy

ACM Reference Format:

Shuichi Katsumata, Takahiro Matsuda, Wataru Nakamura, Kazuma Ohara, and Kenta Takahashi. 2021. Revisiting Fuzzy Signatures: Towards a More Risk-Free Cryptographic Authentication System based on Biometrics. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), November 15–19, 2021, Virtual Event, Republic of Korea. ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3460120.3484586

1 INTRODUCTION

Background. A user authentication system is a central infrastructure in a digital society. One of the most widely used methods for authentication is those using passwords. However, today it is becoming increasingly more difficult to protect passwords and to securely manage password-based authentication from the emerging advanced forms of cyberattacks. For example, the ENISA Threat Landscape 2020 report [11] states that 64% of the publicly exposed personal data due to security breaches in 2019 contained passwords.

One of the most promising alternatives to password-based authentication that has been gradually gaining traction is biometric authentication [35], where a user's identity is verified through its biometrics such as face, iris, fingerprint, and finger-vein. A familiar example is those widely implemented on personal smartphones such as the Touch ID on iPhone. These types of authentication relies on the users holding a device embedding some information on their biometrics. In contrast, recently, biometric authentication without relying on these personal devices — (personal) device-free biometric authentication — is beginning to be deployed in commercial and governmental services. Here, anybody can authenticate using the same publicly available device. This includes for instance the facial recognition payment service Alipay managed by Alibaba in China [19], and the world's largest biometric ID system Aadhaar used in India [27]. Due to their convenience and digital inclusiveness [26], the demand for such device-freeness is expected to grow further in the future in other applications ranging from payment and ATM transactions to medical systems, immigration control, and for building a national digital identity infrastructure. The focus of this article is on such device-free biometric authentication.

In biometric authentication, the most salient problem is how to securely protect biometric information. As Visa [35] stated, the "Top concerns of using biometric authentication for payments" is "The risk of a security leak of sensitive information, e.g., you can't change your fingerprint if it is compromised." To realize a devicefree biometric authentication, biometric information is typically stored and maintained on a central server. However, this opens up the risk of exposing user biometric information due to a security breach on the server. Although standard practices such as encrypting the database and placing appropriate access control on the users can mitigate the risk, as history shows, these common procedures are not easy to enforce or to execute in real-life due to human errors or lack of a security background. For instance, a vulnerability in the Aadhaar system was recently exploited and anybody had unrestricted access to the biometric information of more than 1 billion Indian citizens [10]. Since leaking biometric information is has an irreversible damage compared to leaking passwords, minimizing the risk on the server is one of the central problems for biometric authentication.

Biometric template protection (BTP) is designed to protect such biometric information stored on a server and has been standardized in recent years (ISO/IEC 24745 [16], 30136 [17]). Fuzzy extractor (FE) [8] [16] is one of the most promising tools for constructing a biometric authentication system with BTP whose (cryptographic) security can be formally analyzed. Informally, an FE enables to extract a fixed secret key from a fuzzy biometric. Here, biometrics are inherently fuzzy objects since they can slightly change over time, and measuring them perfectly is impossible due to measurement errors. The extracted fixed secret key is then used as a secret key of an ordinary signature scheme to achieve a "biometric-based" signature scheme, which can, in turn, be used for a biometric authentication system with BTP. More accurately, a user also needs a user-specific helper data¹ to reconstruct the fixed secret key from its biometric. Intuitively, a helper data encodes some information on user biometrics to help reconstruct the same fixed secret key from the fuzzy biometric. Since the helper data does not directly reveal the secret key nor the user biometric, it is considered more secure to store the helper data rather than the user biometric on the server.

A typical flow of an FE-based device-free biometric authentication system is given in Fig. 1 (left). Each service provider (e.g., bank, supermarket, hospital) has a client device. A user can use any of these devices to authenticate itself to the server by scanning its biometrics. A user first accesses the client device and makes an ID claim to the server. The client device downloads the corresponding helper data from the server, and the user then uses its biometric to reconstruct the fixed secret key (denoted as KeyExtract in Fig. 1) used by the underlying ordinary signature scheme. Since the server only needs to store the user's helper data, the FE-based system provides BTP and successfully decreases the level of confidential information stored on the server. However, due to the added interaction between the client device and server, this opens up another type of risk. Notice that once an attacker obtains a client device, it can freely make ID claims to the server to collect the helper data of any user. Therefore, considering the attacker only needs to

steal/compromise one of the many client devices, the possibility of a database exposure is much higher compared to the naive system without BTP; the system where the biometrics are all stored on the server and the only way to retrieve them is through breaching the server. Of course, the concrete amount of biometric information leaked from the helper data in an FE depends on the specific construction and the security parameter used therein. However, in any case, we cannot take the risk zero since the adversary can collect many helper data easily and target to break any one of them; this is similar to the issue raised by reverse brute-force attacks. Thus, although the FE-based system lowers the level of confidential information stored on the server, it does so by increasing the possibility of such confidential information being exposed. Since the security risk of a system (R) is given by the product of the possibility of the data on the server being exposed (P) and the impact of such data being exposed (*I*), this brings us to our central question:

Can we lower the level of confidential information stored on the server (as in the FE-based system) while simultaneously lowering the possibility of such information being leaked (as in the naive system)? Fuzzy signature. The main primitive we focus on in this paper — fuzzy signatures — can potentially be used to solve this question. Fuzzy signatures, originally introduced in [32], are a special type of signature schemes that allow users to directly use their fuzzy biometrics as the signing key without requiring any additional information. The description of a fuzzy signature is provided in Fig. 2. Note that a verification key vkFS of a fuzzy signature is implicitly associated to the user biometric and informally holds a similar purpose as a helper data for FE.

Using a fuzzy signature, we can construct a device-free biometric authentication system as in Fig. 2 (right). From a user experience perspective, it is identical to the naive and FE-based systems: a user can show up empty-handed and authenticate itself by scanning its biometrics. In contrast, from a security point of view, the fuzzy signature-based system takes the best of the two systems: it provides BTP since the server no longer needs to store the user biometrics, and an adversary cannot collect user-specific information (e.g., helper data) since the client device and the server communicate non-interactively. Tab. 1 gives a qualitative comparison of the risk $(R) = (P) \times (I)$ of the three systems: the naive, FE-based, and fuzzy signature-based systems. It can be checked that the fuzzy signature-based biometric authentication system achieves the lowest security risk among the three systems.

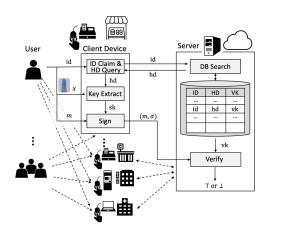
Table 1: Comparison of the security risk of three biometric authentication systems.

Caratana	Possibility of	Impact of	Risk
System	data exposure (P)	data exposure (I)	(R)
Naive	Small (from server)	Very Large (biometrics)	Large
FE-based	Large (from server + client devices)	Small [†] (hd)	Middle
FS-based	Small (from server)	Small [†] (vk _{FS})	Small

 $^{^\}dagger$ Note FE and FS are designed so that recovering biometric information from hd (helper data) and vkFS (verification key) are hard, respectively.

So far, fuzzy signatures sound all good and well. However, despite its potential values, subsequent researches on the original paper [32]

¹This is also called associated data or helper string in the literature.



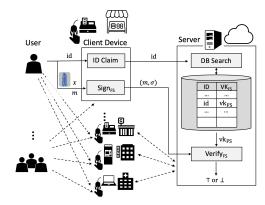


Figure 1: Authentication systems based on fuzzy extractor (left) and fuzzy signature (right). The user authenticates/signs by using its fuzzy biometric x (depicted as a finger-vein). Each service provider (depicted as a supermarket, bank, etc...) has a single or several client devices (depicted as a finger-vein scanner) and a user can use any of them to authenticate itself to the server. ID denotes the user identity, HD denotes the helper data, and VK and VKFS denote the verification key of a standard signature scheme and a fuzzy signature scheme, respectively.

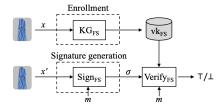


Figure 2: Description of fuzzy signature. A user with biometrics x enrolls by generating a verification key vk. The same user with a slightly different biometrics x' can sign only using x' and creates a signature σ that verifies with respect to vk generated via x.

are quite limited and only produced by a small group [18, 21, 33]². The main reason for the unfortunate disparity between its potential value and amount of related work seems to stem from the fact that fuzzy signatures are mainly presented in a very cryptographically heavy and theory-oriented manner. Indeed, the state-of-theartwork [33] provides a generic construction of fuzzy signatures but the underlying building blocks are themselves novel to their work, and it is difficult to extract the practical relevance of such construction. An equally large (or perhaps larger) issue is that a critical discussion on whether real-life biometrics can be used to securely instantiate fuzzy signatures is completely missing. [33] builds on the assumption that biometrics can provide large min-entropy. However, it is not clear whether this is a feasible assumption to make for real-world biometrics, and besides, it is not even clear how to validate the feasibility of such an assumption. Therefore, although [33] provides a potentially elegant solution to a more ideal biometric authentication system, the feasibility of the solution is completely left open to questions. We finally note that concrete discussions on biometric entropy is a reoccurring issue for FE as well and this is usually one of the main sources impeding a theoretically sound deployment of biometric authentication in practice.

1.1 Our Contribution

In this work, we show that fuzzy signatures can indeed be efficiently and securely implemented in practice, and advocate the benefit of further practice-oriented research on fuzzy signatures. Our contribution is threefold: (i) we provide a much simpler, more efficient, and direct construction of fuzzy signature compared to [33]. Very roughly, depending on the amount of min-entropy we can extract from the fuzzy biometric, our construction can be proven secure based on the standard discrete logarithm (DL) assumption or proven unconditionally secure in the generic group model [31]; (ii) we establish novel statistical techniques to experimentally evaluate the conditions on biometrics that are required to securely instantiate fuzzy signatures; and (iii) we provide experimental results using real-world finger-vein dataset to show that finger-veins from a single hand can be used to construct efficient and secure fuzzy signatures. The statistical method provided in this work is quite general so we believe this to be an independent interest for other works such as evaluating the biometric entropy to securely instantiate fuzzy extractors. Below, we expand on each of our contributions. (i) Simple and efficient construction of fuzzy signature. We provide a simple and efficient fuzzy signature scheme by tweaking the classical Schnorr signature scheme [29]. Similarly to prior works, we rely on a tool called linear sketch to bridge fuzzy biometrics and cryptographic primitives (e.g., signing keys). In our work, we simplify the definition of linear sketch and provide a conceptually cleaner construction of linear sketch based on a fundamental mathematical object called *lattices*. At a high level, the specific type of lattice being used dictates how unwastefully we use the entropy provided by the fuzzy biometrics and how well we approximate the distance metric of the fuzzy biometrics by the distance metric induced by the underlying cryptographic primitive. With this abstraction, we show that a so-called triangular lattice allows to best approximate the Euclidean distance and observe that previous works [33] implicitly used a suboptimal lattice. The security assumption that underlies the security of our fuzzy signature is a simple-to-state variant of the standard DL assumption considered

 $^{^2[33]}$ is the full-version of [32] and [21] with additional sections. Below, we mainly cite the full-version [33].

jointly with the security of a linear sketch scheme, which we coin as the DL with sketch (DL sketch) assumption. We provide discussion on the hardness of DL sketch, and give some collateral evidence that if the quantity that we call conditional false matching rate ConFMR of the distribution of fuzzy biometrics is sufficiently small, then the DL sketch assumption is implied by the standard DL assumption. Moreover, even if ConFMR is relatively large (which may be the case in practice), we show that the DL sketch assumption holds uncontionally in the generic group model [31] .

(ii) Statistical method for evaluating fuzzy biometrics. There are two conditions that fuzzy biometrics must satisfy for fuzzy signatures. As mentioned above, one is that ConFMR must be small. The other is that another quantity called the false non-matching rate FNMR must be small. Roughly, FNMR and ConFMR dictate the correctness and security of fuzzy signatures, respectively, where concretely we require FNMR $\leq 5\% (\approx 2^{-4.32})$ and ConFMR $\leq 2^{-112}$. Prior works [18, 33] failed to provide any formal evidence as to whether natural real-world biometrics can provide such amount of FNMR and ConFMR. This is a major setback for fuzzy signatures (and possibly one of the reasons why it has not attracted much serious attention) since a user may end up requiring multiple biometrics, say its iris and fingerprints of both hands, to authenticate itself. Such a procedure would severely deteriorate user experience and would defeat the purpose of using fuzzy signatures. While FNMR is a standard metric in the area of biometrics authentication and we know how to empirically estimate them using real-world biometric datasets, no such method is known for ConFMR since it is a metric intertwined with a linear sketch. To make matters worse, since ConFMR is a much smaller value (i.e., 2^{-112}) compared to FNMR (i.e., 5%), we cannot use prior methods to provide any meaningful estimations.

Thus, our second contribution is to establish a systematic procedure to evaluate the values of ConFMR of any fuzzy biometrics. We divide the problem of estimating ConFMR into two subproblems and provide details on how to solve them individually. The first subproblem is formulated in a way to detach the notion of linear sketch from ConFMR and allows us to view the problem entirely as a biometric problem, while the second subproblem deals with converting the solution of the biometric problem to the initial problem. At a high level, our approach to the two subproblems is the following: To solve the first subproblem, we borrow techniques from *extreme* value analysis (EVA), a statistical method for evaluating very rare events by using only an "extreme" subset of a given dataset [4, 30]. This allows us to estimate ConFMR $\leq 2^{-112}$ with high confidence. For the second subproblem, we use statistical *t*-tests to (informally) establish that certain statistics of biometrics are uncorrelated with the sketch.

(iii) Efficiency analysis. Finally, we use real-world finger-vein biometrics to conclude that fuzzy signatures can be constructed efficiently and securely using only 4 finger-veins from a single hand. That is, a user only needs to put one of their hands on the device to authenticate itself and nothing more. We first estimate the concrete values of FNMR and ConFMR using the method stated above and experimentally show that the conditions FNMR $\lesssim 5\%$ and ConFMR $\lesssim 2^{-112}$ hold. We then combine everything and provide a concrete set of parameters for our fuzzy signature scheme. For instance, to achieve 112-bits of security, the signature size can be as small as 1256 bytes,

and the running time for both signing and verification is only a few milliseconds.

Organization. In Sec. 2, we define fuzzy signatures and prepare the notion of *fuzzy key setting* that allows us to handle biometrics in a cryptographically sound manner. In Sec. 3, we define linear sketch: a tool allowing to bridge biometric data and cryptographic keys. In Sec. 4, we provide a simple construction of fuzzy signature based on a slight variant of the DL problem assuming that the biometrics satisfies some conditions. In Sec. 5, we equip the fuzzy key setting with a tool called lattice, and propose a concrete instantiation of a linear sketch scheme. In Sec. 6, we provide statistical techniques to estimate whether a specific type of biometrics satisfies the above mentioned conditions. Finally, in Sec. 7, we combine all the discussions together and provide a concrete instantiation of fuzzy signature using real-world finger-vein biometrics.

2 FUZZY DATA AND FUZZY SIGNATURES

To formally define fuzzy signatures, we must first formalize how we treat fuzzy data (i.e., biometrics); how are fuzzy data represented, what is the metric to argue closeness of fuzzy data, what kind of error distribution we consider to model "fuzziness" of data, and so on. To this end we first define the notion of *fuzzy key setting* below.

2.1 Fuzzy Key Setting

A fuzzy key setting $\mathcal F$ consists of the following 5-tuple $(X,\mathcal X,\mathsf{AR},\Phi,\epsilon)$ and defines all the necessary information to formally treat fuzzy data in a cryptographic scheme.

Fuzzy Data Space X: This is the space to which a possible fuzzy data x belongs. We assume that X forms an abelian group. Distribution X: The distribution of fuzzy data over X. I.e., $X : X \to \mathbb{R}$.

Acceptance Region Function $AR: X \to 2^X:$ This function maps a fuzzy data $x \in X$ to a subspace $AR(x) \subset X$ of the fuzzy data space X. (If $x' \in AR(x)$, then x' is considered "close" to x.) We require $x \in AR(x)$ for all $x \in X$. Based on AR, the *false matching rate* (FMR) and the *false non-matching rate* (FNMR) are determined. We define FMR by FMR := $Pr[x, x' \leftarrow X: x' \in AR(x)]$. FNMR is defined below.

Error Distribution Φ : This models the measurement error of fuzzy data. We assume the "universal error model" where the measurement error is independent of the users.

Error Parameter ϵ : The error parameter $\epsilon \in [0,1]$ defines FNMR, where FNMR := $\Pr[x \leftarrow X; e \leftarrow \Phi : x + e \notin AR(x)] \le \epsilon$.

2.2 Fuzzy Signatures

Using the fuzzy key setting, we can formally define fuzzy signatures. Note that in a fuzzy signature scheme, a signing key sk will not be explicitly defined since the fuzzy data x will play the role of the signing key.

Definition 2.1 (Fuzzy Signature). A fuzzy signature scheme Π_{FS} for a fuzzy key setting $\mathcal{F} = (X, X, \mathsf{AR}, \Phi, \epsilon)$ with message space \mathcal{M} is defined by the following algorithms:

FS.Setup $(1^{\kappa}, \mathcal{F}) \to pp_{FS}$: The setup algorithm takes as inputs the security parameter 1^{κ} and the fuzzy key setting \mathcal{F} as input and outputs a public parameter pp_{FS} .

- FS.KeyGen(pp_{FS}, x) \rightarrow vk_{FS}: The key generation algorithm takes as inputs the public parameter pp_{FS} and a fuzzy data $x \in X$, and outputs a verification key vk_{FS}.
- FS.Sign(pp_{FS}, x', M) $\rightarrow \sigma_{FS}$: The signing algorithm takes as inputs the public parameter pp_{FS}, a fuzzy data $x' \in X$ and a message $M \in \mathcal{M}$, and outputs a signature σ_{FS} .

We define correctness and EU-CMA security for fuzzy signatures. Roughly, correctness stipulates that a signature generated using fuzzy data x verifies with respect to a verification key generated by a fuzzy data $x' \in AR(x)$. EU-CMA security is similar to those of standard signatures except that the challenger uses $x' \in AR(x)$ to respond to signing queries rather than the original x used to generate the verification key.

Formally, we define δ -correctness and EU-CMA security of a fuzzy signature. δ -Correctness. We say a fuzzy signature scheme Π_{FS} for a fuzzy key setting \mathcal{F} is δ -correct if the following holds for all $M \in \mathcal{M}$:

$$\begin{split} \Pr[\mathsf{pp}_{\mathsf{FS}} \leftarrow \mathsf{FS.Setup}(1^\kappa, \mathcal{F}); \ x \leftarrow X; \mathsf{vk}_{\mathsf{FS}} \leftarrow \mathsf{FS.KeyGen}(\mathsf{pp}_{\mathsf{FS}}, x); \\ e \leftarrow \varPhi; \sigma_{\mathsf{FS}} \leftarrow \mathsf{FS.Sign}(\mathsf{pp}_{\mathsf{FS}}, x + e, \mathsf{M}): \\ \mathsf{FS.Vrfy}(\mathsf{pp}_{\mathsf{FS}}, \mathsf{vk}_{\mathsf{FS}}, \mathsf{M}, \sigma_{\mathsf{FS}}) = \top] \geq 1 - \delta. \end{split}$$

EU-CMA **Security.** The security of a fuzzy signature scheme Π_{FS} for a fuzzy key setting $\mathcal F$ is defined by the following game. The model captures the scenario where the signatures are generated by a slightly different fuzzy data each time.

Setup: The challenger runs $pp_{FS} \leftarrow FS.Setup(1^k, \mathcal{F}), x \leftarrow X$, $vk_{FS} \leftarrow FS.KeyGen(pp_{FS}, x)$, and provides the adversary \mathcal{A} with the public parameter pp_{FS} and the verification key vk_{FS} . Finally, it prepares an empty set $Q = \emptyset$.

Signing Queries: The adversary \mathcal{A} may adaptively submit messages. When \mathcal{A} submits a message $M \in \mathcal{M}$ to the challenger, the challenger samples $e \leftarrow \Phi$ and runs $\sigma_{FS} \leftarrow FS.Sign(pp_{FS}, x + e, M)$. It then provides pp_{FS} to \mathcal{A} and updates the set as $Q \leftarrow Q \cup \{M\}$.

Output: Finally, $\mathcal A$ outputs a pair (M^*, σ_{FS}^*) . The adversary $\mathcal A$ wins if $M^* \notin \mathcal Q \wedge FS.Vrfy(pp_{FS}, vk_{FS}, M^*, \sigma_{FS}^*) = \top$.

The advantage of $\mathcal A$ is defined as its probability of winning the above game. A fuzzy signature scheme Π_{FS} is called EU-CMA secure if the advantage is negligible for all PPT adversaries.

3 LINEAR SKETCH

In this section, we define a *linear sketch*, which has served as the main building block in previous generic constructions of fuzzy signature [33].³ Recall the main purpose of this was to "bridge" fuzzy data and standard cryptographic operations. It is associated with a fuzzy key setting and consists of two main algorithms Sketch and DiffRec (see Fig. 3). The formal definition is provided in Def. 3.1 and a high-level description of the linear sketch follows.

Overview. Sketch is used to "process" a fuzzy data x to extract a cryptographic secret (that we call a proxy key) a that is an element of some abelian group and with which actual cryptographic operations (such as the signing operation of the Schnorr signature scheme) are performed. Sketch also generates a corresponding "sketch" c of

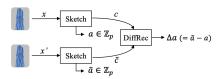


Figure 3: Illustration of the linear sketch when $\Lambda = \mathbb{Z}_p$.

x and a. The sketch c is used to "absorb" the fluctuation occurred in measuring fuzzy data. For example, suppose a fuzzy data is measured twice (e.g. once for key generation and once for signing) and sketch-proxy key pairs (c,a) and $(\widetilde{c},\widetilde{a})$ are generated (see Fig. 3). Then by using the *difference reconstruction* algorithm DiffRec with the two sketches c and \widetilde{c} , we are able to compute the difference $\Delta a = \widetilde{a} - a$. This difference Δa is then used in the verification algorithm of our fuzzy signature scheme to "adjust" the difference in the proxy keys a and \widetilde{a} . Since the proxy key a and the original fuzzy data a are used as secret information, it is naturally required that the sketch a does not reveal too much of a and a.

Definition. Formally, a linear sketch associated with a fuzzy key setting $\mathcal F$ and an abelian group Λ is defined as follows.

Definition 3.1 (Linear Sketch). Let $\mathcal{F}=(X,\mathcal{X},\mathsf{AR},\Phi,\epsilon)$ be a fuzzy key setting and $\Lambda=(\mathcal{K},+)$ be a description of a (finite) abelian group. A linear sketch scheme Π_{LinS} for \mathcal{F} and Λ is defined by the following P PT algorithms:

Lins.Setup(\mathcal{F}, Λ) \rightarrow pp_{LS}: The setup algorithm takes as input the fuzzy key setting \mathcal{F} and the description Λ , and outputs a public parameter pp_{LS}. Here, we assume pp_{LS} includes the information of $\Lambda = (\mathcal{K}, +)$.

Sketch(pp_{LS}, x) \rightarrow (c, a): The deterministic sketch algorithm takes as inputs the public parameter pp_{LS} and a fuzzy data $x \in X$, and outputs a *sketch* c and a *proxy key* $a \in \mathcal{K}$.

DiffRec(ppLS, c, \widetilde{c}) $\rightarrow \Delta a$: The deterministic difference reconstruction algorithm takes as inputs the public parameter ppLS and two sketches (c,\widetilde{c}) (supposedly output by Sketch), and outputs the *difference* $\Delta a \in \mathcal{K}$.

Correctness. We say a linear sketch scheme Π_{LinS} for a fuzzy key setting \mathcal{F} and Λ is *correct* if for all $x, x' \in X$ such that $x' \in \mathsf{AR}(x)$ and all $\mathsf{pp}_{\mathsf{LS}} \in \mathsf{LinS.Setup}(\mathcal{F}, \Lambda)$, if $(c, a) \leftarrow \mathsf{Sketch}(\mathsf{pp}_{\mathsf{LS}}, x)$ and $(\widetilde{c}, \widetilde{a}) \leftarrow \mathsf{Sketch}(\mathsf{pp}_{\mathsf{LS}}, x')$, then we have $\widetilde{a} - a = \mathsf{DiffRec}(\mathsf{pp}_{\mathsf{LS}}, c, \widetilde{c})$.

Linearity. We say a linear sketch scheme Π_{LinS} satisfies *linearity* if there exists a deterministic PT algorithm M_C satisfying the following: For all $\mathsf{pp}_\mathsf{LS} \in \mathsf{LinS}.\mathsf{Setup}(\mathcal{F},\Lambda)$ and all $x,e \in X$, if $(c,a) \leftarrow \mathsf{Sketch}(\mathsf{pp}_\mathsf{LS},x)$ and $(\widetilde{c},\Delta a) \leftarrow \mathsf{M}_\mathsf{C}(\mathsf{pp}_\mathsf{LS},c,e)$, then we have $\mathsf{Sketch}(\mathsf{pp}_\mathsf{LS},x+e) = (\widetilde{c},a+\Delta a)$.

In above, we have not formally defined the intuition that a sketch c does not leak the information of the fuzzy data x and proxy key a. This is implicitly handled by the hardness assumption underlying the security of the fuzzy signature, and we discuss it in the next section (see Def. 4.2 for an overview).

4 FUZZY SIGNATURE FROM DISCRETE LOG

In this section, we provide a simple and efficient construction of fuzzy signature based on a variant of the DL problem.

³We slightly deviate from prior definitions: we adopt a "key encapsulation"-like syntax while [33] adopts an "encryption"-like syntax. Our syntax allows for a more simple, direct, and efficient construction.

4.1 Construction

An overview of the construction of our fuzzy signature scheme $\Pi^{\mathrm{DL}}_{\mathrm{FS}}$ is depicted in Fig. 4. At a high level, our construction can be seen as providing a wrapper around the classical Schnorr signature [29] to additionally handle fuzzy biometrics via the linear sketch. During key generation (KeyGen in Fig. 4), a user with biometrics x generates a sketch and a proxy key (c, a) from x using Sketch, and sets vk_{FS} as the sketch c and a verification key $h = q^a$ of the Schnorr signature. To sign (Sign in Fig. 4), the user with biometrics x' (slightly different from x) generates (\tilde{c}, \tilde{a}) from x' using Sketch and uses $\tilde{a} \in \mathbb{Z}_p$ as an "ephemeral" signing key for the Schnorr signature and constructs a Schnorr signature $\tilde{\sigma}$. Here, note that the Schnorr verification key of this signature is implicitly set as $\tilde{h} = g^{\tilde{a}}$. The fuzzy signature σ_{FS} consists of $\tilde{\sigma}$ and the sketch \tilde{c} . Finally, to verify (Verify in Fig. 4) a fuzzy signature σ_{FS} , we first use the algorithm DiffRec of the linear sketch to recover $\Delta a = \tilde{a} - a$. Then, we use Δa to recover the implicit Schnorr verification key \tilde{h} from h, and use it to verify $\tilde{\sigma}$.

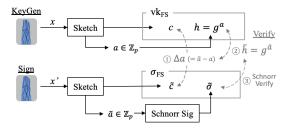


Figure 4: Our fuzzy signature scheme Π_{FS}^{DL} . The gray items indicate the procedure of the verification algorithm.

The formal description of our fuzzy signature scheme is provided in Fig. 5. The image of the hash function H is \mathbb{Z}_p and is modeled as a random oracle in the security proof.

$FS.Setup(1^\kappa,\mathcal{F})$:	$FS.KeyGen(pp_{FS},x)$
1: $\mathcal{G} \leftarrow GGen(1^{\kappa})$ 2: $\Lambda = (\mathbb{Z}_p, +)$ 3: $pp_{LS} \leftarrow LinS.Setup(\mathcal{F}, \Lambda)$ 4: $return \ pp_{FS} = (\mathcal{G}, pp_{LS})$	1: $(\mathcal{G}, pp_LS) \leftarrow pp_FS$ 2: $(c, a) \leftarrow Sketch(pp_LS, x)$ 3: return $vk_FS = (g^a, c)$
$FS.Sign(pp_{FS}, x', M)$:	$FS.Vrfy(pp_{FS}, vk_{FS}, M, \sigma_{FS})$
1: $(\mathcal{G}, \operatorname{pp}_{LS}) \leftarrow \operatorname{pp}_{FS}$ 2: $(\tilde{c}, \tilde{a}) \leftarrow \operatorname{Sketch}(\operatorname{pp}_{LS}, x')$ 3: $r \leftarrow \mathbb{Z}_p$ 4: $\beta \leftarrow \operatorname{H}(g^{\tilde{a}}, g^r, M)$ 5: $z \leftarrow \beta \cdot \tilde{a} + r$ 6: $\operatorname{return} \sigma_{FS} = (\beta, z, \tilde{c})$	1: $(\mathcal{G}, pp_{LS}) \leftarrow pp_{FS}$ 2: $(h, c) \leftarrow vk_{FS}$ 3: $(\beta, z, \widetilde{c}) \leftarrow \sigma_{FS}$ 4: $\Delta a \leftarrow DiffRec(pp_{LS}, c, \widetilde{c})$ 5: $\tilde{h} \leftarrow h \cdot g^{\Delta a}$ 6: $R \leftarrow g^z \cdot \tilde{h}^{-\beta}$
	7: return \top iff $\beta = H(h, R, M)$

Figure 5: Construction of fuzzy signature $\Pi^{DL}_{FS}.$

Efficiency. The verification key consists of one group element in \mathbb{G} and a sketch. The signature consists of two elements in \mathbb{Z}_p and a sketch. Notably, the only difference from the Schnorr signature is the linear sketch component.

4.2 Correctness and Security Proof

Correctness. The correctness of Π^{DL}_{FS} is provided below. As correctness is evident from Fig. 4, we omit the proof to App. B.

Theorem 4.1. If the linear sketch Π_{LinS} is correct, then the fuzzy signature Π_{FS}^{DL} in Fig. 5 is ϵ -correct, where ϵ is the error parameter of the fuzzy key setting \mathcal{F} .

Security. The security of our fuzzy signature Π_{FS}^{DL} is based on a variant of the DL problem where the secret exponent is a proxy key a generated by the linear sketch scheme on input a random fuzzy data $x \leftarrow X$. The adversary is given the DL instance g^a along with the sketch c. Formally, we define the DL with sketch assumption in Def. 4.2. We provide detailed discussions in Sec. 5.4 to validate that the DL with sketch problem is as hard as the standard DL problem for our specific choice of biometrics and the linear sketch scheme.

Definition 4.2 (DL with sketch). Let Π_{LinS} be a linear sketch scheme for a fuzzy key setting $\mathcal{F} = (X, X, \mathsf{AR}, \Phi, \epsilon)$ with respect to a (finite) abelian group $\Lambda = (\mathbb{Z}_p, +)$. We say the discrete logarithm problem with sketch (DL sketch) assumption holds (relative to GGen) if for all PPT adversaries \mathcal{A} , the following probability is upper bounded by $\mathsf{negl}(\kappa)$:

$$\Pr\left[\begin{array}{c} \mathcal{G} = (\mathbb{G}, p, g) \leftarrow \mathsf{GGen}(1^\kappa); \\ \mathsf{pp}_\mathsf{LS} \leftarrow \mathsf{LinS.Setup}(\mathcal{F}, \Lambda); & : \quad \mathcal{A}(\mathcal{G}, \mathsf{pp}_\mathsf{LS}, g^a, c) = a \\ x \leftarrow \mathcal{X}; (c, a) \leftarrow \mathsf{Sketch}(\mathsf{pp}_\mathsf{LS}, x) \end{array}\right].$$

The following theorem guarantees security of our fuzzy signature scheme Π_{FS}^{DL} under the DL^{sketch} assumption.

Theorem 4.3. If the DL^{sketch} problem is hard and the linear sketch scheme Π_{LinS} satisfies linearity, then the fuzzy signature scheme Π_{FS}^{DL} in Fig. 5 is EU-CMA secure.

The proof is similar to that of the Schnorr signature [29], except that we additionally need to simulate the sketch c in the verification key and signatures without knowledge of the secret fuzzy data x. At a high level, the sketch in the verification key is handled by the DL sketch assumption and the sketches in the signatures are handled by the *linearity* of the linear sketch (see Def. 3.1). We omit the full proof to App. B.

5 INSTANTIATING LINEAR SKETCH OVER LATTICES

In this section, we present our linear sketch scheme. Our scheme is constructed over a fuzzy key setting with fuzzy data space $X = \mathbb{R}^{n}$. Namely, we consider the natural setting where biometrics are represented by an n-dimensional vector in \mathbb{R} . However, working directly with fuzzy data in \mathbb{R}^n is non-trivial since typical computations of cryptographic primitives (and in particular the Schnorr signature scheme) are performed over a discrete space such as \mathbb{Z}_p . Moreover, recall that a linear sketch scheme needs to satisfy correctness and linearity, which roughly requires a *linearity preserving* mapping of the fuzzy data space X to the sketch and proxy key spaces. To deal with these issues, we associate the fuzzy data space X with a mathematical object called lattice known to have a discretized and

 $^{^4{\}rm Throughout}$ the rest of the paper we implicitly assume that real numbers are represented by some pre-determined number of significant digits in order to handle them on computers.

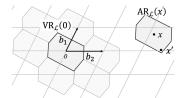
linear nature. This connects fuzzy data and cryptographic primitives together, and allows to construct a linear sketch scheme.

We also introduce a specific lattice called a *triangular* lattice and show that it fits well with a fuzzy data space endowed by Euclidean metric. We finally discuss the hardness of the DL^{sketch} assumption with respect to such concrete linear sketch scheme in Sec. 5.4.

5.1 Fuzzy Key Setting with a Lattice

We first introduce the notion of lattices and then provide a concrete definition of a fuzzy key setting based on lattices. Lattice background. Let $n \in \mathbb{N}$ and $\mathbf{B} \in \mathbb{R}^{n \times n}$.

- A *lattice* spanned by B, denoted by $\mathcal{L}(B)$, is defined by $\mathcal{L}(B) := \{Bz | z \in \mathbb{Z}^n\}$. B is called the *basis* of $\mathcal{L}(B)$.
- For a vector $\mathbf{x} \in \mathbb{R}^n$ and a lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$, the *closest vector* (or *lattice point*) of \mathbf{x} in \mathcal{L} , denoted by $\mathsf{CV}_{\mathcal{L}}(\mathbf{x})$, is a vector $\mathbf{y} \in \mathcal{L}$ satisfying $\|\mathbf{x} \mathbf{y}\|_2 \le \|\mathbf{x} \mathbf{Bz}\|_2$ for any $\mathbf{z} \in \mathbb{Z}^{n.5}$
- For a lattice $\mathcal{L} = \mathcal{L}(B)$ and a vector $y \in \mathcal{L}$, the *Voronoi region* of y, denoted by $VR_{\mathcal{L}}(y)$, is defined by $VR_{\mathcal{L}}(y) := \{x|y = CV_{\mathcal{L}}(x)\}$. Due to the translational symmetry of a lattice, we have $VR_{\mathcal{L}}(y) = VR_{\mathcal{L}}(0) + y$. See Fig. 6 for a pictorial example of $VR_{\mathcal{L}}(0)$.



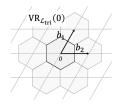


Figure 6: The left and right light gray grids depict two different lattices $\mathcal L$ and $\mathcal L_{\text{tri}}.$ The regions $\mathsf{VR}_{\mathcal L}(0)$ is the set of points that has 0 as the closest vector in the lattices and the region $\mathsf{AR}_{\mathcal L}(x)$ denotes all the point that is considered to be "close" to x.

Fuzzy key setting based on a lattice. We define a fuzzy key setting $\overline{\mathcal{F}} = (X, X, AR, \Phi, \epsilon)$ with respect to a lattice as follows.

Fuzzy data space X: The fuzzy data space X is \mathbb{R}^n , where $n \in \mathbb{N}$ is specified by the context (e.g. the device which we use to measure fuzzy data). We associate X with a lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$ spanned by some basis $\mathbf{B} \in \mathbb{R}^{n \times n}$ such that the closest vectors in \mathcal{L} can be *efficiently* computed. We also associate X with a natural number $p \in \mathbb{N}$ that determines the support of \mathcal{X} (see below).

Distribution X: An efficiently sampleable distribution such that the support of X satisfies the property that if $\mathbf{x} \leftarrow X$, then $\mathbf{B}^{-1}\mathbf{x} \in [0,p)^n$.

Acceptance region function AR: We define the acceptance region function AR by $\mathsf{AR}(x) = \mathsf{AR}_{\mathcal{L}}(x) := \{x' | \mathsf{CV}_{\mathcal{L}}(x - x') = \mathbf{0}\}$. Note that we have $\mathsf{AR}_{\mathcal{L}}(x) = \mathsf{VR}_{\mathcal{L}}(\mathbf{0}) + x$. See Fig. 6 for a pictorial example of $\mathsf{AR}_{\mathcal{L}}(x)$.

Error distribution Φ and Error parameter $\epsilon\colon \Phi$ is any efficiently samplable distribution over X such that FNMR $\leq \epsilon$.

5.2 Construction of Linear Sketch

Let $\mathcal{F}=(X=\mathbb{R}^n,\mathcal{X},\mathsf{AR},\Phi,\epsilon)$ be the fuzzy key setting as defined above. Let $g_{\mathcal{L}}:X\to\mathcal{L}$ be the function $g_{\mathcal{L}}(\mathbf{x}):=\mathbf{B}\lfloor\mathbf{B}^{-1}\mathbf{x}\rfloor$. Let $\mathcal{UH}=\{\mathsf{UH}:\mathbb{Z}_p^n\to\mathbb{Z}_p\}$ be a family of universal hash functions that satisfies linearity, namely, for all $\mathsf{UH}\in\mathcal{H}$ and $\mathsf{x},\mathsf{y}\in\mathbb{Z}_p^n$, we have $\mathsf{UH}(\mathsf{x}+\mathsf{y})=\mathsf{UH}(\mathsf{x})+\mathsf{UH}(\mathsf{y})$. Using these ingredients, the description of our linear sketch scheme $\Pi_{\mathsf{LinS}}=(\mathsf{LinS}.\mathsf{Setup},\mathsf{Sketch},\mathsf{DiffRec})$ for \mathcal{F} and the additive group $(\mathbb{Z}_p,+)$ (=: Λ) is provided in Fig. 7. The auxiliary algorithm M_c used to show the linearity property is also included. A pictorial example (Fig. 8) and an intuitive explanation of Sketch and DiffRec follow.

$LinS.Setup(\mathcal{F},\Lambda=(\mathbb{Z}_p,+))\colon$	$Sketch(pp_{LS}, \mathbf{x})$
1: UH \leftarrow \mathcal{UH}	1: $\mathbf{y} \leftarrow g_{\mathcal{L}}(\mathbf{x})$
2: return $pp_{1S} = (\Lambda, UH)$	2: $\mathbf{c} \leftarrow \mathbf{x} - \mathbf{y}$
1 2 20	$3: a \leftarrow UH(\mathbf{B}^{-1}\mathbf{y})$
	4: return (c , <i>a</i>)
$DiffRec(pp_LS, \mathbf{c}, \mathbf{c'})$:	$M_c(pp_{LS}, c, e)$
1: $\Delta \mathbf{y} \leftarrow CV_{\mathcal{L}}(\mathbf{c} - \mathbf{c'})$	1: $\mathbf{c'} \leftarrow \mathbf{c} + \mathbf{e} - g_{\mathcal{L}}(\mathbf{c} + \mathbf{e})$
2: $\Delta a \leftarrow UH(\mathbf{B}^{-1}(\Delta \mathbf{y}))$	2: $\mathbf{y'} \leftarrow g_{\mathcal{L}}(\mathbf{c} + \mathbf{e})$
3: return Δa	3: $\Delta a \leftarrow UH(\mathbf{B}^{-1}\mathbf{y'})$
	4: return $(\mathbf{c}', \Delta a)$

Figure 7: Construction of linear sketch Π_{LinS} and the auxiliary algorithm M_c for the linearity property.

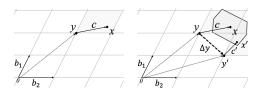


Figure 8: The left (resp. right) figure depicts algorithm Sketch (resp. DiffRec). The shaded gray parallelogram denotes the fundamental parallelepiped spanned by the basis $B = [b_1, b_2]$. The gray hexagon denotes the acceptance region $AR(\mathbf{x})$ of \mathbf{x} .

Algorithm Sketch(pp_{LS}, x) first deterministically computes a lattice point $y = g_{\mathcal{L}}(x) \in \mathcal{L}$ with respect to the basis B. As depicted in Fig. 8, the fundamental parallelepiped spanned by B^9 originated at y always contains x. Then, the sketch c is simply the shift x - y and the proxy key a is a hash of some "canonical representation" of the lattice point y. Now, it is clear from Fig. 8 that if $x' \in AR(x)$ is contained in the same fundamental parallelepiped originated at y, then it produces the same proxy key a since $y = g_{\mathcal{L}}(x')$. However, as in the right figure in Fig. 8, this is not always the case. Therefore, we require a mechanism to relate the proxy key

 $^{^5}$ If there are multiple vectors $y \in \mathcal{L}$ satisfying this condition, then we consider some canonical ordering of the lattice points in \mathcal{L} and choose the first one according to the ordering to make it unique.

⁶For $X \subset \mathbb{R}^n$ and y, we define $X + y =: \{x + y | x \in X\}$.

 $^{^{7}}$ As far as correctness and linearity are concerned, the function $g_{\mathcal{L}}$ can be any efficiently computable deterministic function satisfying (1) $g_{\mathcal{L}}(\mathbf{x}+\mathbf{y})=g_{\mathcal{L}}(\mathbf{x})+\mathbf{y}$ for all $\mathbf{x}\in X$ and $\mathbf{y}\in \mathcal{L}$, and (2) $\|\mathbf{B}^{-1}\mathbf{x}\|_{\infty}\approx \|\mathbf{B}^{-1}g_{\mathcal{L}}(\mathbf{x})\|_{\infty}$. We choose this particular function for its simplicity and efficiency.

⁸Recall that $\dot{\mathcal{UH}} = \{\mathsf{UH} : D \to R\}$ is called universal if for all distinct elements $x, x' \in D$, we have $\mathsf{Pr}_{\mathsf{UH} \leftarrow \mathcal{UH}}[\mathsf{UH}(x) = \mathsf{UH}(x')] \leq |R|^{-1}$.

⁹The fundamental parallelepiped spanned by **B** is defined as the set $\{\mathbf{B}\mathbf{w} \mid \mathbf{w} \in [0,1)^n\}$.

a' (or equivalently y') generated by x' and those by x only given their sketches c' and c. Recall that this was the core property of linear sketch that allowed us to meaningfully relate the secret keys generated from different a and a' for our fuzzy signature scheme (see Fig. 3). Now, algorithm DiffRec(pp_{LS}, c, c') exactly offers this mechanism. First, by definition c - c' = (x - x') + (y' - y). Then since the vector $x - x' \in VR(0)$ (see Fig. 8), $CV_{\mathcal{L}}(c - c')$ is the same as $CV_{\mathcal{L}}(y' - y) = y' - y$ since y' - y are points contained in \mathcal{L} . Hence, we can recover $\Delta y = y' - y$ (or equivalently $\Delta a = a' - a$) only from the sketches c and c'.

Formally, we have the following theorem.

Theorem 5.1. The linear sketch scheme Π_{LinS} in Fig. 7 satisfies correctness and linearity (as per Def. 3.1).

PROOF. To prove the theorem, we show that Π_{LinS} given in Sec. 5 satisfies correctness and linearity (Def. 3.1).

<u>Correctness.</u> Fix $pp_{LS} = (\Lambda = (\mathbb{Z}_p, +), UH)$, and $\mathbf{x}, \mathbf{x}' \in X$ such that $\mathbf{x}' \in AR(\mathbf{x})$, which implies $CV_{\mathcal{L}}(\mathbf{x} - \mathbf{x}') = \mathbf{0}$. Let $\mathbf{y} = g_{\mathcal{L}}(\mathbf{x})$ and $\mathbf{y}' = g_{\mathcal{L}}(\mathbf{x}')$, and let

$$(\mathbf{c}, a) = (\mathbf{x} - \mathbf{y}, \mathsf{UH}(\mathbf{B}^{-1}\mathbf{y})) = \mathsf{Sketch}(\mathsf{pp}_{\mathsf{LS}}, \mathbf{x}),$$

 $(\mathbf{c}', a') = (\mathbf{x}' - \mathbf{y}', \mathsf{UH}(\mathbf{B}^{-1}\mathbf{y}')) = \mathsf{Sketch}(\mathsf{pp}_{\mathsf{LS}}, \mathbf{x}').$

We have

$$\Delta \mathbf{y} = \mathsf{CV}_{\mathcal{L}}(\mathbf{c} - \mathbf{c}') = \mathsf{CV}_{\mathcal{L}}((\mathbf{x} - \mathbf{y}) - (\mathbf{x}' - \mathbf{y}'))$$

$$\stackrel{(*)}{=} \mathsf{CV}_{\mathcal{L}}(\mathbf{x} - \mathbf{x}') + \mathbf{y}' - \mathbf{y} \stackrel{(**)}{=} \mathbf{y}' - \mathbf{y},$$

where the equality (*) uses the fact that $y, y' \in \mathcal{L}$, and the equality (**) uses $CV_{\mathcal{L}}(x - x') = 0$. Using this, we see that

$$\Delta a = \mathsf{UH}\big(\mathbf{B}^{-1}(\Delta \mathbf{y})\big) = \mathsf{UH}\big(\mathbf{B}^{-1}(\mathbf{y'} - \mathbf{y})\big)$$

$$\stackrel{(*)}{=} \mathsf{UH}(\mathbf{B}^{-1}\mathbf{y'}) - \mathsf{UH}(\mathbf{B}^{-1}\mathbf{y}) = a' - a,$$

where the equality (*) uses the linearity of \mathcal{UH} . This shows that DiffRec(pp_{LS}, c, c') = a' - a. Thus, Π_{LinS} satisfies correctness.

<u>Linearity.</u> We use the auxiliary algorithm M_c in Fig. 7. Fix $pp_{LS} = \overline{(\Lambda = (\mathbb{Z}_p, +), UH)}$ and $\mathbf{x}, \mathbf{e} \in X$. Let

$$\begin{split} (\mathbf{c}, a) &= (\mathbf{x} - g_{\mathcal{L}}(\mathbf{x}), \mathsf{UH}(\mathbf{B}^{-1}g_{\mathcal{L}}(\mathbf{x}))) = \mathsf{Sketch}(\mathsf{pp}_{\mathsf{LS}}, \mathbf{x}), \\ (\mathbf{c}', a') &= ((\mathbf{x} + \mathbf{e}) - g_{\mathcal{L}}(\mathbf{x} + \mathbf{e}), \mathsf{UH}(\mathbf{B}^{-1}g_{\mathcal{L}}(\mathbf{x} + \mathbf{e}))) \\ &= \mathsf{Sketch}(\mathsf{pp}_{\mathsf{LS}}, \mathbf{x} + \mathbf{e}) \end{split}$$

In order to show the linearity of Π_{LinS} , it is sufficient to show that the following equality holds:

$$(\mathbf{c} + \mathbf{e} - g_{\mathcal{L}}(\mathbf{c} - \mathbf{e}), \ \mathsf{UH}(\mathbf{B}^{-1}g_{\mathcal{L}}(\mathbf{c} + \mathbf{e})) = (\mathbf{c}', a' - a),$$
 (1)

since the left hand sice is exactly $M_c(pp_{1.5}, c, e)$.

For the first element in Eq. (1), we have

$$\mathbf{c} + \mathbf{e} - g_{\mathcal{L}}(\mathbf{c} - \mathbf{e}) = \mathbf{x} - g_{\mathcal{L}}(\mathbf{x}) + \mathbf{e} - g_{\mathcal{L}}(\mathbf{x} - g_{\mathcal{L}}(\mathbf{x}) + \mathbf{e})$$

$$\stackrel{(*)}{=} \mathbf{x} - g_{\mathcal{L}}(\mathbf{x}) + \mathbf{e} - (g_{\mathcal{L}}(\mathbf{x} + \mathbf{e}) - g_{\mathcal{L}}(\mathbf{x}))$$

$$= \mathbf{x} + \mathbf{e} - g_{\mathcal{L}}(\mathbf{x} + \mathbf{e}) = \mathbf{c}',$$

where the equality (*) uses the property of $g_{\mathcal{L}}$ that $g_{\mathcal{L}}(\mathbf{x}' + \mathbf{y}') = g_{\mathcal{L}}(\mathbf{x}') + \mathbf{y}'$ for $\mathbf{x}' \in X$ and $\mathbf{y}' \in \mathcal{L}$, and that $g_{\mathcal{L}}(\mathbf{x}) \in \mathcal{L}$.

For the second element in Eq. (1), we have

$$\mathsf{UH}\big(\mathsf{B}^{-1}g_{\mathcal{L}}(\mathsf{c}+\mathsf{e})\big) = \mathsf{UH}\big(\mathsf{B}^{-1}(g_{\mathcal{L}}(\mathsf{x}-g_{\mathcal{L}}(\mathsf{x})+\mathsf{e}))\big)$$

$$\stackrel{(*)}{=} \mathsf{UH} \big(\mathsf{B}^{-1} (g_{\mathcal{L}}(\mathsf{x} + \mathsf{e}) - g_{\mathcal{L}}(\mathsf{x})) \big)$$

$$= \mathsf{UH} \big(\mathsf{B}^{-1} g_{\mathcal{L}}(\mathsf{x} + \mathsf{e}) - \mathsf{B}^{-1} g_{\mathcal{L}}(\mathsf{x}) \big)$$

$$\stackrel{(**)}{=} \mathsf{UH} (\mathsf{B}^{-1} g_{\mathcal{L}}(\mathsf{x} + \mathsf{e})) - \mathsf{UH} (\mathsf{B}^{-1} g_{\mathcal{L}}(\mathsf{x}))$$

$$= a' - a,$$

where the equality again uses the property of $g_{\mathcal{L}}$ that $g_{\mathcal{L}}(\mathbf{x}'+\mathbf{y}') = g_{\mathcal{L}}(\mathbf{x}') + \mathbf{y}'$ for $\mathbf{x}' \in X$ and $\mathbf{y}' \in \mathcal{L}$, and the equality (**) uses the linearity of \mathcal{UH} . Hence, we can conclude that Π_{LinS} satisfies linearity.

5.3 Concrete Lattice for Efficient Linear Sketch

Depending on the type of lattice \mathcal{L} (or equivalently basis B), the computational complexity of $CV_{\mathcal{L}}$ and $g_{\mathcal{L}}$ differs greatly. For our concrete instantiation of linear sketch, we use triangular lattices. Geometrically, they are lattices that have regular hexagons as the Voronoi region VR (see the right hand side of Fig. 6 for an illustration). Over such a lattice, $\mathsf{CV}_{\mathcal{L}_\mathsf{tri}}$ can be computed in time $O(n^2)$. Moreover, other than they allow for efficient computations of $CV_{\mathcal{L}_{tri}}$ and $g_{\mathcal{L}_{tri}}$, the acceptance region $AR_{\mathcal{L}_{tri}}$ of triangular lattices reflects nicely the notion of "closeness" of most natural biometrics. In a typical biometric authentication, the most natural and widely-used way to judge two biometrics x, x' are "close" is to calculate how close they are with respect to the Euclidean distance $\|\mathbf{x} - \mathbf{x}'\|_2$. The triangular lattice is a very suitable lattice in the sense that AR f_{tri} (which is a regular hexagon) best approximates the closeness induced by the Euclidean distance compared to any other lattice \mathcal{L} . We note that casting the linear sketch schemes in [33] in the framework of lattices, we see that they considered lattices with a square as the AR f (i.e., a lattice with basis $\mathbf{B} = d \cdot \mathbf{I}_n$ for some positive real $d \in \mathbb{R}$). See Fig. 10 for a visual aid. Effectively, our lattice allows to extract more entropy from the underlying biometric since we are able to model more accurately the real closeness metric.

A formal description of triangular lattices and how $\text{CV}_{\mathcal{L}_{\text{tri}}}$ is implemented are provided in App. C.

5.4 Security of the DL Assumption with Sketch

In Sec. 4, we introduced the DL^{sketch} assumption on which the security of our fuzzy signature scheme is based. The main question is of course: how plausible is this assumption? We argue that for our linear sketch scheme Π_{LinS} presented in this section, the DL^{sketch} assumption is plausible if:

- the quantity that we call the *conditional false matching rate* (ConFMR) is "small", say, $\approx 2^{-\kappa}$ for a cryptographic security parameter κ , and
- the standard DL assumption holds.

Here, for the linear sketch scheme Π_{LinS} over a fuzzy key setting $\mathcal{F} = (X, \mathcal{X}, \mathsf{AR}, \Phi, \epsilon)$ with which a lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$ is associated, we define ConFMR by

$$\text{ConFMR} := \Pr \left[\begin{array}{cc} x, x \leftarrow \mathcal{X}; c \leftarrow x - g_{\mathcal{L}}(x); \\ c' \leftarrow x' - g_{\mathcal{L}}(x') \end{array} \right] : x' \in \mathsf{AR}(x) \middle| c = c' \right].$$

In other words, ConFMR is the conditional probability that \mathbf{x}' belongs to AR(\mathbf{x}) conditioned on the event that their sketch values $\mathbf{c} = \mathbf{x} - g_{\mathcal{L}}(\mathbf{x})$ and $\mathbf{c}' = \mathbf{x}' - g_{\mathcal{L}}(\mathbf{x}')$ are identical.

Our argument is based on the following two facts:

- (1) If ConFMR $\lessapprox 2^{-(2\kappa+\omega(\log\kappa))}$, then the standard DL assumption implies the DL^{sketch} assumption;
- (2) If ConFMR $\approx 2^{-\kappa}$ (or even $2^{-\omega(\log \kappa)}$), then the DL^{sketch} assumption holds in the generic group model [31].

We give an explanation for each item. Below, recall that for a joint distribution (X, C), the (average) *conditional collision entropy* of X given C is defined by $\tilde{H}_2(X|C) := -\log_2 COL(X|C)$, where

$$COL(X|C) := Pr[(x,c), (x',c') \leftarrow (X,C) : x = x'|c = c'].$$
 (2)

Here, COL(X|C) is called the *conditional collision probability of X* given C. When the context is clear, we often abuse notation and write $\tilde{H}_2(x|c)$ instead of $\tilde{H}_2(X|C)$, and we do a similar treatment for COL.

(1) DL implies DLsketch when ConFMR is sufficiently small. Identifying the joint distribution (\mathcal{X}, C) in Eq. (2) with $\{\mathbf{x} \leftarrow \mathcal{X} : (\mathbf{x}, \mathbf{c} = \mathbf{x} - g_{\mathcal{L}}(\mathbf{x}))\}$, we clearly have COL $(\mathbf{x}|\mathbf{c}) \leq \text{ConFMR}$. Moreover, observe COL $(\mathbf{x}|\mathbf{c}) = \text{COL}(\mathbf{B}^{-1}\mathbf{y}|\mathbf{c})$, since recovering \mathbf{x} given \mathbf{c} implies recovering $\mathbf{B}^{-1}\mathbf{y}$ given \mathbf{c} and vice versa due to $\mathbf{c} = \mathbf{x} - \mathbf{y}$. Now, suppose we had an upper bound of COL $(\mathbf{B}^{-1}\mathbf{y}|\mathbf{c}) \leq p^{-1} \cdot 2^{-\omega(\log \kappa)}$, or equivalently $\tilde{\mathbf{H}}_2(\mathbf{B}^{-1}\mathbf{y}|\mathbf{c}) \geq \log_2 p + \omega(\log \kappa)$, when we sample $\mathbf{x} \leftarrow \mathcal{X}$ and calculate $(\mathbf{c}, a) \leftarrow \text{Sketch}(pp_{\text{LS}}, \mathbf{x})$, where $\mathbf{c} = \mathbf{x} - \mathbf{y} = \mathbf{x} - g_{\mathcal{L}}(\mathbf{x})$. Then, the leftover hash lemma of [7], formally recalled in App. A, guarantees that the proxy key $a = \text{UH}(\mathbf{B}^{-1}\mathbf{y}) \in \mathbb{Z}_p$ is statistically close to a uniformly random element even given \mathbf{c} , and thus the standard DL assumption implies the DLsketch assumption.

Putting things together, if $\mathsf{ConFMR} \lessapprox 2^{-(2\kappa + \omega(\log \hat{\kappa}))}$, then standard DL implies the $\mathsf{DL}^\mathsf{sketch}$ assumption since $p \approx 2\kappa$. However, since typically $\kappa \geq 80$, this condition on ConFMR may be somewhat too expensive to assume for fuzzy biometrics. Nevertheless, we believe the above provides us an intuition that the $\mathsf{DL}^\mathsf{sketch}$ assumption is not an esoteric assumption and justifies that ConFMR is the right quantity to care about.

(2) DL sketch is hard in the generic group model. The generic group model [31] is an idealized model of computation for a cyclic group, where algorithms do not use the representation (or, the encoding) of the group elements, other than testing the equality of group elements. When a new computational problem related to a cyclic group is introduced, this model is typically used to reason about its hardness. Specifically, if some computational problem is proved to be hard for PPT adversaries in the generic group model, then it formally guarantees that one cannot solve the problem efficiently as long as one is performing only group operations. To solve it efficiently, one must rely on a weakness of a particular group. Thus, the hardness of a computational problem in the generic group model serves as a strong evidence that if we use a cyclic group where no such weakness is known (e.g. a group over an elliptic curve).

Based on existing works, we can observe that if $\mathsf{ConFMR} \approx 2^{-\kappa}$ (or even $2^{-\omega(\log \kappa)} = \kappa^{-\omega(1)}$), then there is no PPT algorithm that can break the $\mathsf{DL}^\mathsf{sketch}$ assumption in the generic group model. Specifically, it is a well-known fact (and formally shown in [9, Lemma 6]) that a universal hash family is a good "strong randomness condenser" and preserves essentially all the (conditional) collision entropy of the input $\mathbf{B}^{-1}\mathbf{y}$ of UH to its output $a = \mathsf{UH}(\mathbf{B}^{-1}\mathbf{y})$. That is, we have $\mathsf{COL}(a|\mathsf{UH},\mathbf{c}) \approx \mathsf{COL}(\mathbf{B}^{-1}\mathbf{y}|\mathbf{c}) \leq \mathsf{ConFMR}$. Moreover, [3] considers a stronger variant of the decisional Diffie-Hellman problem

where the exponents are not uniformly distributed but of superlogarithmic min-entropy $\omega(\log \kappa)$, and showed that this problem is hard for any PPT adversary in the generic group model. This directly implies the hardness of the DL problem in which the exponent of a problem instance is chosen from a distribution with min-entropy $\omega(\log \kappa)$ in the generic group model. Finally, min-entropy and collision entropy are linearly related. Hence, taking average over the choice of $\mathbf{c} = \mathbf{x} - \mathbf{y}$, we can conclude that the DL sketch problem is hard to solve for any PPT adversary in the generic group model if ConFMR = $2^{-\omega(\log \kappa)}$.

6 EXPERIMENTAL METHOD FOR ESTIMATING BIOMETRIC ENTROPY

The final and most important content to cover is the question of whether we can use real-world biometrics to realize efficient and secure fuzzy signatures. For instance, it is clear that if everybody had similar biometrics, then there is no way to realize a secure fuzzy signature since everybody can impersonate each other. However, in reality, everyone possesses different biometrics. Therefore, if we used all of our personal biometrics, then fuzzy signatures should intuitively be secure since it would be extremely hard to impersonate someone. But obviously, collecting vast amount of biometrics from a user will make the signing procedure very expensive and drastically decrease user experience. The question is then, how much entropy does a specific biometric have, and can it be used to securely and efficiently instantiate fuzzy signatures?

This section can be divided into two parts: we first provide an easy-to-state sufficient condition for "fuzzy signature compatible" biometrics, and then we establish an experimental method to show that a given biometric satisfies this condition.

6.1 Preprocessing Biometrics

Before getting into the main content of this section, we first clarify how fuzzy biometrics are handled in more detail. That is, given, say a raw image of a fingerprint, what is the corresponding fuzzy biometric *x* that we have been abstractly using throughout the paper. As with any real-world data, we preprocess (e.g., conduct feature extraction on) raw biometric data obtained via some measurement and represent them in a meaningful way. This preprocessed data is in fact what we have been calling "fuzzy biometrics x" throughout the paper. A pictorial explanation is provided in the bottom of Fig. 9. The method of preprocessing raw biometric data depends on the concrete type of biometrics being used. We provide a concrete example in Sec. 7.1, where we conduct experiments using real-world finger-veins. In the following, when we mention fuzzy biometrics, we always assume the preprocessed version. Moreover, the distribution X of fuzzy biometrics is the distribution induced by preprocessing a randomly sampled raw biometrics.

6.2 Preparation

What is required from fuzzy biometrics? As we have seen in Secs. 4.2 and 5.4, the concrete values of FNMR and ConFMR of the fuzzy biometrics dictate the applicability to fuzzy signatures. Recall the former and latter correspond to the correctness and security of fuzzy signatures, respectively:

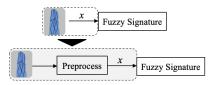


Figure 9: The above (resp. bottom) depicts a simplified (resp. realistic) version of how we handle biometrics.

False Non-Matching Rate (FNMR): Informally, this was the probability that two honestly generated fuzzy biometrics x and x' from the same user are "far". Formally, FNMR := $\Pr[x \leftarrow \mathcal{X}, e \leftarrow \Phi : x + e \notin AR(x)]$.

Conditional False Matching Rate (ConFMR): Informally, this was the collision probability of fuzzy biometrics conditioned on the sketch being identical. Formally, ConFMR := $\Pr[x, x' \leftarrow X, (c, a) \leftarrow \text{Sketch}(x), (\widetilde{c}, \widetilde{a}) \leftarrow \text{Sketch}(x') : x' \in \text{AR}(x) | c = \widetilde{c}]$, where recall Sketch is a deterministic algorithm. In particular, the probability is only over the randomness used to sample x and x'. ¹⁰

Observe that the values of FNMR and ConFMR are determined uniquely by the following factors: the distribution X of fuzzy biometrics, the definitions of the linear sketch, and the acceptance region AR used by the linear sketch. Furthermore, observe that X is implicitly defined by the concrete type of biometrics being used, and the linear sketch only depends on the definition of AR (or equivalently to the lattice as explained in Sec. 5). Therefore, AR is the only parametric term that we can experimentally tune that would affect the values of FNMR and ConFMR. Namely, the choice of AR, which roughly is a metric for deciding whether two fuzzy biometrics x and x' are "similar", is the main term that determines FNMR and ConFMR. As a rule of thumb, we like to define AR to be efficiently computable and to reflect the actual closeness metric of the underlying fuzzy biometric. For instance, if the closeness is measured by the Euclidean metric, the hexagon AR may be better than the square AR as in Fig. 10. (See also Sec. 5.3).

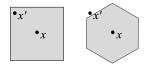


Figure 10: The gray area depicts the acceptance region AR(x) of some fuzzy biometric x. Although x and x' are the same, they may be considered to be close (left) or far (right) depending on AR.

In real-world applications of fuzzy signature, we can typically tolerate correctness error of at most 5% and security level of at least 112-bits. We can tolerate the correctness to be much larger than the security level since we can simply retry till signing succeeds.

To summarize thus far, once we fix a (set of) biometrics, e.g., iris, fingerprint, and finger-vein, used by the fuzzy signature and a description of the linear sketch scheme, the remaining issue is to define an appropriate acceptance region AR and show that the fuzzy

biometric provides FNMR $\lesssim 5\% (\approx 2^{-4.32})$ and ConFMR $\lesssim 2^{-112.11}$ In the following, we show how to experimentally estimate the values of FNMR and ConFMR for a given definition of AR.

What kind of fuzzy biometrics is required for the experiment? For the experiments, we assume a natural type of biometric dataset to be provided: $S = \{x_j^{(i)}\}_{(i,j) \in [N] \times [0:\ell]}$, where $x_j^{(i)}$ is the j-th fuzzy biometric of the i-th user. ¹² That is, S contains $(\ell+1)$ fuzzy biometrics from N users. Such a dataset can be collected in practice by scanning each user i's biometrics $(\ell+1)$ -times. Looking ahead, $x_0^{(i)}$ is a special biometric scanned at the enrollment phase (i.e., generation of the verification key) and $\{x_j^{(i)}\}_{j \in [\ell]}$ are biometrics scanned during signing. Finally, denote $\bar{S} := \{x_j^{(i)}\}_{(i,j) \in [N] \times [\ell]}$ and $\bar{S}^{(i)} := \{x_j^{(i)}\}_{j \in [\ell]}$.

Note that we can always define AR such that $x_j^{(i)} \in AR(x_0^{(i)})$ for all $j \in [\ell]$ and $x_{j'}^{(i')} \notin AR(x_0^{(i)})$ for all $i' \neq i$ and $j \in [\ell]$, i.e., a perfect definition of AR for the specific dataset S. However, it is clear that such an AR is overfitting to the particular dataset S and will not generalize to unseen fuzzy biometrics S. Moreover, since typically, such an AR cannot be computed efficiently we will not be able to efficiently construct an associating linear sketch scheme or perform the experiments explained below. Therefore, in practice, we use natural definitions of AR as those explained in Sec. 5.

6.3 Estimating FNMR of Biometrics

We first estimate FNMR by $\widetilde{\text{FNMR}}$. Given a dataset S of the above type, we empirically calculate $\widetilde{\text{FNMR}}$ as follows:

$$\widetilde{\mathsf{FNMR}} := \frac{\sum_{i \in [N]} \left| \left\{ x \in \bar{\mathcal{S}}^{(i)} \mid x \notin \mathsf{AR}(x_0^{(i)}) \right\} \right|}{\sum_{i \in [N]} \left| \bar{\mathcal{S}}^{(i)} \right|}, \tag{3}$$

where we assume AR is efficiently computable. It is easy to see that the numerator counts all the fuzzy biometric of each user that does not lie inside the acceptance region $AR(x_0^{(i)})$.

6.4 Estimating ConFMR of Biometrics

Difficulty of estimation. We next estimate ConFMR by ConFMR. Computing ConFMR experimentally turns out to be much harder compared to computing FNMR. The main reason is the value of ConFMR that we wish to evaluate is much smaller than FNMR; while we only needed to show that FNMR is smaller than 5%, we need to show that ConFMR is smaller than 2^{-112} to be cryptographically useful. In fact, even if we waived the condition c=c', it is still non-trivial to estimate ConFMR, which is by definition FMR, since the event we are trying to check happens with probability only 2^{-112} . According to the rule of three [13], more than $3 \cdot 2^{112}$ independent impostor biometrics (i.e., pairs of x, x' from different users such that $x' \in AR(x)$) are required in the dataset S to conclude that FMR is smaller than 2^{-112} with 95% confidence. However, collecting such

 $^{^{10}\}mbox{For simplicity,}$ we omit the randomness of the public parameter $\mbox{pp}_{\mbox{\scriptsize LS}}.$

¹¹Note that we need the additional condition that a linear sketch scheme with respect to AR is efficiently constructible. We intentionally keep this requirement implicit to make the presentation simple.

¹²For a non-abstract treatment of fuzzy biometric, see Secs. 6.1 and 7.

S is highly impractical. This is in sharp contrast to FNMR where we only needed to assume that the dataset S contains more than $3 \cdot (1/5\%) \approx 3 \cdot 2^{4.32}$ pairs of biometrics x, x' from the *same* user such that $x' \in AR(x)$ to get a meaningful estimate. We note that estimating FMR, let alone ConFMR, is generally a difficult problem in biometrics due to the difficulty in collecting sufficient data, e.g., [5, 6].

Our approach. We divide the problem of estimating ConFMR into two subproblems as follows:

- (1) First, evaluate $\widehat{\mathsf{FMR}}$. Namely, ignore the condition c = c' on the sketch in ConFMR and simply estimate FMR, where $\mathsf{FMR} := \Pr[x, x' \leftarrow \mathcal{X} : x' \in \mathsf{AR}(x)]$.
- (2) Then, show that FMR and the value of sketch are uncorrelated. Namely, experimentally show that ConFMR can be approximated by FMR.

By individually solving the two subproblems, we eventually estimate the value $\widetilde{\mathsf{ConFMR}}$ by $\widetilde{\mathsf{FMR}}$. The details of the solution to the individual subproblems follow.

Subproblem item 1. As mentioned before, the value FMR is typically too small to perform a simple estimation as we did for FNMR. To overcome this issue, we borrow techniques from *extreme value analysis* (EVA), a statistical method for evaluating very rare events by using only an "extreme" subset of the dataset *S* [4, 30].

We explain how to estimate FMR using EVA below. First, define a continuous function called scaled acceptance region sAR(w,x) defined for all $x \in X$ and w > 0 such that sAR(1,x) := AR(x) and sAR(w,x) is an isotropic scaling of the original set AR(x) by a factor w. A pictorial example is provided in Fig. 11. Notice that although sAR(1,x) does not include many points from the dataset \bar{S} , we can increase them by enlarging w and considering a larger set sAR(w,x). Also, define the scaled false matching rate function $sFMR(w) := Pr[x,x' \leftarrow X : x' \in sAR(w,x)]$, where we have sFMR(1) := FMR by definition. In the following, we estimate the probability distribution sFMR(w), denoted as sFMR(w), and then indirectly estimate the desired sAR(w) by plugging in sR(w) into sFMR(w). Note that this is different from how we were able to directly estimate sR(w) through the dataset sR(w).

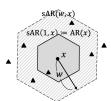


Figure 11: The bold gray area sAR(1,x) is the original set AR(x). The triangles are the fuzzy biometrics in \overline{S} that are different from x. The shaded area sAR(w,x) is the set AR(x) scaled by a factor w.

The core of EVA is how to estimate a probability distribution sFMR(w) in the extremely rare setting $w \approx 1$. The high level idea is as follows. We first hypothesize that sFMR(w) can be explained by a particular class of natural probability density function when w is smaller than some appropriately chosen w^* . For instance, in our case, the class we consider is the family of power distributions

 $\mathcal{F}_{pow} = \{aw^b\}_{a,b>0}$. ¹⁴ So as not to interrupt the explanation of EVA, we provide rational behind the choice in Rem. 1. Now, since sFMR(w^*) = $\int_0^{w^*} aw^b dw$ for some unknown values of a and b, we first estimate $\widetilde{SFMR}(w^*)$ from the dataset S, and then further estimate a and b via the maximal likelihood analysis (MLA) [2]. Here, notice we can properly estimate $sFMR(w^*)$ from the dataset S for an appropriate value of w^* since enough points in the dataset S will lie in the region sAR(w^* , x) for large enough w^* (see Fig. 11). Finally, once a and b are computed via the MLA, we obtain $\widetilde{\mathsf{sFMR}}(1)$ by computing $\int_0^1 aw^b dw$. Note that the main idea behind EVA is to only use an appropriately chosen small w^* so that we can focus on estimating the range where sFMR(w) has extremely small values, rather than estimating the entire function sFMR(w). Specifically, if we use a too large w^* , we may be able to estimate sFMR(w) well in its entirety, however, it will not produce good estimates when conditioning on sFMR(w) with small values. The appropriate choice of w^* is dataset dependent and we discuss this in Rem. 1.

We now provide a more formal description of the above procedure. First, consider the function k(w) defined as

$$k(w) = \sum_{i \in [N]} \left| \left\{ x \in \bar{S} \backslash \bar{S}^{(i)} \mid x \in \mathsf{sAR}(w, x_0^{(i)}) \right\} \right|.$$

Since the dataset S is discrete, we can efficiently compute a sequence of positive reals $w_1 < w_2 < \cdots$, where each w_n is the smallest w satisfying $k(w_n) = n$. We then pick an appropriate $w^* \in \{w_n\}_n$ as explained in Rem. 1, and denote $k^* := k(w^*)$, that is, $w^* = w_{k^*}$. Also set $k_{\max} = \lim_{w \to \infty} k(w)$, where by definition k_{\max} is the number of total impostor biometrics. We estimate the value of $\widehat{\mathsf{sFMR}}(w^*)$ by k^*/k_{\max} . Then, by the hypothesis that the probability density function $\mathsf{sFMR}(w)$ for small $w \le w^*$ follows $f(w) = aw^b$ for some positive reals a and b, we have

$$\frac{k^*}{k_{\max}} = \widetilde{\mathsf{sFMR}}(w^*) \approx \int_0^{w^*} f(w) dw = \frac{a}{b+1} w^{*(b+1)}.$$

Solving the above for a and plugging it into the likelihood function [2], we obtain the following.

$$L(b) = \prod_{n=1}^{k^*} f(w_n) = a^{k^*} \prod_{n=1}^{k^*} w_n^b = \left(\frac{k^*(b+1)}{k_{\max} w^{*(b+1)}}\right)^{k^*} \cdot \prod_{n=1}^{k^*} w_n^b.$$

Taking the logarithm of L(b), we can show it is maximized when $b=k^*/(k^*\ln w^*-\sum_{n=1}^{k^*}\ln w_n)-1$. Setting r=b+1 and combining everything, we conclude that $\widehat{\mathsf{sFMR}}(w)=\frac{k^*}{k_{\max}\cdot w^*r}w^r$. Finally, plugging in w=1, the desired estimate for $\mathsf{sFMR}(1)=\mathsf{FMR}$ is

$$\widetilde{\mathsf{sFMR}}(1) = \frac{k^*}{k_{\max} \cdot w^{*r}}.$$
 (4)

REMARK 1 (CHOICE OF w^* AND \mathcal{F}_{pow}). When using EVA, the particular choice of w^* is data specific, and we typically check whether the choice was reasonable by plotting the estimated function (see Sec. 7.1 for a concrete example). Noticing that w^* and k^* are in one-to-one relation, we can choose k^* instead. The concrete choice of k^* may be data specific but they are typically small values ranging from, say 0.1% to 5%. Put differently, we use only 0.1% to 5% of k_{max} (i.e., the number of

 $^{^{13}} Looking$ ahead, in our experiment, we consider settings where we only need to show 2^{-28} since we use 4 independent biometrics. However, this is still difficult to collect in practice.

¹⁴Note that the class \mathcal{F}_{pow} is not itself a probability density function. We only assume that the probability density function of sFMR(w) for the narrow range $w \in [0, w^*]$ can be explained by \mathcal{F}_{pow} .

impostor biometrics in the dataset S) to estimate the extremely rare events. Moreover, we hypothesize that the probability density function of sFMR(w) for very small values of w is contained in \mathcal{F}_{pow} by making a natural assumption that the local probability distribution around a fuzzy biometrics x is smooth. That is, we assume that for any fuzzy biometrics x, any x' in the vicinity of x occurs with equal probability. Let g(x) be the distribution of x, i.e., X. Then, when w is small, for any x we can approximate sFMR(w) = $\int_{x} \int_{x' \in sAR(w,x)} g(x') dx' dx \approx \int_{x} g(x) \int_{x' \in sAR(w,x)} dx' dx \propto \int_{x} g(x) w'' dx = w''$, where r is the size of the dimension the fuzzy biometric lies in and we used the fact that $f(x) \approx f(x')$. Finally, by taking the derivative of w', we see that the probability density function of sFMR(w) is included in \mathcal{F}_{pow} .

Subproblem item 2. As explained before, directly estimating ConFMR is difficult since the sketch being identical is an extremely rare event and no practical dataset S will contain such samples. Therefore, we instead provide an empirical evidence that ConFMR can be approximated by FMR, and indirectly estimate the value of ConFMR by $\overline{\text{SFMR}}(1)$ obtained above.

Recall Sketch is a deterministic function. Let $q_c(x)$ be the function that ignores the proxy key a and simply outputs the sketch c of $(c,a) \leftarrow \operatorname{Sketch}(x)$. Then, we can rewrite $\operatorname{ConFMR} = \operatorname{Pr}_{x,x' \leftarrow X}[x' \in \operatorname{AR}(x) \mid q_c(x) = q_c(x')]$. Assume the space the sketch c lies in is endowed with the Euclidean metric (which holds true for all known linear sketch scheme). Then, for any $\ell \geq 0$, consider a variant of ConFMR defined as

$$\ell\text{-ConFMR} := \Pr_{x,x' \leftarrow X} [x' \in \mathsf{AR}(x) \mid \mathsf{dist}(q_c(x), q_c(x')) = \ell],$$

where $\operatorname{dist}(z,z'):=\|z-z'\|_2$. If we can show that ℓ is uncorrelated to the value of ℓ -ConFMR for all $\ell \geq 0$, then we can ignore the sketch condition in ConFMR and conclude that ConFMR \approx FMR = $\Pr_{x,x'\leftarrow\mathcal{X}}[x'\in \mathsf{AR}(x)]$. However, unfortunately, since the condition on ℓ -ConFMR is still a very rare event, which we cannot expect to have in our dataset, we still cannot empirically estimate ℓ -ConFMR. To this end, we further relax the condition in ℓ -ConFMR. For any large enough integer M, consider a sequence of reals $0=\ell_0<\dots<\ell_M$ such that $\Pr_{x,x'\leftarrow\mathcal{X}}[\operatorname{dist}(q_c(x),q_c(x'))\in[\ell_{t-1},\ell_t)]=1/M$ for all $t\in[M]$. Then, for each $t\in[M]$, we consider the following alternative variant of ConFMR:

$$\mathsf{ConFMR}_t := \Pr_{x,x' \leftarrow X} \big[x' \in \mathsf{AR}(x) \mid \mathsf{dist}(q_c(x), q_c(x')) \in \big[\ell_{t-1}, \ell_t\big) \big].$$

Due to how the way we partition the ℓ_t 's, ConFMR $_t$ is an approximation of $\frac{(\ell_{t-1}+\ell_t)}{2}$ -ConFMR. Hence, our goal now is to show that for all $t \in [M]$, the value of ℓ_t is uncorrelated to the value of ConFMR $_t$, which in particular approximately establishes that any ℓ is uncorrelated to ℓ -ConFMR. Concretely, we will perform a hypothesis test using t-statistics on the pair $(\ell_t, \text{ConFMR}_t)$ to conclude that ConFMR $_t$ is not significantly correlated with ℓ_t . We refer the standard explanation of statistical t-test to textbooks such as [12].

To perform the statistical t-test, we first prepare the values of ℓ_t and ConFMR_t for all $t \in [M]$. Since we cannot exactly compute them, we estimate them, denoted as $\tilde{\ell}_t$ and $\widehat{\mathsf{ConFMR}}_t$. Estimating $\tilde{\ell}_t$ is simple; we compute $\mathsf{dist}(q_c(x_0^{(i)}), q_c(x'))$ for all $i \in [N]$ and $x' \in \bar{S} \backslash \bar{S}^{(i)}$ and sort them. That is, we compute the distance of the sketches of all impostor pairs in S. Let the obtained distances

be $L_1 \leq \cdots \leq L_{k_{\max}}$, where recall k_{\max} was the number of total impostor pairs in S. Then, we set $\tilde{\ell}_t = L_{\lfloor t \cdot k_{\max}/M \rfloor}$ for $t \in [M]$. To estimate ConFMR $_t$, we use the same method used to estimate FMR to solve subproblem item 1. Namely, we use EVA to estimate ConFMR $_t$ by parameterizing the acceptance region AR. The way the estimation proceeds is exactly the same as before except that we condition on the subset of the dataset S so that the distance of the sketches are within $[\tilde{\ell}_{t-1}, \tilde{\ell}_t)$.

Finally, after obtaining the samples $\{(\tilde{\ell}_t, \mathsf{ConFMR}_t)\}_{t\in[M]}$, we perform the statistical t-test. Below, denote $y_t := \log(\widehat{\mathsf{ConFMR}}_t)$ for $t \in [M]$. We perform a hypothesis test against the samples $\{(\tilde{\ell}_t, y_t)\}_{t\in[M]}$, where the null hypothesis H_0 is that the variables are uncorrelated. To this end, we first compute the sample correlation coefficient r as

$$r = \frac{\sum_{t=1}^{M} (\tilde{\ell}_{t} - \bar{\ell}) (y_{t} - \bar{y})}{\sqrt{\sum_{t=1}^{M} (\tilde{\ell}_{t} - \bar{\ell})^{2}} \cdot \sqrt{\sum_{t=1}^{M} (y_{t} - \bar{y})^{2}}},$$
 (5)

where $\tilde{\ell}$ and \bar{y} are the average of the samples. In case ℓ and y are uncorrelated, then the value $t = \frac{r\sqrt{M-2}}{\sqrt{1-r^2}}$ follows the t-distribution with M-2 degree of freedom. Therefore, we compute the p-value from t and conclude that the null hypothesis H_0 is not rejected at the 0.05 significance level if

$$p \ge 0.05. \tag{6}$$

Hence, if $p \ge 0.05$, then we conclude with high confidence that the value of ℓ -ConFMR is uncorrelated with the value of ℓ , and in particular, approximate ConFMR \approx FMR \approx SFMR(1).

7 EFFICIENCY ANALYSIS OF OUR FUZZY SIGNATURE

In this section, we combine all the tools we developed thus far to show that we can instantiate fuzzy signatures efficiently and securely using real-world biometrics. In Sec. 7.1, we first conduct experiments using the statistical methods presented in Sec. 6 with real-world finger-vein biometrics, and conclude that finger-vein biometric from a single hand is sufficient for fuzzy signature. Then, in Sec. 7.2, we instantiate our fuzzy signature with a concrete set of parameters and provide efficiency analysis of our proposed scheme.

7.1 Estimating Quality of Real-World Finger-Vein Biometrics

We use real-world finger-veins (see Fig. 12) to show that 4 finger-vein scans from a single hand is sufficient to instantiate fuzzy signature. To this end, we provide an appropriate definition of acceptance region AR and provide experimental results using the methods presented in Secs. 6.3 and 6.4 to conclude FNMR $\lessapprox 5\%$ and ConFMR $\lessapprox 2^{-112}$, respectively.

Figure 12: Example of an extracted finger-vein image. The image is taken from [22, Figure 4].



<u>Description of preprocessing and dataset S.</u> There are several publicly available finger-vein datasets such as SDUMLA-HMT [37]

 $^{^{15}}$ We note that r may be smaller than the concrete dimension n of the fuzzy biometrics obtained through some feature extraction.

and Hong Kong Polytechnic University Finger Image Database Version 1.0 [20]. However, in this work, we use the dataset used by [36] as they contain the largest number of users and finger-vein images (roughly 3 to 5 times more).

The finger-vein database contains 505 users where each user provided images of 6 fingers (index, middle, and ring fingers for both hands), and the collection for each finger was repeated 3 times to obtain 3 images (one for the enrollment phase and the other two for the signing phase). We eliminated 36 users that had finger-veins images that were not properly scanned. Moreover, since fingerveins of different fingers from the same user are believed to be independently distributed (a standard assumption used in many prior works [23-25, 34]), we can alternatively view the database as containing $(505 - 36) \times 6 = 2814$ users each providing 3 images of a single finger-vein.

We preprocess these raw finger-vein images into data compatible with our linear sketch (see Sec. 5.4.) We perform feature extraction on the finger-vein and represent it as an *n*-dimensional vector where we experimented with n = 200, 300, 400. For each n, we first randomly selected 1410 users from the 2814 users and performed principal component analysis [2] to extract the *n*-dimensional subspace that best explains the data. Then, we projected the remaining 1404 users' finger-vein onto the *n*-dimensional subspace and prepared the dataset $S = \{x_j^{(i)}\}_{(i,j) \in [1404] \times [0:2]}$. Each n-dimensional vector $x_i^{(i)}$ is represented as a 32-bit float.

Type of AR. We consider a regular hexagon as the acceptance region AR since we use the triangular lattice to instantiate the linear sketch scheme (see Fig. 6 and Sec. 5.3). Denote *d* as the basis length of the triangular lattice (see App. C). Then, since the triangular lattice uniquely defines AR, the value of *d* indirectly parameterizes AR; a larger d results in a larger region for AR. Therefore, given the dataset S, we find the value d that provides us with an AR that satisfies the conditions FNMR $\leq 5\%$ and ConFMR $\leq 2^{-112}$. Note the concrete value of d has no significant meaning as its length is relative to the scaling of the concrete fuzzy biometrics.

Estimating FNMR and ConFMR. We use 4 finger-veins for the fuzzy signature¹⁶ and since each finger-vein is assumed to be distributed independently, we empirically evaluate whether the following holds for each n = 200, 300, 400:

- $\widetilde{\text{FNMR}} \le 1 (1 5\%)^{1/4} \approx 2^{-6.29} \text{ (see Eq. (3))}$ $\widetilde{\text{ConFMR}} \le (2^{-112})^{1/4} = 2^{-28} \text{ (see Eq. (4))}$
- p-value is larger than 0.05 (see Eq. (6))

Here, the first item follows from the fact that we need all 4 fingerveins to be correct to obtain a total of 5% of false non-matching rate. Moreover, the last requirement is to check the validity of our estimation method in Sec. 6.4. Recall that if the p-value is larger than 0.05, then we conclude that ConFMR can be estimated by FMR.

The following Sec. 7.1 summarizes our experimental result. For better readability we present the values of FNMR and ConFMR where 4 finger-veins are simultaneously used, denoted as FNMR^4 and ConFMR^4 . For each dimension n, we varied the basis length d (i.e., acceptance region AR) to see its effect. For each dimension n, we chose three

values for d by targeting $\widehat{ConFMR}^4 = 2^{-80}, 2^{-128}$ and $\widehat{FNMR}^4 = 5\%$, respectively. Although 112-bits is the recommended security level for fuzzy signatures, we also benchmarked 80-bits of security since 80-bits may suffice in adversarially restricted scenarios, e.g., the system blocks the account after a few false attempts at signing. We also included the correlation coefficient r of the t-test (see (Eq. (5))) to show that their absolute values are all below 0.2.

Table 2: n denotes the dimension of fuzzy biometrics, r is the correlation coefficient of the t-test, and d denotes the basis length of the triangular lattice.

n	FNMR ⁴	ConFMR ⁴	<i>p</i> -value	r	d
200	2.4%	2^{-80}	0.35	0.095	43.4
200	5%	$2^{-106.6}$	0.27	-0.111	39.6
200	9.7%	2^{-128}	0.15	0.146	36.8
300	1.4%	2^{-80}	0.57	-0.057	44.6
300	5%	$2^{-113.0}$	0.89	-0.015	40.2
300	7.6%	2^{-128}	0.78	0.028	38.4
400	1.4%	2^{-80}	0.50	-0.068	44.8
400	5%	$2^{-113.6}$	0.65	-0.046	40.3
400	8.0%	2^{-128}	0.88	-0.015	38.6

The entries in bold-fonts in Sec. 7.1 indicate those satisfying either ConFMR⁴ $\leq 2^{-112}$ or FNMR⁴ $\leq 5\%$. When the dimension of the feature vectors of the finger-vein is n = 300 (resp. n = 400) and when the basis length is d = 40.2 (resp. d = 40.3), both conditions on ConFMR4 and FNMR4 are satisfied. Therefore, our result indicates that 4 finger-veins are sufficient to provide the required properties to instantiate fuzzy signatures by taking those appropriate choices of *n* and *d*. Since a larger dimension *n* leads to a less efficient linear sketch scheme, taking n = 300 suffices. In addition, our experimental results also confirm the relationship between the size of AR and the tradeoff between ConFMR⁴ and FNMR⁴. Observe that decreasing the size of AR (i.e., smaller d) has the effect of lowering ConFMR⁴ while increasing FNMR4 as expected; a smaller AR makes it harder to impersonate while it also makes it more sensitive to measurement error. We note that although the false non-matching rate (FNMR⁴) is typically set below 5 % in practice, we can tolerate a higher value of correctness error by allowing the signer to repeat until it succeeds. Therefore, in case we require a higher level of security such as 128-bits, then we can achieve this by increasing FNMR⁴.

To see more closely the effect of varying the dimension n, we plot the detection error tradeoff (DET) curve [15, Sec. 4.7]. The case for n = 200 and 300 is provided in Fig. 13. We refer the case for n = 400 to App. E.1 since it is similar to n = 300. For each dimension *n*, the DET curve is plotted by varying the size of *d* (hence AR). Any region that lies above the DET curve is realizable. For instance, since the coordinate indicating $(\overline{FNMR^4}, \overline{ConFMR^4}) = (2^{-112}, 5\%)$ (denoted as a red star in Fig. 13) is below the DET curve for n = 200, this means that there is no d that satisfies the condition of FNMR⁴ and ConFMR⁴ when representing the finger-vein by a feature vector of only dimension n = 200. In contrast, for n = 300 and 400, it can be checked that the there exists some choice of d such that the

 $^{^{16}\}mathrm{Although}$ our linear sketch is defined for a single biometric source, it is clear that they generalize to multiple independent biometric sources. For completeness, details are provided in App. D.

conditions are satisfied since the red star is above their respective DET curves.

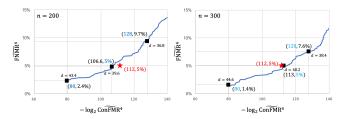


Figure 13: DET Curve for n = 200 (left) and 300 (right).

Finally, we provide graphical evidence on the validity of our estimation of FMR and $\widetilde{\text{ConFMR}}_t$ for $t \in [M]$; such method is one of the standard ways of assessing the quality of EVA [4, 30]. To perform EVA to estimate \widetilde{FMR} , we set $k^* = 0.1\% \times k_{max}$, where recall k_{max} is the number of total impostor pairs which is equal to 3,925,584 for our dataset (see Rem. 1). We also set M=100to define the M-variant of $\{\operatorname{ConFMR}_t\}_{t\in[M]}$ and perform EVA to estimate each $\widetilde{\mathsf{ConFMR}}_t$ by setting $k^* = 0.5\% \times k_{\max}$. The following Fig. 14 illustrates the validity of our estimation for FMR when the dimension n = 300 and ConFMR_t for t = 50. It can be visibly checked looking at the gray region that the estimation (in red line) aligns with the values of w that we were able to measure with our database (in blue line). Hence, EVA allows us to conclude that the extremely small values that we were not able to measure with our dataset can be approximated with our estimation. Additional experiments for other parameters are provided in App. E.2.

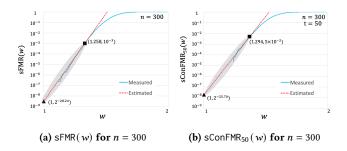


Figure 14: The blue line indicates the measured values of SFMR(w) and SCONFMR₅₀(w) w.r.t to our dataset S. The red line indicates our estimation of the probability distribution of SFMR(w) and SCONFMR₅₀(w) via EVA. The gray region is the region for which EVA provides a reliable estimation. The square plots (w^* , k^*) and the triangle plots (1, X), where X is the estimation for FMR and CONFMR₅₀.

7.2 Efficiency Analysis of Our Fuzzy Signature

We finish with a concrete analysis of our fuzzy signature scheme. We consider the 112 and 128-bit security levels (i.e., ConFMR $\leq 2^{-112}$) and use the settings in Tab. 2 for n=300 to define the fuzzy key setting. Recall from Fig. 5 that the verification key consists of one group element in $\mathbb G$ and a sketch, and the signature consists of two elements in $\mathbb Z_p$ and a sketch. The running time is the sum of the individual runtime of the linear sketch and the Schnorr signature.

Specifically, the only difference from the Schnorr signature is the linear sketch component. Tab. 3 gives the concrete parameters. In

Table 3: Benchmark for our fuzzy signature with n = 300.

Sec.	Signature		Verification		Correct
level	size (byte)	time (ms)	size (byte)	time (ms)	ness err.
112	1256	0.50	1228	1.4	5.0%
128	1264	0.50	1232	1.4	7.6%

more detail, the sketch has size 4n bytes in general, where 4 bytes is used to represent each element by a 32-bit float. Plugging in n = 300, it can be checked that the size of the sketch dominates the signature and verification key size. The run time of Sketch and DiffRec are 0.45ms and 1.3 ms for both security levels¹⁷, run on a machine with Intel(R) Core(TM) i7-8700K CPU at 3.70GHz. Here, the universal hash UH used within our linear sketch scheme (see Fig. 7) simply computes the inner-product with a random n-dimensional vector over a prime field defined by the secret key space \mathbb{Z}_p of the Schnorr signature scheme. We also implement the Schnorr signature at the 112 and 128-bit security levels using elliptic curves with 224 and 256-bit primes, respectively, run on a machine with Intel(R) Core(TM) i7-1065G7 CPU at 1.30GHz. For both security levels, the run times for signing and verification are at most several tens of microseconds, thus at least an order of magnitude smaller than the time taken by the linear sketch scheme.

We note that we can lower the sketch size by a factor of 2 by representing the fuzzy biometrics by 16 bits rather than 32 bits. In this case, the signature size will roughly be twice as small. Here, treating less number of significant digits for the sketch value may affect the correctness (i.e., FNMR) of the scheme, but not its security as formally discussed in [33, Section 8].

Acknowledgement. A part of this work was supported by JST CREST Grant Number JPMJCR19F6.

REFERENCES

- [1] Mihir Bellare and Gregory Neven. 2006. Multi-signatures in the plain public-Key model and a general forking lemma. In ACM CCS 2006, Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati (Eds.). ACM Press, 390–399. https://doi.org/10.1145/1180405.1180453
- [2] Christopher M Bishop. 2006. Pattern recognition and machine learning. springer.
- [3] Nir Bitansky and Ran Canetti. 2010. On Strong Simulation and Composable Point Obfuscation. In CRYPTO 2010 (LNCS, Vol. 6223), Tal Rabin (Ed.). Springer, Heidelberg, 520–537. https://doi.org/10.1007/978-3-642-14623-7_28
- [4] Stuart Coles, Joanna Bawa, Lesley Trenner, and Pat Dorazio. 2001. An introduction to statistical modeling of extreme values. Vol. 208. Springer.
- [5] John Daugman. 2003. The importance of being random: statistical principles of iris recognition. *Pattern recognition* 36, 2 (2003), 279–291.
- [6] John Daugman. 2004. How Iris Recognition Works. IEEE Transactions on Circuits and Systems for Video Technology 14, 1 (2004), 21–30.
- [7] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. 2008. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM journal on computing 38, 1 (2008), 97–139.
- [8] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. 2004. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In EUROCRYPT 2004 (LNCS, Vol. 3027), Christian Cachin and Jan Camenisch (Eds.). Springer, Heidelberg, 523–540. https://doi.org/10.1007/978-3-540-24676-3_31
- [9] Yevgeniy Dodis and Yu Yu. 2013. Overcoming Weak Expectations. In TCC 2013 (LNCS, Vol. 7785), Amit Sahai (Ed.). Springer, Heidelberg, 1–22. https://doi.org/ 10.1007/978-3-642-36594-2 1

 $^{^{17}}$ The only step dependent on the security parameter in our linear sketch scheme is the field size in UH, but its computation takes time that is at least two orders of magnitude smaller than computing $g_{\mathcal{L}_{tri}}$ or $\text{CV}_{\mathcal{L}_{tri}}$, so its effect on run time is negligible.

- [10] Vidhi Doshi. 2018. A security breach in India has left a billion people at risk of identity theft. https://www.washingtonpost.com/news/worldviews/wp/2018/01/04/a-security-breach-in-india-has-left-a-billion-people-at-risk-of-identity-theft/. Accessed: 2020-12-22.
- [11] The European Union Agency for Cybersecurity. 2020. ENISA Threat Landscape 2020 - Data Breach. https://www.enisa.europa.eu/publications/enisa-threatlandscape-2020-data-breach. Accessed: 2020-12-22.
- [12] Jean Dickinson Gibbons and Subhabrata Chakraborti. 2014. Nonparametric Statistical Inference: Revised and Expanded. CRC press.
- [13] James A Hanley and Abby Lippman-Hand. 1983. If nothing goes wrong, is everything all right?: interpreting zero numerators. JAMA 249, 13 (1983), 1743– 1745.
- [14] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. 1999. A Pseudorandom Generator from any One-way Function. SIAM J. Comput. 28, 4 (1999), 1364–1396.
- [15] ISO/IEC 19795:2006 2006. Information technology Biometric performance testing and reporting – Part 1: Principles and framework. Standard. International Organization for Standardization, Geneva, CH.
- [16] ISO/IEC 24745:2011 2011. Information technology Security techniques Biometric information protection. Standard. International Organization for Standardization, Geneva, CH.
- [17] ISO/IEC 30136:2018 2018. Information technology Performance testing of biometric template protection schemes. Standard. International Organization for Standardization, Geneva, CH.
- [18] Yosuke Kaga, Masakazu Fujio, Ken Naganuma, Kenta Takahashi, Takao Murakami, Tetsushi Ohki, and Masakatsu Nishigaki. 2017. A secure and practical signature scheme for blockchain based on biometrics. In *International Conference on Information Security Practice and Experience*. Springer, 877–891.
- [19] Takashi Kawakami and Yusuke Hinata. 2019. Pay with your face: 100m Chinese switch from smartphones. https://asia.nikkei.com/Business/China-tech/Paywith-your-face-100m-Chinese-switch-from-smartphones. Accessed: 2020-1-14.
- [20] Ajay Kumar and Yingbo Zhou. 2011. Human identification using finger images. IEEE Transactions on image processing 21, 4 (2011), 2228–2244.
- [21] Takahiro Matsuda, Kenta Takahashi, Takao Murakami, and Goichiro Hanaoka. 2016. Fuzzy Signatures: Relaxing Requirements and a New Construction. In ACNS 16 (LNCS, Vol. 9696), Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider (Eds.). Springer, Heidelberg, 97–116. https://doi.org/10.1007/978-3-319-39555-5 6
- [22] Naoto Miura, Akio Nagasaka, and Takafumi Miyatake. 2002. Automatic Feature Extraction from non-uniform Finger Vein Image and its Application to Personal Identification.. In MVA. Citeseer, 253–256.
- [23] Takao Murakami, Yosuke Kaga, and Kenta Takahashi. 2016. Information-theoretic performance evaluation of multibiometric fusion under modality selection attacks. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 99, 5 (2016), 929–942.
- [24] Karthik Nandakumar, Anil K Jain, and Arun Ross. 2009. Biometric fusion: Does modeling correlation really matter?. In *International Conference on Biometrics: Theory, Applications, and Systems*. IEEE, 1–6.
- [25] Karthik Nandakumar, Anil K Jain, and Arun Ross. 2009. Fusion in multibiometric identification systems: What about the missing data?. In *International Conference* on *Biometrics*. Springer, 743–752.
- [26] United Nations. 2020. Report of the Secretary-General Roadmap for Digital Cooperation. https://www.un.org/en/content/digital-cooperation-roadmap/ assets/pdf/Roadmap_for_Digital_Cooperation_EN.pdf.
- [27] Government of India. 2019. What is Aadhar. https://uidai.gov.in/my-aadhaar/ about-your-aadhaar.html. Accessed: 2021-1-18.
- [28] David Pointcheval and Jacques Stern. 2000. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology* 13, 3 (June 2000), 361–396. https://doi.org/10.1007/s001450010003
- [29] Claus-Peter Schnorr. 1990. Efficient Identification and Signatures for Smart Cards. In CRYPTO'89 (LNCS, Vol. 435), Gilles Brassard (Ed.). Springer, Heidelberg, 239–252. https://doi.org/10.1007/0-387-34805-0_22
- [30] Michael Schuckers. 2012. Scaling of Biometric False Match Rates Using Extreme Value Theory. (2012). https://www.nist.gov/system/files/documents/2016/11/ 30/345_schuckers_ibpc.pdf International Biometrics Performance Conference (NIST).
- [31] Victor Shoup. 1997. Lower Bounds for Discrete Logarithms and Related Problems. In EUROCRYPT'97 (LNCS, Vol. 1233), Walter Fumy (Ed.). Springer, Heidelberg, 256–266. https://doi.org/10.1007/3-540-69053-0_18
- [32] Kenta Takahashi, Takahiro Matsuda, Takao Murakami, Goichiro Hanaoka, and Masakatsu Nishigaki. 2015. A Signature Scheme with a Fuzzy Private Key. In ACNS 15 (LNCS, Vol. 9092), Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis (Eds.). Springer, Heidelberg, 105–126. https://doi.org/10.1007/978-3-319-28166-7_6
- [33] Kenta Takahashi, Takahiro Matsuda, Takao Murakami, Goichiro Hanaoka, and Masakatsu Nishigaki. 2019. Signature schemes with a fuzzy private key. *International Journal of Information Security* 18 (2019), 581–617.

- [34] Qian Tao and Raymond Veldhuis. 2012. Robust biometric score fusion by naive likelihood ratio via receiver operating characteristics. IEEE transactions on information forensics and security 8, 2 (2012), 305–313.
- [35] VISA. 2017. Goodbye, passwords. Hello, biometrics. https://usa.visa.com/dam/VCOM/global/visa-everywhere/documents/visa-biometrics-payments-study.pdf. Accessed: 2021-1-14.
- [36] Takashio Yanagawa, Satoshi Aoki, and Tetsuji Oyama. 2009. Diversity of human finger vein patterns and its application to personal identification. *Bulletin of informatics and cybernetics* 41 (2009), 1–9.
- [37] Yilong Yin, Lili Liu, and Xiwei Sun. 2011. SDUMLA-HMT: a multimodal biometric database. In Chinese Conference on Biometric Recognition. Springer, 260–268.

A LEFTOVER HASH LEMMA

We recall the leftover hash lemma of [7]. To see the connection with the explanation in Sec. 5.4, we state it using conditional collision probability.

Recall that for a joint distribution (X, \mathcal{Y}) , the (average) conditional collision probability of X given \mathcal{Y} is defined by $COL(X|\mathcal{Y}) = Pr_{(x,y),(x',y') \leftarrow (X,\mathcal{Y})}[x = x'|y = y']$.

Recall also that the statistical distance between two distributions \mathcal{X} and \mathcal{Y} is defined by $\mathrm{SD}(\mathcal{X},\mathcal{Y}):=\frac{1}{2}\sum_z|\mathrm{Pr}[\mathcal{X}=z]-\mathrm{Pr}[\mathcal{Y}=z]|$. It is known that $\mathrm{SD}(\mathcal{X},\mathcal{Y})$ upper-bounds the best (computationally unbounded) adversary's advantage in distinguishing the distribution using a single sample.

LEMMA A.1 (SLIGHTLY ADAPTED FROM [7]). Let $\mathcal{UH} = \{ UH : D \to R \}$ be a family of universal hash functions. Let X and Y be distributions such that (X,Y) forms a joint distribution, and the support of X is contained in D. Then, the statistical distance of the following two distributions is at most $\frac{1}{2}\sqrt{|R|} \cdot COL(X|Y)$:

$$\left\{ \mathsf{UH} \leftarrow \mathcal{UH}; \ (x,y) \leftarrow (X,\mathcal{Y}) : (\mathsf{UH},\mathsf{UH}(x),y) \right\},$$

$$\left\{ \mathsf{UH} \leftarrow \mathcal{UH}; \ (x,y) \leftarrow (X,\mathcal{Y}); \ r \leftarrow R : (\mathsf{UH},r,y) \right\}.$$

In particular, if $COL(X|\mathcal{Y}) \leq |R|^{-1} \cdot 2^{-\omega(\log \kappa)}$, then the statistical distance is $negl(\kappa)$.

Strictly speaking, [7] showed the above lemma using the (average) conditional min-entropy (rather than conditional collision entropy/probabiltiy). However, the above lemma can be easily inferred from the proof of [7, Lemma 2.4] and the fact that the most basic form of the leftover hash lemma [14] (without taking into account the existence of \mathcal{Y}) works with collision entropy.

B OMITTED PROOF OF OUR FUZZY SIGNATURE Π_{FS}^{DL}

B.1 Omitted Proof of Correctness: Thm. 4.1

The complete proof of Thm. 4.1 is provided below. It establishes the correctness of our fuzzy signature Π^{DL}_{ES} .

PROOF. Recall that by the definition of the fuzzy key setting \mathcal{F} , we have $\Pr[x \leftarrow \mathcal{X}; e \leftarrow \Phi : x + e \in \mathsf{AR}(x)] \ge 1 - \epsilon$. Hence, to show correctness, it is sufficient to show that if $x' \in \mathsf{AR}(x)$, then a signature generated using x' is always accepted under a verification key generated using x.

Fix arbitrarily a message M and fuzzy data $x, x' \in X$ such that $x' \in AR(x)$. Let $(c, a) = Sketch(pp_{LS}, x)$ and $(\widetilde{c}, \widetilde{a}) = Sketch(pp_{LS}, x')$. Also, let $pp_{FS} = (\mathcal{G}, pp_{LS}) \leftarrow FS.Setup(1^K, \mathcal{F})$, $vk_{FS} = (h = g^a, c) \leftarrow FS.KeyGen(pp_{FS}, x)$, and $\sigma_{FS} = (\beta, z, \widetilde{c}) \leftarrow FS.Sign(pp_{FS}, x', M)$, where $\beta = H(g^{\widetilde{a}}, g^r, M)$ and $z = \beta \cdot \widetilde{a} + r$.

Now, consider an execution of FS.Vrfy(pp_{FS}, vk_{FS}, M, σ_{FS}). Since $x' \in AR(x)$, the correctness of Π_{LinS} implies $\Delta a = DiffRec(pp_{LS}, c, \tilde{c}) = \tilde{a} - a$. Hence, FS.Vrfy sets $\tilde{h} = h \cdot g^{\Delta a} = g^{a+(a'-a)} = g^{\tilde{a}}$ and $R = g^z \cdot \tilde{h}^{-\beta} = g^{\beta \cdot \tilde{a} + r} \cdot g^{-\tilde{a} \cdot \beta} = g^r$. Hence, $\beta = H(g^{\tilde{a}}, g^r, M) = H(\tilde{h}, R, M)$ holds, and consequently FS.Vrfy outputs \top , as desired.

B.2 Omitted Proof of Security: Thm. 4.3

The complete proof of Thm. 4.3 is provided below. It establishes the security of our fuzzy signature Π^{DL}_{FS} under the DL^{sketch} assumption.

PROOF OVERVIEW. Before diving into the full proof, we provide an overview. The proof is similar to that of Schnorr signature [29]. The main difference is that in our proof, we additionally have to simulate the sketch c without knowledge of the secret fuzzy data x. To this end, we use the *linearity* of the linear sketch (see Def. 3.1) that informally stipulates that given a sketch c where $(c, a) \leftarrow \text{Sketch}(\text{pp}_{\text{LS}}, x)$, there exists an algorithm M_c that simulates a fresh sketch \widetilde{c} for a proxy key \widetilde{a} with knowledge of only $(c, \Delta a := \widetilde{a} - a)$. Specifically, since the DL^{sketch} problem implicitly provides us with an "initial" sketch c of the proxy key (or secret exponent) a, we can easily construct an adversary $\mathcal B$ against the DL^{sketch} problem that simulates the EU-CMA security game to an adversary $\mathcal A$ by running M_c . The full proof follows.

PROOF. Let \mathcal{A} be any PPT adversary against the fuzzy signature scheme Π_{FS}^{DL} that makes at most \mathcal{Q} -signing queries and \mathcal{Q}_H -random oracle queries and breaks the EU-CMA security with probability ϵ . Consider the following sequence of games, where the first game is equivalent to the original EU-CMA game. Let E_i denote the event that \mathcal{A} wins in Game $_i$.

- Game₁: We define Game₁ as the actual game played between the challenger and the adversary \mathcal{A} . By assumption the winning probability of \mathcal{A} in this game is $\Pr[E_1] = \epsilon$. In this game, the public parameter pp_{FS} and the verification key vk_{FS} are generated as follows:

$$\begin{split} & \left[\mathsf{pp} \leftarrow \mathsf{S.Setup}(1^{\kappa}); \; \mathsf{pp}_{\mathsf{LS}} \leftarrow \mathsf{LinS.Setup}(\mathcal{F}, \Lambda); \; \mathsf{pp}_{\mathsf{FS}} \leftarrow (\mathsf{pp}, \mathsf{pp}_{\mathsf{LS}}); \right. \\ & \left. x \leftarrow \mathcal{X}; \; (c, a) \leftarrow \mathsf{Sketch}(\mathsf{pp}_{\mathsf{LS}}, x); \; \mathsf{vk}_{\mathsf{FS}} = (h \leftarrow g^a, c) \right]. \end{split} \tag{7}$$

Furthermore, when \mathcal{A} makes the *i*-th signing query (for $i \in [Q]$) on message M_i , the challenger generates a signature σ_{FS_i} as follows:

$$\begin{split} & \left[e_i \leftarrow \Phi; \; (\widetilde{c}_i, \widetilde{a}_i) \leftarrow \mathsf{Sketch}(\mathsf{pp}_\mathsf{LS}, x + e_i); \; r_i \leftarrow \mathbb{Z}_p; \right. \\ & \left. \beta_i \leftarrow \mathsf{H}(g^{\widetilde{a}_i}, g^{r_i}, \mathsf{M}_i); \; z_i \leftarrow \beta_i \cdot \widetilde{a}_i + r_i; \; \sigma_{\mathsf{FS}\,i} = (\beta_i, z_i, \widetilde{c}_i) \right]. \end{split}$$

Throughout the proof, we call $g^{\tilde{a}_i} (= \tilde{h}_i)$ and \tilde{a}_i as *ephemeral* verification and signing keys, respectively, as it can be seen as an intermediate key used during the signing phase.

- Game₂: In this game, we change how the signing queries are answered by the challenger. Instead of using the ephemeral signing key \tilde{a}_i to create the sketch \tilde{c}_i as in the previous game, the challenger uses the auxiliary algorithm M_c of the linear sketch Π_{LinS} (Def. 3.1) with input c and e_i . Specifically, when \mathcal{A} makes the i-th signing query (for $i \in [Q]$) on message M_i , the challenger generates a signature $\sigma_{\text{FS}i}$ as follows: (where the difference from Game₁ is underlined.)

$$\begin{aligned} & \left[e_i \leftarrow \Phi; \ \underline{(\tilde{c}_i, \Delta a_i)} \leftarrow \mathsf{M}_\mathsf{c}(\mathsf{pp}_\mathsf{LS}, c, e_i); \ \tilde{a}_i \leftarrow a + \Delta a_i; \ r_i \leftarrow \mathbb{Z}_p; \\ & \beta_i \leftarrow \mathsf{H}(q^{\tilde{a}_i}, q^{r_i}, \mathsf{M}_i); \ z_i \leftarrow \beta_i \cdot \tilde{a}_i + r_i; \ ; \ \sigma_\mathsf{FS}_i = (\beta_i, z_i, \widetilde{c}_i) \right]. \end{aligned}$$

By the linearity of the linear sketch scheme Π_{LinS} , the distribution of $(\widetilde{c}_i)_{i \in [Q]}$ generated in Game_1 and Game_2 are identical. Therefore, we have $\mathsf{Pr}[\mathsf{E}_1] = \mathsf{Pr}[\mathsf{E}_2]$.

- Game₃: In this game, we further modify how the signing queries are answered by the challenger. In the previous game, after $(g^{\tilde{a}_i}, g^{r_i}, M_i)$ were set, the challenger checked whether the random oracle H was set on that point. If not, it sampled a random $\beta_i \leftarrow \mathbb{Z}_p$ and set the random oracle as $H(g^{\tilde{a}_i}, g^{r_i}, M_i) := \beta_i$. Otherwise, it outputs the already programmed output. In this game, the challenger will abort the game when the input was already programmed. Since the random oracle is ever programmed on at most $(Q + Q_H)$ inputs, and r_i is randomly sampled from \mathbb{Z}_p , the probability of an abort occurring on any of the signing query can be upper bounded by $Q \cdot (Q + Q_H)/p$. Hence, $|\Pr[E_2] - \Pr[E_3]| \leq Q \cdot (Q + Q_H)/p$.

- Game₄: In this game, we make a final modification on how the signing queries are answered by the challenger. In particular, we alter the signing procedure so that the challenger no longer requires the secret key a to sign; instead it will indirectly use the public key $h = g^a$. Conditioning on an abort not occurring, the challenger performs the following: (where the difference from Game₃ is underlined.)

$$\begin{aligned} \left[e_{i} \leftarrow \Phi; \; (\widetilde{c}_{i}, \Delta a_{i}) \leftarrow \mathsf{M}_{\mathsf{c}}(\mathsf{pp}_{\mathsf{LS}}, c, e_{i}); \; \beta_{i} \leftarrow \{0, 1\}^{2\kappa}; \\ z_{i} \leftarrow \mathbb{Z}_{p}; \; R_{i} \leftarrow g^{z_{i}} \cdot (h \cdot g^{\Delta a_{i}})^{-\beta_{i}}; \; \mathsf{H}(h \cdot g^{\Delta a_{i}}, R_{i}, \mathsf{M}_{i}) \coloneqq \beta_{i}; \\ \sigma_{\mathsf{FS}, i} = (\beta_{i}, z_{i}, \widetilde{c}_{i}) \right]. \quad (8) \end{aligned}$$

The only difference from the previous game is the order of which r_i and z_i are constructed. In the previous game, a uniform random $r_i \leftarrow \mathbb{Z}_p$ was sampled and then z_i was set as $\beta_i \cdot \tilde{a}_i + r_i = \beta_i \cdot (a + \Delta a_i) + r_i$. However, in this game, a uniform random $z_i \leftarrow \mathbb{Z}_p$ is sampled and then r_i is implicitly set to $z_i - \beta_i \cdot (a + \Delta a_i)$. We say "implicitly" since the challenger actually only computes r_i in the exponent, that is, $R_i = g^{r_i}$. Since the joint distribution of (r_i, z_i) is identical in Game₃ and Game₄, we conclude $\Pr[E_3] = \Pr[E_4]$.

Summarizing thus far, we upper bound the advantage of $\mathcal A$ winning the EU-CMA game as follows:

$$\epsilon = \Pr[\mathsf{E}_1] \le \sum_{i=1}^{3} |\Pr[\mathsf{E}_i] - \Pr[\mathsf{E}_{i+1}]| + \Pr[\mathsf{E}_4]$$

$$\le \Pr[\mathsf{E}_4] + \frac{Q \cdot (Q + Q_\mathsf{H})}{p}. \tag{9}$$

Therefore, in order to conclude the proof, it suffices to show that $\epsilon_4 := \Pr[\mathsf{E}_4]$ is negligible. Below, we show that an adversary $\mathcal A$ against Game₄ can be used to construct an adversary $\mathcal B$ against the DL^{sketch} assumption. This is a direct consequence of the forking lemma [1, 28]. The description of $\mathcal B$ follows:

- $\mathcal{B}(\mathcal{G},\operatorname{pp}_{\mathsf{LS}},h,c)$: Given a DL^{sketch} instance, \mathcal{B} simulates the Game4-challenger to \mathcal{A} by appropriately programming the random oracle. Note that \mathcal{B} can answer all queries via Eqs. (7) and (8). If \mathcal{A} outputs a valid forgery $(\mathcal{M}^*,\sigma_{\mathsf{FS}}^*=(\beta^*,z^*,\widetilde{c}^*))$, \mathcal{B} then checks if it ever replied back to \mathcal{A} with β^* to a random oracle query of the form $(\tilde{h},R,\mathcal{M}^*)$. If not, \mathcal{B} aborts. Otherwise, assume \mathcal{A} queried $(\tilde{h},R,\mathcal{M}^*)$ to the random oracle as its I^* -th query, where $I^*\in [Q]$. \mathcal{B} then reruns \mathcal{A} on the same randomness tape and answers the random oracle queries identically to the previous run up until the I^* -th query and with fresh random outputs from the I^* -th query. Then, if \mathcal{A} outputs another valid forgery $(\mathcal{M}'^*,\sigma_{\mathsf{FS}}'^*=(\beta'^*,z'^*,\widetilde{c}'^*))$, \mathcal{B} then checks if $\mathcal{M}'^*=\mathcal{M}^*,\beta'^*\neq\beta^*$, and that β'^* is the output of the

 I^* -th random oracle query. If not $\mathcal B$ aborts. Otherwise, $\mathcal B$ outputs $\frac{z^*-z'^*}{\beta^*-\beta'^*}-\Delta a$ as the solution to the DL^{sketch} problem and terminates, where $\Delta a \leftarrow \mathsf{DiffRec}(\mathsf{pp}_\mathsf{LS}, c, \widetilde{c}^*)$

Let us analyze algorithm \mathcal{B} . It is easy to see that the first run simulates the view of Game₄ perfectly to \mathcal{A} . Therefore, standard argument using the forking lemma [1, 28] tells us that \mathcal{B} outputs something (i.e., will not abort) with probability $\epsilon_4 \cdot (\frac{\epsilon_4}{Q} - \frac{1}{p})$ and runs in time about twice as \mathcal{A} . Next, we show that condition on \mathcal{B} outputting something, it solves the DL sketch problem with probability 1. Observe that since the two runs are identical up till the point \mathcal{A} makes the I^* -th random oracle query, we must have that \mathcal{A} queried (\tilde{h}, R, M^*) to the random oracle as its I^* -th query in the second run as well. Due to validity of the forgery in the two runs, we have $\tilde{h} = h \cdot g^{\Delta a}$, $R = g^{z^*} \cdot \tilde{h}^{-\beta^*}$, and $R = g^{z^{\prime*}} \cdot \tilde{h}^{-\beta^{\prime*}}$. Simple calculation shows that $\mathrm{dlog}_g(h) = \frac{z^* - z^{\prime*}}{\beta^* - \beta^{\prime*}} - \Delta a$. Therefore, \mathcal{B} correctly solves the DL sketch problem.

The above shows

$$\epsilon_4 = \Pr[\mathsf{E}_4] \le \sqrt{Q \cdot \left(\epsilon_{\mathsf{DL}^\mathsf{sketch}}(\kappa) + \frac{1}{p}\right)},$$

where $\epsilon_{\mathsf{DL}^{\mathsf{sketch}}}(\kappa)$ is the maximum advantage of a PPT adversary against the $\mathsf{DL}^{\mathsf{sketch}}$ problem. Combining this with Eq. (9) completes the proof of Thm. 4.3.

C FURTHER DETAILS ON TRIANGULAR LATTICES

We introduce the formal definition of triangular lattices here: Let d be any positive real and let $\mathbf{B}_{tri} = [\mathbf{b}_1, \dots, \mathbf{b}_n]$ be a basis matrix such that

- (1) $\|\mathbf{b}_i\|_2 = d$ for all $i \in [n]$, and
- (2) $\mathbf{b}_i \cdot \mathbf{b}_j = d^2/2$ for all $i, j \in [n]$ with $i \neq j$, where "·" denotes the inner product.

We call $\mathcal{L}_{tri} = \mathcal{L}(B_{tri})$ the triangular lattice (with basis length d). Notice that changing the d has the effect of changing the size of the acceptance region AR. That is, using a larger d results in a larger AR.

A triangular lattice enjoys the property that for any $\mathbf{x} \in \mathbb{R}^n$, we can calculate $\mathrm{CV}_{\mathcal{L}_{\mathrm{tri}}}(\mathbf{x})$ efficiently in terms of the dimension n. Concretely, its computational cost is $O(n^2)$. Since the following description of the closest vector algorithm is invariant to the choice of d, we assume d=1. The details follow.

For simplicity, we use the representation with respect to \mathcal{L}_{tri} for the target vector \mathbf{x} . If an input vector is with respect to the standard basis, then it can be converted to one with the representation with respect to \mathcal{L}_{tri} by multiplying with \mathbf{B}^{-1} . Let $\mathbf{x} = x_1\mathbf{b}_1 + \cdots + x_n\mathbf{b}_n$ (where $x_i \in \mathbb{R}$ for each $i \in [n]$) be the target vector for which we would like to compute the closest vector $\mathbf{y} = \mathsf{CV}_{\mathcal{L}_{tri}}(\mathbf{x})$. For simplicity, we first explain the case $x_i \in [0,1)$ for each $i \in [n]$, and later explain how to extend it to the general case $(x_i \in \mathbb{R})$. In this case, $\mathbf{y} = \mathsf{CV}_{\mathcal{L}_{tri}}(\mathbf{x})$ can be written as $\mathbf{y} = y_1\mathbf{b}_1 + \cdots + y_n\mathbf{b}_n$ with $y_i \in \{0,1\}$ for each $i \in [n]$.

Due to the property of the triangular lattice, for all $i, j \in [n]$, we have the following properties:

$$x_i \le x_j \Rightarrow (y_i, y_j) \in \{(0, 0), (0, 1), (1, 1)\},\$$

$$x_i \ge x_j \Rightarrow (y_i, y_j) \in \{(0, 0), (1, 0), (1, 1)\}.$$

In other words, the magnitude relation among the coordinates $\{x_i\}_{i\in[n]}$ of the target vector and that among the coordinates $\{y_i\}_{i\in[n]}$ of the closest vector are synchronized. Hence, the above two relations can be equivalently written as

$$x_i \le x_j \Rightarrow y_i \le y_j,$$

 $x_i \ge x_j \Rightarrow y_i \ge y_j.$

Using this fact, we consider the sorting of $\{(x_i, y_i)\}_{i \in [n]}$ in ascending order by using $\{x_i\}_{i \in [n]}$ as the sorting key. Let $(x_i^*, y_i^*)_{i \in [n]}$ be the result of the sorting. Then, due to the above relations, we have

$$y_1^* \le y_2^* \le \dots \le y_n^*. \tag{10}$$

Since we are considering the case that $y_i \in \{0, 1\}$ for each $i \in [n]$, the sequence $(y_i^*)_{i \in [n]}$ has the property that there exists an index $k \in \{0, 1, \ldots, n\}$ such that $y_1^* = \cdots = y_k^* = 0$ and $y_{k+1}^* = \cdots = y_n^* = 1$. There are n+1 candidates for k. Hence, by computing the distance between the target vector \mathbf{x} and n+1 vectors satisfying Eq. (10), we can compute the closest vector \mathbf{y} .

The concrete procedure for computing the closest vector $\mathbf{y} = \text{CV}_{\mathcal{L}_{\text{tri}}}(\mathbf{x}) = \sum_{i \in [n]} y_i \mathbf{b}_i$ (with $y_i \in \{0, 1\}$) from a target vector $\mathbf{x} = \sum_{i \in [n]} x_i \mathbf{b}_i$ (with $x_i \in [0, 1)$) is as follows:

- (1) Sort $\{x_i\}_{i\in[n]}$ in ascending order. Let (x'_1,\ldots,x'_n) be the result of the sorting, and let $\sigma:[n]\to[n]$ be the permutation representing this sorting. Namely, we have $x'_i=x_{\sigma(i)}\Leftrightarrow x_i=x'_{\sigma^{-1}(i)}$ for each $i\in[n]$.
- (2) For each $k \in \{0, 1, ..., n\}$, let $y_k = y_{k,1}b_1 + \cdots + y_{k,n}b_n$) be the vector satisfying $y_{k,\sigma(j)} = 0$ for $j \le k$ and $y_{j,\sigma(j)} = 1$ for j > k. Note that $\{y_0, y_1, ..., y_k\}$ is the set of candidates of the closest vector $CV_{\mathcal{L}_{tri}}(\mathbf{x})$.
- (3) Compute $\|\mathbf{x} \mathbf{y}_k\|_2$ for each $k \in \{0, 1, ..., n\}$, and find the index $k^* \in \{0, 1, ..., n\}$ of the smallest vector such that $\mathbf{y}_{k^*} = \min_k \|\mathbf{x} \mathbf{y}_k\|_2$.
- (4) Output $y_{\min} := y_{k^*}$ as the closest vector $CV_{\mathcal{L}_{tri}}(x)$ of x.

The computational cost of the above procedure in terms of n can be estimated as follows: The sorting in Step 1 costs $O(n \log n)$. The calculation of $\|\mathbf{x} - \mathbf{y}_k\|_2$ in Step 3 for each $k \in \{0, 1, \ldots, n\}$ costs O(n). Since we calculate the distance n+1 times, Step 3 costs in total $O(n^2)$. Hence, in total we can calculate $\mathsf{CV}_{\mathcal{L}_{\mathsf{tri}}}(\mathbf{x})$ with computational cost $O(n^2)$.

The above algorithm can be extended to cover the general case where $\mathbf{x} = \sum_{i \in [n]} x_i \mathbf{b}_i$ with $x_i \in \mathbb{R}$ for each $i \in [n]$. Specifically, before executing the above algorithm, we decompose each x_i as $x_i = z_i + x_i'$ where $z_i \in \mathbb{Z}$ and $x_i' \in [0, 1)$. We then apply the above algorithm to $\{x_i'\}_{i \in [n]}$. Let \mathbf{y}_{\min} be the result. Then, the closest vector $\mathbf{CV}_{\mathcal{L}_{\text{tri}}}(\mathbf{x})$ of \mathbf{x} is $\mathbf{y}_{\min} + \sum_{i \in [n]} z_i \mathbf{b}_i$. It is easy to see the correctness of this algorithm, and that the asymptotic computational cost in terms of n remains the same.

D COMPOSING MULTIPLE FUZZY KEY SETTINGS

Our formalization of a lattice-based fuzzy key setting and the linear sketch scheme in Sec. 5 can easily be adapted to handle a "composed" fuzzy key setting and associated linear sketch scheme.

Specifically, suppose we have m kinds of fuzzy data, and for $i \in [m]$, let $\mathcal{F}_i = (X_i, X_i, \mathsf{AR}_i, \varPhi_i, e_i)$ be a lattice-based fuzzy key setting for the i-th fuzzy data, where the i-th fuzzy data space $X_i = \mathbb{R}^{n_i}$ is associated with a lattice with the basis matrix $\mathbf{B}_i \in \mathbb{R}^{n_i \times n_i}$. For simplicity, assume that the parameter p is common for all of $\{\mathcal{F}_i\}_{i \in [m]}$. Then, we can consider the composed fuzzy key setting $\mathcal{F}^* = (X^*, X^*, \mathsf{AR}^*, \varPhi^*, e^*)$ that is a natural combination of the fuzzy key settings $\{\mathcal{F}_i\}_{i \in [m]}$: The fuzzy data space X^* is the direct product $\prod_{i \in [m]} X_i$ for which the lattice \mathbf{B}^* of the following form is associated:

$$\mathbf{B}^* = \left[\begin{array}{ccc} \mathbf{B}_1 & & & \\ & \mathbf{B}_2 & & \\ & & \ddots & \\ & & & \mathbf{B}_m \end{array} \right];$$

The fuzzy data distribution X^* is the joint distribution (X_1, \ldots, X_m) , and the same for the error distribution Φ^* ; The acceptance region function AR^* has the property that for $\mathbf{x}^* = (\mathbf{x}_1, \ldots, \mathbf{x}_m), \mathbf{x}'^* = (\mathbf{x}_1', \ldots, \mathbf{x}_m') \in X^*$, we have $\mathbf{x}'^* \in AR^*(\mathbf{x}^*)$ if and only if $\mathbf{x}_i' \in AR_i(\mathbf{x}_i)$ for all $i \in [m]$; The error parameter ϵ^* can be upperbounded by $\sum_{i \in [m]} \epsilon_i$ by the union bound.

Furthermore, the algorithms of the linear sketch scheme for the composed fuzzy key setting \mathcal{F}^* can be computed by computing those for the linear sketch scheme for each fuzzy key setting \mathcal{F}_i with which \mathbf{B}_i is associated, and concatenate the results, except for the proxy key a in Sketch and the difference Δa in DiffRec. For a in Sketch and Δa in DiffRec, we need an application of the universal hash function UH for the combined linear sketch scheme whose domain is $(\mathbb{Z}_p)^{\sum_{i\in [n_i]} n_i}$ and which takes the concatenated results as input.

E FURTHER EXPERIMENTAL RESULTS

This section provides details on experimental results that were omitted in the main body.

E.1 DET curve for dimension n = 400

Fig. 15 plots the detection error tradeoff (DET) curve for fuzzy biometrics with dimension n=400. Any region that lies above the DET curve is realizable. It can be checked that for a small (resp. large) size of d (hence the acceptance region AR), we achieve better values for $\overline{\mathsf{FNMR}}$ (resp. $\overline{\mathsf{ConFMR}}$) as expected.



Figure 15: DET Curve for n = 400.

E.2 Validity of EVA result

We provide the omitted graphical evidence on the validity of our estimation of FMR and ConFMR $_t$ for $t \in [M]$. Recall that we set $k^* = 0.1\% \times k_{\max}$, where $k_{\max} = 3,925,584$ to estimate FMR using EVA. We also set M = 100 to define the M-variant of $\{\text{ConFMR}_t\}_{t \in [M]}$ and performed EVA to estimate each ConFMR $_t$ by setting $k^* = 0.5\% \times k_{\max}$. Fig. 16 illustrates the validity of our estimation for FMR when the dimension $n \in \{200,300,400\}$ and ConFMR $_t$ for $t \in \{25,50,100\}$.

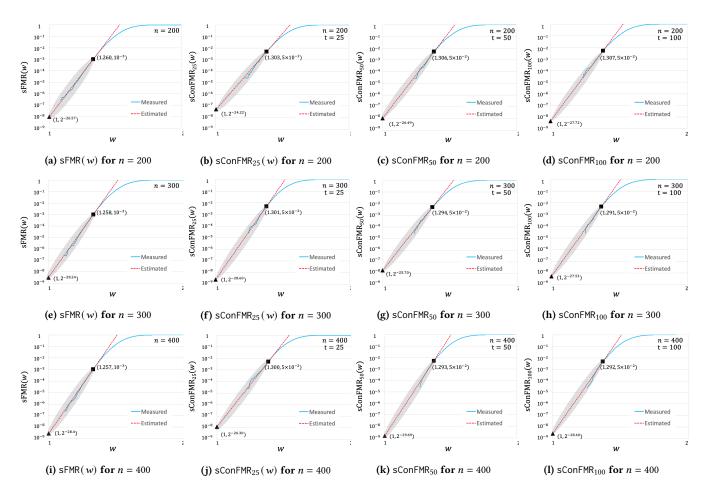


Figure 16: The blue line indicates the measured values of SFMR(w) and SCOnFMR $_{50}$ (w) w.r.t to our dataset S. The red line indicates our estimation of the probability distribution of SFMR(w) and SCOnFMR $_t$ (w) for $t \in \{25, 50, 100\}$ via EVA. The gray region is the region for which EVA provides a reliable estimation. The square plots (w^*, k^*) and the triangle plots (1, X), where X is the estimation for FMR and $\overline{\text{ConFMR}}_t$ for $t \in \{25, 50, 100\}$.

Contents 19