Improved Biometrics-Authenticated Key Exchange

Pia Bauspieß*, Tjerand Silde*, Alexandre Tullot†, Anamaria Costache*,
Christian Rathgeb‡, Jascha Kolberg‡, Christoph Busch‡
* *NTNU Norwegian University of Science and Technology,*
Email: {pia.bauspiess, tjerand.silde, anamaria.costache}@ntnu.no
† *National Higher French Institute of Aeronautics and Space, Email: alexandre.tullot@student.isae-supaero.fr*
‡ *Hochschule Darmstadt, Email: {christian.rathgeb, jascha.kolberg, christoph.busch}@h-da.de*

***Abstract*—Biometric data are uniquely suited for connecting individuals to their digital identities. Deriving cryptographic key exchange from successful biometric authentication therefore gives an additional layer of trust compared to password-authenticated key exchange. However, biometric data differ from passwords in two crucial points: firstly, they are sensitive personal data that need to be protected on a long-term basis. Secondly, efficient feature extraction and comparison components resulting in high intra-subject tolerance and low inter-subject distinguishability, documented with good biometric performance, need to be applied in order to prevent zero-effort impersonation attacks.**

In this work, we present a protocol for secure and efficient biometrics-authenticated key exchange that fulfils the above requirements of biometric information protection compliant with ISO/IEC 24745. The protocol is based on established fuzzy vault schemes and validated with good recognition performance. We build our protocol from established primitives for password-authenticated key exchange using oblivious pseudo-random functions. Our protocol is independent of the biometric modality and can be instantiated both based on the security of discrete logarithms and lattices.

We provide an open-source implementation of our protocol instantiated with elliptic curves and a state-of-the art unlinkable fingerprint fuzzy vault scheme that is practical with transaction times of less than one second from the image capture to the completed key exchange.

***Index Terms*—authenticated key exchange, oblivious pseudo-random function, fuzzy vault, biometric information protection**

1. Introduction

Biometric characteristics provide unique, non-reputable, and accurate identification of individuals over several decades [1]. This makes them suited for bridging the gap between real and digital identities in a way passwords or other machine-generated identifiers cannot. At the same time however, these properties also make them uniquely vulnerable. In particular, biometric information cannot be revoked or replaced in the same way a password or cryptographic token can. Once a digital representation of a biometric characteristics or *template* has been leaked, the underlying source (e.g., a particular finger or eye), can no longer be used securely for authentication. In fact, biometric templates provide no form of protection of the underlying data, as they can be reversed to samples sufficient for attacks [2]–[4].

Due to this risk, biometric data have been recognised as sensitive personal data by the European Union's General Data Protection Regulation (GDPR) [5] and the ISO/IEC 24745 International Standard on Biometric Information Protection [6]. The latter defines three security requirements for secure biometric systems: *i) unlinkability and renewability*, meaning that an attacker cannot connect two protected biometric templates stored in different applications, and new templates from the same source look indistinguishable to a previously stored reference, *ii) irreversibility*, it should be impossible for an attacker to retrieve original samples given only protected templates, and *iii) performance preservation*, the computational performance and the recognition accuracy of the system should not be impacted significantly by adding a layer of protection to the original data.

At first sight, the performance preservation requirement in ISO/IEC 24745 seems to be a question of convenience only. However, it details a second and crucial dimension that determines the security of biometric authentication: the accuracy of the underlying biometric comparison function. Contrary to passwords, which can be compared in an exact manner, captured samples of the same biometric characteristic are never exactly equal, but *fuzzy*. They are subject to noise such as aging, environmental influence, or image quality. Comparison of two samples is therefore based on some measure of similarity. If this measure is too imprecise, or the feature representation is not discriminative enough, an authentication system is not capable of accurately distinguishing between mated comparisons, where the samples stem from the same subject, and non-mated authentication attempts, where the samples stem from different subjects. Trust in the derived authentication is consequently low.

Recently, the idea of building authenticated key exchange on the basis of biometrics has gained interest with the proposal of Biometrics-Authenticated Key Exchange (BAKE) [7]. Analogously to password-authenticated key exchange (PAKE) [8], a client and server negotiate a shared cryptographic key that should be equal if and only if the biometric authentication was successful.

With their protocol, the authors of [7] achieve security in terms of the protection of the biometric data. However, their biometric comparator is vulnerable, as we show by reproducing their results experimentally. The reason for this imprecision is a fingerprint comparison algorithm that is specific to their protocol, but has not been evaluated in terms of biometric performance (i.e., accuracy). We give this evaluation and show that the algorithm is barely able to distinguish between mated comparison trials within the same identity and non-mated comparison trials between different identities. More generic protocols both on symmetric fuzzy PAKE [9] and asymmetric fuzzy PAKE [10] have been proposed. However, with regard to biometrics, they have the following shortcomings: symmetric fuzzy PAKE [9] does not achieve protection of the biometric data, which is shared with the server in cleartext. Asymmetric fuzzy PAKE [10] achieves security in both dimensions in theory, but is inefficient in practice as it is based on generic oblivious transfer which is performed once for each bit in the biometric template. In addition, [9] and [10] only enable comparison of fixed-length biometric representations. The most accurate comparison metric for fingerprints, one of the most popular biometric modalities, is however based on variable-length representations, the similarity of which cannot be expressed as a distance function.

1.1. Contribution

In this work, we present a protocol for biometrics-authenticated key exchange that addresses the deficiencies of previous works [7], [9], [10] and achieves effective protection of the biometric data while building on accurate and established biometric comparison functions that have been evaluated and improved in the literature over many years. Our protocol is efficient with execution times of under one second from the biometric capture to the completed key exchange, including communication cost. More precisely, we contribute:

- Secure biometrics-authenticated key exchange from established primitives: fuzzy vaults [11], oblivious pseudo-random functions (OPRF), and key encapsulation mechanisms (KEM). Our two-round protocol can be instantiated both with a discrete logarithm OPRF [8] and Diffie-Hellman key exchange [12] as well as lattice-based OPRF [13] and KEM [14].
- Interchangeability of biometric modalities: our protocol can be instantiated with any fuzzy vault scheme and thereby different biometric modalities and feature representations. In particular, it is compatible with both fixed-length and variable-length representations of biometric characteristics.
- Resistance against offline attacks: one known flaw of fuzzy vault schemes for biometric authentication are offline attacks. In our protocol, we remove the checksum typically used to verify authentication attempts and replace it with authenticated key exchange that requires interaction for every attempt.
- Protection of the biometric data in storage and transfer compliant with ISO/IEC 24745 [6]: if the underlying fuzzy vault scheme achieves unlinkability, renewability, irreversibility and performance preservation, our protocol preserves these properties.
- Open-source implementation: an implementation of our protocol instantiated with elliptic curves and a state-of-the art unlinkable fingerprint fuzzy vault scheme [15] is available at https://anonymous.4open.science/r/bake-7057. We show that our protocol achieves real-world efficiency with transaction times of under one second from the fingerprint image capture at the sensor to the completed key exchange.

1.2. State-of-the-Art

We look at the state-of-the-art to motivate two principles for secure biometrics-authenticated key exchange: recognition accuracy and reciprocal interaction.

The main concern with the protocol proposed in [7] is the generation of the biometric secret key constructed from fingerprint representations. The authors use a simplified version of the well-studied nearest-neighbour approach first proposed by [16], which they chose due to its anticipated rotation invariance. However, this algorithm and its flaws have been studied for two decades, specifically, its inability to tolerate missing genuine minutiae [17]. It has therefore been found unusable in practice, and improved rotation-invariant fingerprint recognition algorithms have been proposed that mitigate the known shortcomings [17]. Such improved algorithms require a more complex comparison subsystem however, and are not compatible with the constructor offered in [7]. Notably, the paper [7] fail to state the recognition accuracy of their iris and fingerprint based protocols, and do not give an experimental evaluation detailing the security with regard to the biometric performance. Their construction for iris is based on the established fixed-length feature representation IrisCode [18] and can be assumed to achieve adequate accuracy.

Secondly, the public keys derived from the biometric secret keys in [7] are vulnerable to offline attacks: in their construction, any adversary can guess a biometric template and attempt to decode a chosen message encapsulated with a public key without interacting with another party. In such an attack, the adversary does not have to guess an exact biometric feature representation, but succeeds as soon as she finds an input that is close enough with regard to the distance metric used. This probability can be expressed as the false-match rate of the biometric system, i.e., the proportion of authentication attempts from non-mated samples falsely accepted as authentication attempts of an enrolled data subject. Again, low accuracy leads to a low effort in an offline search attack.

Even with high accuracy, offline attacks expose biometric data to high risks. Therefore, we construct our protocol such that interaction is required for every adversarial guess,

TABLE 1. COMPARISON OF OUR PROTOCOL TO RELATED WORK.

Scheme	Asymmetric	Efficient	Accurate	ISO/IEC 24745
fPAKE [9]	✗	✓	✓	✗
fuzzy aPAKE [10]	✓	✗	✓	✗
BAKE [7]	✓	✓	✗	✗
Ours	✓	✓	✓	✓

which allows for rate-limiting that can be enforced as long as at least one party remains honest.

An overview of how our proposed scheme compares to the works discussed above can be found in Table 1.

Other related work has been directed on extracting uniformly distributed cryptographic keys directly from biometric templates without running an interactive protocol [19]. Similar to [9] and [10], only fixed-length representations are considered that can be compared with some distance metric. From fuzzy extractors, two-factor authentication protocols have been built [20]. More recently, [21] proposed a session key generation protocol specifically for fingerprint based on so-called cancellable biometrics, which are one-way transforms on the biometric data that are not based on well-studied cryptographic problems and can therefore not be assumed to underlie certain hardness assumptions.

1.3. Technical overview

Before we describe our protocol in detail, we give a conceptual overview of our approach.

Fuzzy vault. The fuzzy vault scheme first proposed by [11] builds on error-correcting codes, more specifically, Reed-Solomon codes. First, a random polynomial f of degree $\tau-1$ is generated. Then, the elements of a biometric feature set t are encoded onto f as $(a, f(a))$ for $a \in t$. In the original scheme, these true points corresponding to t are then hidden by a large number of random coordinates (x, y) that do not lie on f. The union of both sets of points is then regarded a *locked fuzzy vault* V, which is stored at a server together with a hash $H(f)$ for further reference.

It has to be noted that this original construction has been found to be insecure against correlation attacks between different locked fuzzy vault records [15]. Next to the large memory requirements, finding random points that hide the secret polynomial f truly is a hard problem. Therefore, improved schemes have been developed by [15], which is the scheme we use in our implementation.

For verification, a biometric probe feature set t' is captured. From the locked fuzzy vault V, only points with x-coordinates in the set t' are selected and a polynomial f' is interpolated over these points. If the intersection is larger than τ, meaning that at least τ points in t and t' align, then $H(f') = H(f)$, and the verification is successful. Here, τ refers to the biometric decision threshold, which is chosen as the correction capacity of the Reed-Solomon code.

In addition to correlation attacks, which have been mitigated by [15], a persisting point of attack in this protocol is provided by the checksum $H(f)$. By guessing a biometric template t', running the verification protocol, and comparing the hash of the result $H(f')$ with the provided checksum $H(f)$, an attacker can run an offline brute-force search effectively and efficiently. Note that there is no need to try and guess random codewords, i.e., secret polynomials. Rather, it is sufficient to guess some template t' that is within distance τ of the stored biometric template t, which has significantly lower brute-force security.

In our protocol, we therefore omit the computation and storage of $H(f)$, and replace it with an OPRF evaluation followed by a KEM. Thereby, we gain two improvements in one: firstly, the fuzzy vault scheme becomes secure against offline brute-force attacks. Secondly, instead of a binary verification, we can derive a shared cryptographic key if and only if the biometric verification was successful.

OPRF evaluation. The secret polynomial f is the information that is used as input to the OPRF. The evaluation that takes place obliviously is a signature using a secret key k held by a third party we call the signer. In practice, this party can be instantiated with a secure hardware execution environment located at the server. Its only objective is evaluating the OPRF, and it therefore at no point sees any biometric information. From the OPRF evaluation, the client receives a signature of f without learning the signing key k, and the signer does not learn f. Based on this signature, a secret key $\mathtt{sk}_t$ with respect to template t is derived, and its corresponding public key $\mathtt{pk}_t$ can be computed accordingly.

The evaluation of the OPRF is the component that enforces an interaction for each guess a biometric template. In terms of a brute-force attack, the public key $\mathtt{pk}_t$ allows for similar confirmation of a correct guess as was previously provided by the hash $H(f)$. However, in order to compute $\mathtt{pk}_t$, a signature needs to be obtained. As long as the OPRF key k remains secret, an offline search is therefore infeasible.

Key exchange. During an enrolment phase, a public key $\mathtt{pk}_t$ derived from template t is stored at the server. For authentication, a client computes a fresh key pair $(\mathtt{sk}_{t'}, \mathtt{pk}_{t'})$ derived from her freshly captured feature vector t'. The server now encapsulates a cryptographic key using the user's stored reference public key $\mathtt{pk}_t$. The user can decapsulate the key if and only if her fresh probe secret key $\mathtt{sk}_{t'}$ corresponds to the stored public key. Due to the fuzzy vault construction, $(\mathtt{sk}_{t'}, \mathtt{pk}_t)$ will only be a meaningful key pair if $f' = f$, i.e., only if t and t' are within correction capacity τ.

1.4. Structure of Paper

The rest of this paper is structured as follows: In Section 2, background information and definitions required for the construction of our protocol are presented. As our main contribution, Section 3 presents our BAKE protocol with security definitions and proof sketches. Section 4 presents the experimental evaluation of the protocol and practical

TABLE 2. SELECTED BIOMETRIC FEATURE REPRESENTATIONS.

Modality	Template	Fixed-Length	Ordered	Type
Face	DCNN embedding	✓	✓	float
Fingerprint	Minutiae	✗	✗	integer
	FingerCode	✓	✓	binary
Iris	IrisCode	✓	✓	binary

comparison with related work, before we outline our conclusions in Section 5.

2. Preliminaries

The framework for automated and interoperable biometric recognition has been standardised in ISO/IEC 19794-1 [22], and subsequent parts of the standard define biometric data interchange formats for the modalities fingerprint, face, iris, voice, handwritten signatures, and vascular biometrics. For the scope of our work, we look at the three most prevalent modalities fingerprint, face, and iris, for which well-tested fuzzy vault schemes exist. An overview of common feature representations is given in Table 2.

2.1. Fingerprint Recognition

The representation extracted from a fingerprint sample to be used for biometric recognition is its ridge lines, which can be captured both with capacitive, optical, or touchless sensors. From the pattern of ridge lines, significant points known as *minutiae* are extracted as compact and distinguishing features, specifically, ridge endings and bifurcations, namely the location and orientation where one ridge line splits into two. As specified in ISO/IEC 19794-2 [23], a minutiae template is represented as a list of tuples (x, y, θ) of the x- and y-coordinates of the minutiae given in pixels from the left upper corner of the captured image along with their tangential angle θ with respect to the x-axis. It is important to note that a set of minutiae has no meaningful inherent ordering, even though the template lists them by x-coordinate. The typical number of genuine minutiae in a human fingerprint ranges from 40-100 [7], depending on the image quality and environmental factors during capture.

While minutiae-based representations have the potential for high recognition accuracy, they come with the challenge of potential rotation of the captured sample and hence the cloud of minutiae points and non-linear transformations that need to be addressed through costly and difficult pre-alignment processes. Therefore, fixed-length fingerprint representations have been proposed, the most prominent of which is the FingerCode [24] representation. Using a set of Gabor filters, FingerCode templates yield a translation-invariant and to some degree rotation-invariant representation of a fingerprint image. Most importantly, FingerCode templates are of fixed-length and ordered by dimension, which enables the use of simple comparison functions such as Hamming distance or Euclidean distance. Such functions produce dissimilarity scores, such that a verification attempt is accepted when the comparison score is below the threshold τ, and rejected else.

Note that generally speaking, rotation invariance is a property independent of minutiae-based or fixed-length representations, even though it is more commonly found in the latter. A fitting example is Minutia Cylinder Code (MCC) [17], a rotation-invariant minutiae-based template representation. Approaches to handle rotation and pre-alignment for minutiae templates include [25] and [26]. For the scope of our work, we do not deal with the challenge of pre-alignment further, but assume user guidance through the capture process, e.g., through the hardware design of the capture device. In our experimental evaluation, we use a pre-aligned dataset to model this scenario adequately [27].

Minutiae-based comparators are more complex due to the problem of finding an accurate mapping between two unordered, noisy sets containing of a variable number of two-dimensional points. Even for mated comparison trials, the number of detected minutiae and their location varies depending on the image quality and possible impairing factors such as dirt, wounds, or water on the finger. Common approaches to minutiae comparators have been based on closest neighbours [16], fixed-radius neighbourhoods [17], or graph-based approaches [28]. Despite their differences and individual shortcomings, they share one common aim: at least τ minutiae points need to be mapped uniquely between the two sets, such that each pairing is considered a matching minutiae pair.

2.2. Biometric Performance Metrics

Biometric performance testing and reporting is standardised in ISO/IEC 19795-1 [29] and subsequent parts of the standard. Reporting the performance of a biometric system within this framework is an important foundation for benchmarking, reproducibility, and reliability of research in biometrics.

The evaluation of biometric systems is based on two components: error rates and throughput rates. In terms of throughput rates, both the computational speed of the transaction and the time needed for the user to interact with the system are considered. Error rates report on the accuracy of the system. For a verification scenario, the most important terms and metrics are:

- *False Non-Match Rate (FNMR):* proportion of mated comparisons that resulted in a reject decision.
- *False Match Rate (FMR):* proportion of non-mated comparisons that resulted in an accept decision.

The FMR can be thought of as the security level of the biometric system, detailing how many zero-effort imposters were able to be verified. In real-life scenarios, systems with a FMR below 1% are considered secure, while high-security applications such as automated border control require a FMR lower than 0.1% [30]. The FNMR on the other hand can be considered as the convenience level of the system, detailing how many mated comparison trials were not able to be verified. A FNMR up to 5% is considered acceptable.

The trade-off between FMR and FNMR can be plotted as a Detection Error Trade-Off (DET) curve, where the FMR and FNMR are computed for every comparison score in the test set as the decision threshold. The advantage of a DET compared to single-number statistics is therefore its threshold independence.

Factors impacting the recognition performance of a biometric system are first and foremost the sample quality both during enrolment and verification, and the robustness of the feature representation and comparison algorithm with regard to rotation, translation, and noise of the samples. Furthermore, any feature transformation such as binarisation may impact the accuracy of the system.

2.3. Fuzzy Vault

The concept of fuzzy vaults was first introduced by [11], who propose a scheme that allows to *lock* a secret f using a biometric feature secret set t using a probabilistic algorithm. The output of this algorithm is a *locked* fuzzy vault that can be *unlocked* using a second biometric feature set t', if there are enough points the intersection of t and t'. We give a short definition of their original scheme before we move on to the state-of-the-art for different biometric modalities.

Definition 1 (Fuzzy Vault Scheme [11])**.** Let $\mathcal{C}$ be an error-correcting code (e.g., a Reed-Solomon code) with correction capacity τ and $H : \mathcal{C} \to \{0,1\}^{2\lambda}$, for security parameter λ, be a cryptographic hash function. Then, a *fuzzy vault scheme* is a set of the following algorithms:

- $(f, V) \leftarrow \texttt{lock}(t)$: On input of a biometric feature set t, the algorithm samples a random secret polynomial $f \in \mathcal{C}$ and outputs a locked fuzzy vault V.
- $f' \leftarrow \texttt{unlock}(V, t')$: On input of a locked fuzzy vault V and a biometric feature set t', the algorithm outputs an opening polynomial $f' \in \mathcal{C}$.

The reconstructed output can be compared against the original input using $H(f)$. A basic authentication protocol based on fuzzy commitment is given in Figure 1.

Enrolment

Setup party		**Server**
t reference template		
$(f, V) \leftarrow \texttt{lock}(t)$		
	$\xrightarrow{H(f), V}$	
		store $(H(f), V)$

Verification

Client		**Server**
t' probe template		$(H(f), V)$
	$\xrightarrow{\text{request } H(f), V}$	
	$\xleftarrow{H(f), V}$	
$f' \leftarrow \texttt{unlock}(V, t')$		
return $H(f) = H(f')$		

Figure 1. Fuzzy vault authentication protocol based on [11].

Instantiation for Fingerprint. The original schemes by [11] and a similar scheme by [31] have been proven to be insecure due their construction based on large point clouds to hide the secret f, which are vulnerable to correlation attacks [32]. Therefore, [15] presented an improved scheme to mitigate correlation attacks (see [15], Section 1.2.3) that fulfils the requirements of ISO/IEC 24745 [6].

The improved fuzzy vault scheme has first been constructed for minutiae-based fingerprint representations [15]. Here, minutiae are encoded into a finite field $\mathbb{F}_p$ using absolute pre-alignment and quantisation to account for a certain degree of noise with regard to the position of the minutiae. The set of minutiae $t \subset \mathbb{F}_p$ is then considered the biometric template. A polynomial $f \in \mathbb{F}_p[x]$ of degree $\tau - 1$ is chosen uniformly at random and locked as

$$\texttt{lock}(t) = (f, f(x) + \prod_{a \in t} (x - a)) =: (f, V).$$

To unlock the vault, V is evaluated on the probe minutiae set t' and decoded using a Reed-Solomon decoder, yielding

$$\texttt{unlock}(V, t') = \text{decode}(\{(b, V(b)) \mid b \in t'\}) =: f'.$$

Lemma 1 (Theorem 1 in [15])**.** Let $(f, V) \leftarrow \texttt{lock}(t)$ be a commitment to a polynomial $f \in \mathbb{F}_p[x]$ with minutiae set t, and $f' \leftarrow \texttt{unlock}(V, t')$ an unlocking of V using a minutiae set t'. Then, $f = f'$ if and only if $|t \cap t'| \geq \tau$.

Analogue constructions exist for iris [33] and face [34] recognition, which we refer the reader to for full details.

2.4. Entropy of Biometric Representations

The entropy of biometric data is a topic that is often referred to in works about fuzzy cryptographic primitives [9]. In the literature, the entropy of a face has been determined at 56 bits [35], a minutiae-based fingerprint representation at 82 bits [36], and an iris at 249 bits [37]. However, these numbers can only be considered as an upper bound of the entropy of a certain biometric instance, as the amount of information in a biometric sample heavily depends on the capture device used and its fidelity (e.g., its resolution) as well as the feature extraction algorithm used.

In addition, [15] argues that it is not in all scenarios sensible to use the entropy of a single biometric template as a measure for security, which is an overestimate when it comes to comparisons between biometric features. Here, the false-accept security defined as $\log_2(FMR^{-1})$ gives a more accurate measure, as it is sufficient for an attacker to guess a template that is close enough to a reference template.

2.5. Cryptographic Primitives

Definition 2 (Pseudo-Random Function, [38])**.** A family of functions $f_k : \{0,1\}^m \to \{0,1\}^n$ with key $k \in \{0,1\}^\lambda$ is called pseudo-random (PRF) if the following holds:

- $f_k(x)$ is efficiently computable from k and x.
- It is not efficiently decidable whether one has access to a computation oracle for $f_k(\cdot)$ or to an oracle producing uniformly random bit strings of length m.

Definition 3 (Oblivious Pseudo-Random Function, [39])**.** A two-party protocol π between a client and a server is an *oblivious pseudo-random function (OPRF)* if there exists some PRF family f_k, such that π privately realizes the following functionality:

- Client has input x; Server has input k.
- Client outputs $f_k(x)$; Server outputs nothing.

Definition 4 (Hashed Diffie-Hellman OPRF, [40])**.** Let $\mathbb{G}$ be a cyclic group, $x \in \{0,1\}^*$ the client input, $k \in \mathbb{Z}_q$ the server's secret key, $H_{\mathbb{G}} : \{0,1\}^* \to \mathbb{G}$ and $H_{\mathbb{Z}_q} : \{0,1\}^* \to \mathbb{Z}_q$ hash functions that output values $\mathbb{G}$ and $\mathbb{Z}_q$, respectively. The protocol `HashDH` consists of the following algorithms:

- $(B, r) \leftarrow \mathtt{blind}(x)$: The client samples a random $r \leftarrow\$\ \mathbb{Z}_q$ and outputs $B \leftarrow [r]H_{\mathbb{G}}(x)$.
- $S \leftarrow \mathtt{sign}(B, k)$: On input $B \in \mathbb{G}$, the server outputs $S \leftarrow [k]B$.
- $\mathtt{sk} \leftarrow \mathtt{unblind}(S, r)$: On input $S \in \mathbb{G}$ and $r \in \mathbb{Z}_q$, the client outputs $\mathtt{sk} \leftarrow H_{\mathbb{Z}_q}(x, [r^{-1}]S)$.

As a result of this protocol, the client privately obtains $H_{\mathbb{Z}_q}(x, [k]H_{\mathbb{G}}(x))$ without learning k and without the server learning x.

Definition 5 (Key Encapsulation Mechanism, [41])**.** A key encapsulation mechanism (KEM) is a scheme with three algorithms `KeyGen`, `encap` and `decap`, where

- $(\mathtt{pk}, \mathtt{sk}) \leftarrow \mathtt{KeyGen}(1^\lambda)$: takes as input the security parameter λ and outputs a public key `pk` and a secret key `sk`.
- $\mathtt{ctx} \leftarrow \mathtt{encap}(\mathtt{ck}, \mathtt{pk})$: takes as input a key `ck` and public key `pk` and outputs an encapsulation $\mathtt{ctx} \leftarrow \mathtt{encap}(\mathtt{ck}, \mathtt{pk})$ of `ck` under the public key `pk`.
- $\mathtt{ck} \leftarrow \mathtt{decap}(\mathtt{ctx}, \mathtt{sk})$: takes as input an encapsulated key `ctx` and a secret key `sk` and outputs a decapsulated key `ck`.

We require that for all $(\mathtt{pk}, \mathtt{sk})$ generated from `KeyGen` we have that $\mathtt{ck} = \mathtt{decap}(\mathtt{encap}(\mathtt{ck}, \mathtt{pk}), \mathtt{sk})$, except with negligible probability, and that the scheme is CCA secure.

A KEM can, e.g., be instantiated with (Elliptic Curve) Diffie Hellman [42], RSA [43] or CRYSTALS-Kyber [14].

3. Biometrics-Authenticated Key Exchange

In this section, we introduce our protocol for Biometrics-Authenticated Key Exchange (BAKE) built from generic fuzzy commitment schemes, oblivious pseudo-random functions (OPRFs) and key-encapsulation mechanisms (KEMs).

3.1. Setting

For our proposed protocol, we assume that a biometric capture device is linked to a client which performs the preprocessing and feature extraction, and acts as a communicating party in the protocol. Its communication counterparts are a server which controls a database of locked fuzzy vaults and client reference public keys, and a signer which is in possession of a secret OPRF key. In practice, the signer can be instantiated by a trusted execution environment at the server. For this reason, we do not model direct communication between the client and the signer, but work under the harder assumption that all communication between client and signer is seen by the server. This is a common practice in biometric information protection [44], as it allows for enhanced network security choices that protect the party handling secret key material. For example, the signer can be set up in a local area network that does not have to be accessible over the internet.

3.2. Threat model

The goal of an adversary taking control over one or more of the parties participating in the `BAKE` protocol is to obtain or guess a biometric feature vector that is close enough to an enrolled reference template to authenticate to either this or other systems, or to retrieve personal information about the enrolled data subjects from it. Guessing a feature vector is always an attack on a biometric system. However, two measures can be taken to prevent an attacker from authenticating with a guess: firstly, presentation attack detection (PAD) can be applied at the capture device. In reality, it is a hard problem to construct a presentation attack artefact, e.g., a silicone finger with a stolen fingerprint, that is sufficiently realistic to pass PAD barriers. Secondly, repeated guesses of biometric feature vectors are only feasible if the attacker receives confirmation that the guess is correct. In our protocol, such confirmation can only be obtained through an OPRF evaluation, which itself requires interaction. Therefore, we enforce rate-limiting on the number of repeated authentication attempts both at the server and the signer, such that brute-force attempts can be detected and denied.

For offline brute-force searches, a secondary attack mechanism of an adversary is to obtain the secret OPRF key held by the signer in order to run an offline brute-force search on the reference database. With regard to the enrolment database, we assume an honest enrolment transaction for all reference subjects for which information is stored in the database. In practice, this could be realised by a trusted third party we refer to as the setup party. Going forward, we only model security for the verification transactions of the system.

Given these threats, we work under the assumption that PAD is applied at the capture device, and that the capture device is always honest in the sense that it does not store or publish the biometric features it sees. It is evident that a client wins the security game trivially when it stores and discloses templates from data subjects. Therefore, we model the case where an adversary wants to learn templates from subjects who do not provide it to the capture device. With regard to man-in-the-middle attacks, we assume authenticated communication channels between all parties, such that an adversary needs to gain control of a party that is actively involved in the protocol. Similarly, the parties win trivially when all three of them collude, which is why we do not model this case in more detail.

Overall, we assume that an adversary implicitly keeps a state of all information it has seen from previous algorithms and that any adversary has access to a realistic amount of classical computing power and is not restricted from running an efficient brute-force search in terms of storage or computation power.

3.3. Modification of Fuzzy Vault Schemes

In the original improved fuzzy vault schemes, the decoding algorithm with highest performance both in terms of execution times and accuracy is the Guruswami-Sudan decoder [45]. In all three fuzzy vault schemes discussed in our work [15], [33], [34], the algorithm of [45] is used in a list decoding mode. Unlocking a fuzzy vault with feature vector t' corresponds to a randomised brute-force decoding strategy, where subsets of t' are chosen uniformly at random and evaluated as unlocking sets for the reference fuzzy vault.

During this randomised decoding, a candidate polynomial f' is generated for each subset and compared against the stored hash $H(f)$ corresponding to the biometric reference template t. When a candidate polynomial is found for which $H(f) = H(f')$, the decoding attempts are stopped. If no candidate polynomial is found within a certain number of decoding attempts, the underlying comparison of t and t' is classified as a non-mated comparison trial.

In our protocol however, we do not wish to store $H(f)$ at the server as it allows for offline brute-force attacks. Instead, we run the full decoding attempts until the threshold for non-mated comparison trials is reached, even when we expect a mated comparison trial. During decoding, we temporarily store all candidate polynomials and sort them with respect to their frequency. For a mated comparison, we expect the correct candidate polynomial f' for which $H(f') = H(f)$ to appear as the most frequently reconstructed polynomial due to the large overlap of the sets t and t'. A similar strategy is applied in [31] and is supported by our experimental evaluation, showing only a negligible deviation with regard to the biometric performance between the hash-verified decoding and highest-frequency decoding strategies.

Notably, the FMR and thereby security of the system is not affected by the change to highest-frequency decoding, as non-mated comparisons still yield no matching candidate polynomial. If no matching candidate polynomial has been found before the threshold of the list decoder, this includes the polynomial that occurs most often. Therefore, only changes in the FNMR rate or convenience of the overall system can be expected, when the most frequent polynomial in a mated comparison is not the matching candidate polynomial. This occurs for example in cases where the second most frequent polynomial is the matching candidate polynomial. If one wished to improve upon the FNMR, a viable strategy would be running the authentication protocol for a certain number of most frequent polynomials. However, for the scope of our work, the FNMR degradation is not significant, and most importantly, the security in terms of FMR is not impacted.

3.4. Protocol

Definition 6. The BAKE protocol consists of the following algorithms:

- $\mathtt{pp} \leftarrow \mathtt{setup}(1^\lambda)$: The setup algorithm defines a universe $\mathcal{P}$, randomness space $\mathcal{R}$, key space $\mathcal{K}$ and a cryptographic hash function $H : \{0,1\}^* \to \{0,1\}^{2\lambda}$. Further, the setup algorithm publicly defines an error-correcting code $\mathcal{C}$ with correction capacity τ. All following algorithms implicitly inherit pp.
- $(f, V) \leftarrow \mathtt{lock}(t)$: The algorithm takes as input a biometric template t, samples a random polynomial $f \in \mathcal{C}$, and outputs f and a locked fuzzy vault V.
- $f' \leftarrow \mathtt{unlock}(V, t')$: The algorithm takes as input a biometric probe feature vector t' and locked fuzzy vault V, and outputs an opening polynomial f'.
- $(B, r) \leftarrow \mathtt{blind}(f)$: The algorithm samples a random element $r \in \mathcal{R}$ and outputs an element $B \in \mathcal{P}$.
- $S \leftarrow \mathtt{sign}(B, k)$: On input $B \in \mathcal{P}$ and key $k \in \mathcal{K}$, the server outputs a signature $S \in \mathcal{P}$.
- $\mathtt{sk} \leftarrow \mathtt{unblind}(S, r)$: On input $S \in \mathcal{P}$ and $r \in \mathcal{R}$, the algorithm outputs an secret key $\mathtt{sk} \in \mathcal{K}$.
- $(\mathtt{sk}, \mathtt{pk}) \leftarrow \mathtt{KeyGen}(1^\lambda)$: The algorithm outputs a secret key $\mathtt{sk} \in \mathcal{K}$ and a public key $\mathtt{pk} \in \mathcal{P}$.
- $\mathtt{pk} \leftarrow \mathtt{pkGen}(\mathtt{sk})$: The algorithm takes as input a secret key $\mathtt{sk} \in \mathcal{K}$ and outputs a corresponding public key $\mathtt{pk} \in \mathcal{P}$.
- $\mathtt{ctx} \leftarrow \mathtt{encap}(\mathtt{ck}, \mathtt{pk})$: The algorithm takes as input a cryptographic key ck and outputs its encapsulation $\mathtt{ctx} \leftarrow \mathtt{encap}(\mathtt{ck}, \mathtt{pk})$ under the public key pk.
- $\mathtt{ck} \leftarrow \mathtt{decap}(\mathtt{ctx}, \mathtt{sk})$: The algorithm takes as input an encapsulated key ctx and a secret key sk and outputs a decapsulated key ck.

The BAKE protocol is executed between the three parties client, server and signer in two modes: enrolment, which is given in Figure 2 and verification, given in Figure 3.

During enrolment (Figure 2), a client public key $\mathtt{cpk}_t$ is derived from a biometric reference template t, and stored at the server together with a locked fuzzy vault V of t using a secret random polynomial f. First, the client generates f and locks the vault with template t. Then, it initiates the OPRF evaluation on input f. The signer signs the blinded input f using the OPRF key k, and client is able to unblind and obtain its secret key $\mathtt{csk}_t$, from which it computes the corresponding public key $\mathtt{cpk}_t$. To conclude the enrolment step, the client sends the tuple $(V, \mathtt{cpk}_t, \mathtt{id})$ to the server to be stored for future reference.

For verification and key exchange (Figure 3), the client requests the fuzzy vault V stored at the server for identity id, and, using a biometric probe t', unlocks the vault to a polynomial f'. Then, the OPRF evaluation on f is computed analogously to the enrolment step. At the same time, the client and server generate ephemeral key pairs to prepare the key exchange. Additionally, the server has a static key pair $(\mathtt{ssk}, \mathtt{spk})$ that is not derived form any biometric information. We assume that the client has already stored spk, such that the server does not need to transmit it each time. Once all keys have been generated, the server encapsulates a common key using the client's public key $\mathtt{cpk}_t$. The client can decapsulate the common key if and only if the secret reconstructed from the fuzzy vault was correct, i.e., in the case where t and t' are closer than threshold τ.

3.5. Security Definitions

Following the definition of the BAKE protocol in Figures 2 and 3, we give formal definitions of the security of the protocol. For simplicity, we model the use of identifiers within the enrolment database implicitly. In theory, an adversary wants to learn a biometric feature vector that is close to any enrolled template. In practice however, it always needs to choose a specific identity to attack or run attacks on multiple specific identities in parallel. The following definitions and proof sketches model security in the case where a template t is enrolled in the database held by the server, and an honest client would use a feature vector t' to authenticate.

Denote by $f^{-1} = \log_2(FMR^{-1})$ the false-accept security of a biometric feature extractor and comparator, let ℓ be the rate limit enforced by the server and the signer, and let $\ell_{\mathcal{A}}$ be the brute-force capacity of the attacker $\mathcal{A}$.

Definition 7. (Correctness) We say that a BAKE protocol is correct, if a capture subject presenting a biometric probe feature vector t' and identifier id can successfully authenticate to an honest server if and only if $\mathtt{dist}(t', t_{\mathtt{id}}) \leq \tau$ for a fixed biometric verification threshold τ, except with negligible probability.

Definition 8. (Client Privacy) We say that a BAKE protocol has client privacy, if an adversary $\mathcal{A}$ controlling the client has the following advantage in obtaining a biometric feature vector t' that is close to an enrolled biometric template t:

$$\Pr\left[\mathtt{dist}(t,t') < \tau \ : \ \begin{array}{c} \mathtt{pp} \leftarrow \mathtt{setup}(1^\lambda) \\ \{V, \mathtt{cpk}_t\} \leftarrow \mathtt{enroll}(\mathtt{pp}, t) \\ (B', \mathtt{cpk}_e) \leftarrow \mathcal{A}(\mathtt{pp}, V) \\ (\mathtt{ssk}_e, \mathtt{spk}_e) \leftarrow \mathtt{KeyGen}(1^\lambda) \\ S' \leftarrow \mathtt{sign}(B', k) \\ t' \leftarrow \mathcal{A}(S', \mathtt{spk}, \mathtt{spk}_e, \mathtt{ctx}) \end{array}\right] \leq \ell f^{-1} + \mathsf{negl}(\lambda).$$

Definition 9. (Server Privacy) We say that a BAKE protocol has server privacy, if an adversary $\mathcal{A}$ controlling the computation server has the following advantage in obtaining a biometric feature vector t' that is close to an enrolled biometric template t:

$$\Pr\left[\mathtt{dist}(t,t') < \tau \ : \ \begin{array}{c} \mathtt{pp} \leftarrow \mathtt{setup}(1^\lambda) \\ \{V, \mathtt{cpk}_t\} \leftarrow \mathtt{enroll}(\mathtt{pp}, t) \\ B' \leftarrow \mathcal{A}(\mathtt{pp}, \{V, \mathtt{cpk}_t\}) \\ S' \leftarrow \mathtt{sign}(B', k) \\ t' \leftarrow \mathcal{A}(S') \end{array}\right] \leq \ell f^{-1} + \mathsf{negl}(\lambda).$$

If client and server run the protocol BAKE honestly, the signer only sees the blinded element, which is information-theoretically secure, and hence, independent of the biometric template. We therefore do not model signer privacy.

The advantage of an adversary controlling both the client and the server effectively reduces to server privacy. In this scenario, the information the adversary needs to guess is the signed element S'. However, as discussed above, the signer cannot distinguish between signing requests for different biometric feature vectors corresponding to mated authentication attempts, or repeated signing requests for a single identity aimed at running a brute-force search. Therefore, rate-limiting at the signer can be enforced by user-specific OPRF keys. This way, the signer will learn the identifier of the user attempting to authenticate, but is not able to gain any more knowledge about her biometric data, while effectively preventing the server from learning it.

The advantage of an adversary controlling both the client and the signer initially reduces to the definition of client privacy, as the adversary seeks to learn the reference public key stored during enrolment. However, after running one (unsuccessful) authentication attempt for a specific identity, the adversary will receive the encapsulated key derived from the biometric reference data of the data subject in question. From that point on, it can guess a biometric feature vector, issue a a signature by use of the signing key, and compare the resulting key against the obtained one. Therefore, we realistically model an adversary controlling both the client and the signer as being able to run an offline search on the biometric enrolment database. We note that due to the architecture considerations, this scenario is somewhat unlikely in practice, and a more realistic threat is the server and signer colluding.

Definition 10. (Client-Signer Privacy) We say that a BAKE protocol has *client-signer privacy*, if an adversary $\mathcal{A}$ controlling both the client and the authentication server does not have an advantage in obtaining a biometric feature vector t' that is close to any enrolled biometric template t above running a brute-force search on V.

$$\Pr\left[\mathtt{dist}(t,t') < \tau \ : \ \begin{array}{c} \mathtt{pp} \leftarrow \mathtt{setup}(1^\lambda) \\ \{V, \mathtt{cpk}_t\} \leftarrow \mathtt{enroll}(\mathtt{pp}, t) \\ (B', \mathtt{cpk}_e) \leftarrow \mathcal{A}(\mathtt{pp}, \mathtt{id}, V) \\ (\mathtt{ssk}_e, \mathtt{spk}_e) \leftarrow \mathtt{KeyGen}(1^\lambda) \\ S' \leftarrow \mathcal{A}(B', k) \\ \mathtt{ctx} \leftarrow \mathtt{encap}(\mathtt{ck}_s, \mathtt{cpk}_t) \\ t' \leftarrow \mathcal{A}(S', \mathtt{spk}, \mathtt{spk}_e, \mathtt{ctx}) \end{array}\right] \leq \ell_{\mathcal{A}} f^{-1} + \mathsf{negl}(\lambda).$$

Definition 11. (Server-Signer Privacy) We say that a BAKE protocol has *server-signer privacy* if an adversary $\mathcal{A}$ controlling both the computation server and the authentication server does not have an advantage in obtaining a biometric

Enrolment

Setup party		**Server**		**Signer**
t reference template		$\mathtt{ssk} \in \mathcal{K}$		$k \in \mathcal{K}$
		$\mathtt{spk} \in \mathcal{P}$		
$(f, V) \leftarrow \mathtt{lock}(t)$				
$(B, r) \leftarrow \mathtt{blind}(f)$				
	$\xrightarrow{B}$		$\xrightarrow{B}$	
				$S \leftarrow \mathtt{sign}(B, k)$
	$\xleftarrow{S}$		$\xleftarrow{S}$	
$\mathtt{csk}_t \leftarrow \mathtt{unblind}(S, r)$				
$\mathtt{cpk}_t \leftarrow \mathtt{pkGen}(\mathtt{csk}_t)$				
	$\xrightarrow{V, \mathtt{cpk}_t, \mathtt{id}}$			
		store		
		$(V, \mathtt{cpk}_t, \mathtt{id})$		

Figure 2. BAKE enrolment protocol.

Verification

Client		**Server**		**Signer**
t' probe feature vector		$\mathtt{ssk} \in \mathcal{K}$		$k \in \mathcal{K}$
$\mathtt{spk} \in \mathcal{P}$		$\mathtt{spk} \in \mathcal{P}$		
		$(V, \mathtt{cpk}_t, \mathtt{id})$		
	$\xrightarrow{\mathtt{id}}$			
	$\xleftarrow{V}$			
$f' \leftarrow \mathtt{unlock}(V, t')$				
$(B', r') \leftarrow \mathtt{blind}(f')$				
$(\mathtt{csk}_e, \mathtt{cpk}_e) \leftarrow \mathtt{KeyGen}(1^\lambda)$		$(\mathtt{ssk}_e, \mathtt{spk}_e) \leftarrow \mathtt{KeyGen}(1^\lambda)$		
	$\xrightarrow{B', \mathtt{cpk}_e}$		$\xrightarrow{B'}$	
		$\mathtt{ctx} \leftarrow \mathtt{encap}(\mathtt{ck}_s, \mathtt{cpk}_t)$		$S' \leftarrow \mathtt{sign}(B', k)$
	$\xleftarrow{S', \mathtt{spk}_e, \mathtt{ctx}}$		$\xleftarrow{S'}$	
$\mathtt{csk}_{t'} \leftarrow \mathtt{unblind}(S', r')$				
$\mathtt{cpk}_{t'} \leftarrow \mathtt{pkGen}(\mathtt{csk}_{t'})$				
$\mathtt{ck}_c \leftarrow \mathtt{decap}(\mathtt{ctx}, \mathtt{csk}_{t'})$				
return $H(\mathtt{ck}_c) = H(\mathtt{ck}_s)$				

Figure 3. BAKE verification protocol.

feature vector t' that is close to any enrolled biometric template t above running a brute-force search on V.

$$\Pr\left[\mathtt{dist}(t, t') < \tau \ : \ \begin{array}{c} \mathsf{pp} \leftarrow \mathtt{setup}(1^\lambda) \\ \{V, \mathtt{cpk}_t\} \leftarrow \mathtt{enroll}(\mathsf{pp}, t) \\ f' \leftarrow \mathtt{unlock}(V, t') \\ B' \leftarrow \mathtt{blind}(f') \\ (\mathtt{csk}_e, \mathtt{cpk}_e) \leftarrow \mathtt{KeyGen}(1^\lambda) \\ t' \leftarrow \mathcal{A}(\mathsf{pp}, \mathtt{id}, V, B', k, \mathtt{cpk}_t, \mathtt{cpk}_e) \end{array}\right] \leq \ell_{\mathcal{A}} f^{-1} + \mathsf{negl}(\lambda).$$

3.6. Instantiation Based on Discrete Logarithms

In this section, we give an instantiation of the protocol defined in Figures 2 and 3 using cryptographic primitives that build on the security of discrete logarithms (DL). Concretely, we instantiate the universe $\mathcal{P}$ with a cyclic group $\mathbb{G}$, which can be the group of points on an elliptic curve, and the key space $\mathcal{K}$ and randomness space $\mathcal{R}$ with a scalar field $\mathbb{Z}_q$, where q is the order of $\mathbb{G}$. Further, we also define two hash functions $H_{\mathbb{G}} : \{0,1\}^* \rightarrow \mathbb{G}$ and $H_{\mathbb{Z}_q} : \{0,1\}^* \rightarrow \mathbb{Z}_q$.

Building on these foundations, the respective algorithms of Definition 6 are instantiated with the Hash-DH OPRF defined in Definition 4 and ephemeral Diffie-Hellman key exchange with a key-derivation function `KDF`. The detailed protocols for enrolment and verification are defined in Figures 4 and 5, respectively. In the following, we refer to the verification protocol in Figure 5 as DL-BAKE.

3.7. Security Proofs

In this section, we provide theorems stating the security of the protocols above based on the hardness of discrete logarithms, and we sketch the security proofs.

Theorem 1 (Correctness)**.** Assume that a probe sample t' is within the verification threshold τ compared to a biometric template $t_{\mathtt{id}}$ for some registered identity $\mathtt{id}$. Then the DL-BAKE protocol in Figure 5 is correct.

Proof sketch. This follows directly from the construction. If the probe sample t' is within the verification threshold τ compared to a biometric template $t_{\mathtt{id}}$ for some registered identity $\mathtt{id}$, then the client will successfully reconstruct the correct polynomial f' using interpolation. From the correctness of the OPRF, the KEM and the KDF, we then conclude that the client and the server compute the same values, and the data subject is correctly authorised. □

Theorem 2 (Client Privacy)**.** Let $\mathcal{A}_0$ be an adversary against *client privacy* in the DL-BAKE protocol in Figure 5 with advantage ϵ_0. Then there exists an adversary $\mathcal{A}_1$ against the fuzzy vault V with advantage ϵ_1 and an adversary $\mathcal{A}_2$ against the OPRF with advantage ϵ_2, such that $\epsilon_0 \leq \epsilon_1 + f^{-1}(1+\epsilon_2)$. The runtime of $\mathcal{A}_0$ is essentially the same as of $\mathcal{A}_1$ and $\mathcal{A}_2$.

Proof sketch. We consider a single log-in attempt by an adversary $\mathcal{A}_0$ controlling the client. If $\mathcal{A}_0$ guesses a biometric probe, the probability that this probe is close to the reference sample is approximately f^{-1}. Furthermore, if $\mathcal{A}_0$ with probability ϵ_0 can output a valid probe sample t' given access to the fuzzy vault V, we can trivially turn $\mathcal{A}_0$ into an adversary $\mathcal{A}_1$ against V with the same advantage. Moreover, if $\mathcal{A}_0$ with advantage f^{-1} can output a valid probe sample t' when having access to values signed with key k, then we can turn $\mathcal{A}_0$ into an adversary $\mathcal{A}_2$ against the OPRF. Finally, we observe that the KEM are independent of $t_{\mathtt{id}}$, and hence, an adversary $\mathcal{A}_0$ cannot learn anything from interacting with this protocol. We conclude that the protocol achieves client privacy. □

Theorem 3 (Server privacy)**.** Let $\mathcal{A}_0$ be an adversary against server privacy the DL-BAKE protocol in Figure 5 with advantage ϵ_0. Then there exists an adversary $\mathcal{A}_1$ against the fuzzy vault V with advantage ϵ_1 and an adversary $\mathcal{A}_2$ against the OPRF with advantage ϵ_2, such that $\epsilon_0 \leq \epsilon_1 + f^{-1}(1+\epsilon_2)$. The runtime of $\mathcal{A}_0$ is essentially the same as of $\mathcal{A}_1$ and $\mathcal{A}_2$.

We omit the proof of Theorem 3 since it is similar to Theorem 2.

Theorem 4 (Client-Signer Privacy)**.** Let $\mathcal{A}_0$ be an adversary against client-signer privacy in the DL-BAKE protocol in Figure 5 with advantage ϵ_0 controlling both the client and the signer. Then $\epsilon_0 \leq f^{-1}$ and $\mathcal{A}_0$ has no advantage in guessing a biometric probe within the threshold of an enrolled template above a brute-force search.

Proof sketch. We consider a colluding malicious client and malicious signer. Assume that $\mathcal{A}_0$ runs the verification protocol once on any input probe t' and receives $(S', \mathtt{spk}_e, H(\mathtt{ck}_s))$ from the server. Then $\mathcal{A}_0$ can guess a biometric probe, interpolate to get a polynomial f' and execute the OPRF on input f' using the signer's key k. For each guess, $\mathcal{A}_0$ can check if the KDF output corresponds to $H(\mathtt{ck}_s)$. No information about any enrolled template $t_{\mathtt{id}}$ is encoded in the messages from the server. □

Theorem 5 (Server-Signer Privacy)**.** Let $\mathcal{A}_0$ be an adversary against server-signer privacy in the DL-BAKE protocol in Figure 5 with advantage ϵ_0 controlling both the server and the signer. Then $\epsilon_0 \leq f^{-1}$ and $\mathcal{A}_0$ has no advantage in guessing a biometric template within the threshold of an enrolled template above a brute-force search.

Proof sketch. We consider a colluding malicious server and malicious signer. Then $\mathcal{A}_0$ can guess a biometric probe, interpolate to get a polynomial f' and execute the OPRF on input f' using the signer's key k. For each guess, $\mathcal{A}_0$ can check if $[H_{\mathbb{Z}_q}(B')]G = \mathtt{cpk}_r$. No information about any enrolled template $t_{\mathtt{id}}$ is encoded in the messages from the client. □

3.8. Improved Security using NIZK

The protocol can be further secured by the addition of non-interactive zero-knowledge proofs (NIZK). Namely, we show how to apply a standard Chaum-Pederson zero-knowledge proof [46] using a Fiat-Shamir transform [47] yielding a non-interactive proof.

The NIZK is added to prove the honest evaluation of the OPRF. Thereby, a client can verify that the signer computed the signature honestly. In the case of an unsuccessful authentication attempt, the client therefore gains more knowledge about the reason of failure, and can potentially reveal a corrupted signer. We note that above this additional information, the passively secure protocol already allows for the protection of the biometric data even in the presence of malicious adversaries, as long as at least one of the parties remains honest as given by the security definitions above. A detailed verification protocol with the addition of NIZK is given in Appendix A.

3.9. Instantiation Based on Lattices

Our protocol can also be instantiated with lattice-based cryptographic primitives, which are assumed to yield post-quantum security for correct parameter choices [48]. Two components in the protocol need to be instantiated: the OPRF and the KEM.

A construction of lattice-based OPRFs has recently been proposed by [13], which builds on the security of the Ring Learning-with-Errors (R-LWE) problem [49], and we give an intuition for how this construction can be embedded in our protocol. The authors of [13] base their OPRF on a PRF using a gadget matrix G^{-1}, the discrete logarithm equivalent

Enrolment (DL instantiation)

Setup party		**Server**		**Signer**
t reference template		$\mathtt{ssk} \in \mathbb{Z}_q$		$k \in \mathbb{Z}_q$
		$\mathtt{spk} \in \mathbb{G}$		
$f \leftarrow \mathbb{F}_p[x] : \deg(f) = \tau + 1$				
$V(x) = f(x) + \prod_{a \in t}(x - a)$				
$x \leftarrow f(0)$				
$r \leftarrow\$ \mathbb{Z}_q$				
$B \leftarrow [r]H_{\mathbb{G}}(x)$				
	$\xrightarrow{B}$		$\xrightarrow{B}$	
				$S = [k]B$
	$\xleftarrow{S}$		$\xleftarrow{S}$	
$U \leftarrow [r^{-1}]S = [k]H_{\mathbb{G}}(x)$				
$\mathtt{csk}_t \leftarrow H_{\mathbb{Z}_q}(U)$				
$\mathtt{cpk}_t \leftarrow [\mathtt{csk}_t]G$				
	$\xrightarrow{V, \mathtt{cpk}_t, \mathtt{id}}$			
		store $(V, \mathtt{cpk}_t, \mathtt{id})$		

Figure 4. BAKE enrolment protocol instantiated with discrete-logarithm OPRF and Diffie-Hellman key exchange.

of which can be thought of as a product of group generators, where a generator is included in the product if the input bit is true, and omitted else, see [50] for more details. This specific PRF construction is put in place to enable verifiability and security against active adversaries.

However, the zero-knowledge proofs appended to the lattice-based PRF for active security are not practical for real-life situations, with proof sizes of several gigabytes [13]. Therefore, we only look at the case of passive security for the lattice instantiation, which can be significantly simplified by replacing the PRF with a hash function.

Then, the OPRF can be executed as given in the paper, continuing to omit all zero-knowledge proofs. The hashed input is blinded by building an R-LWE sample from it, and sent to the signer to obtain a signature with a secret OPRF key k. The signer computes the signature as another R-LWE sample, and the signed input can be recovered by the client by subtracting a public commitment to k and rounding.

Finally, the Diffie-Hellman key exchange can easily be replaced with a lattice-based KEM, e.g., the recently standardised CRYSTALS-Kyber [14]. Then, the server encapsulates a common key using the client's stored reference public key. The client can only decapsulate the key if her secret key constructed from the probe feature vector aligns with the public key previously stored at the server, i.e., if and only if the biometric inputs were found to be a mated comparison trial.

4. Experimental Evaluation

We evaluated our protocol instantiated with elliptic curves presented in Figure 5 experimentally and show the results in this section. Our experiments were run on a commodity notebook with Intel Core i7-8565U CPU@1.80GHz and 8GB RAM. Our code is available at https://anonymous.4open.science/r/bake-7057 and includes automated installation scripts with all dependencies for easy reproducibility.

For the fingerprint fuzzy vault instantiation, we used the open-source implementation provided by [15] with all original parameter settings. This ensures perfect replaceability with other state-of-the-art fuzzy vault instantiations, such as [33] for iris and [34] for face. In particular, we run our implementation on the same fingerprint database MCYT-330 [27] and same feature extractor, Digital Persona's FingerJetFX open source edition minutiae extractor[1]. This means that all evaluations of biometric performance can be compared directly to the original paper of [15] and papers that compare their work with the latter [33], [34].

The only modification applied to the implementation of [15] is in the unlocking function. Here, [15] use the stored hash $H(f)$ of the secret polynomial f corresponding to a reference template t, which allows for offline brute force attacks. Our protocol prevents offline attacks by removing the hash and using highest-frequency decoding in its place (see Section 3.3). As discussed above, this does not impact the security in terms of the false-match rate of our protocol.

Our implementation of the OPRF and Diffie-Hellman key exchange is based on OpenSSL. For all cryptographic operations, we used P-256 [51] as the elliptic curve and SHA-256 as the hash function.

To begin, we give a more detailed comparison of our work with closely related work in Table 3 by extending

1. http://www.digitalpersona.com/fingerjetfx

Verification (DL instantiation)

Client		**Server**		**Signer**
t' probe feature vector		$\texttt{ssk} \in \mathbb{Z}_q$		$k \in \mathbb{Z}_q$
$\texttt{spk} \in \mathbb{G}$		$\texttt{spk} \in \mathbb{G}$		
		$(V, \texttt{cpk}_t, \texttt{id})$		
	$\xrightarrow{\texttt{id}}$			
	$\xleftarrow{V}$			
find $\{(b, y(b)) : b \in t'\}$ and decode to $f' \in \mathbb{F}_p[x]$ $x' \leftarrow f'(0)$ $r' \leftarrow\$ \mathbb{Z}_q$ $B' \leftarrow [r']H_{\mathbb{G}}(x')$				
$\texttt{csk}_e \leftarrow\$ \mathbb{Z}_q$ $\texttt{cpk}_e \leftarrow [\texttt{csk}_e]G$		$\texttt{ssk}_e \leftarrow\$ \mathbb{Z}_q$ $\texttt{spk}_e \leftarrow [\texttt{ssk}_e]G$		
	$\xrightarrow{B', \texttt{cpk}_e}$		$\xrightarrow{B'}$	
		$\texttt{ck}_s \leftarrow \texttt{KDF}([\texttt{ssk}_e]\texttt{cpk}_e,$ $[\texttt{ssk}]\texttt{cpk}_e, [\texttt{ssk}_e]\texttt{cpk}_t,$ $\texttt{cpk}_e, \texttt{spk}_e, \texttt{cpk}_t, \texttt{spk})$		$S' = [k]B'$
	$\xleftarrow{S', \texttt{spk}, \texttt{spk}_e, H(\texttt{ck}_s)}$		$\xleftarrow{S'}$	
$B' \leftarrow [r'^{-1}]S' = [k]H_{\mathbb{G}}(x')$ $\texttt{csk}_{t'} \leftarrow H_{\mathbb{Z}_q}(B')$ $\texttt{cpk}_{t'} \leftarrow [\texttt{csk}_{t'}]G$ $\texttt{ck}_c \leftarrow \texttt{KDF}([\texttt{csk}_e]\texttt{spk}_e,$ $[\texttt{csk}_e]\texttt{spk}, [\texttt{csk}_{t'}]\texttt{spk}_e,$ $\texttt{cpk}_e, \texttt{spk}_e, \texttt{cpk}_{t'}, \texttt{spk})$ return $H(\texttt{ck}_c) = H(\texttt{ck}_s)$				

Figure 5. BAKE verification protocol instantiated with discrete-logarithm OPRF and Diffie-Hellman key exchange.

TABLE 3. TABLE 1 IN [7] UPDATED WITH OUR PROTOCOL.

Scheme	Technique	Rounds	Compatibility	ISO/IEC 24745 [6]
fPAKE-1 [9]	Garbled Circuits	5	iris, fixed-length fingerprint	✗
fPAKE-2 [9]	PAKE + Secret Sharing	2		✗
fuzzy aPAKE-1 [10]	Secret Sharing + OT	2		✗
fuzzy aPAKE-2 [10]	aPAKE	2		✗
BAKE-1 [7]	Random Linear Codes	1	minutiae-based fingerprint	✗
BAKE-2 [7]	Secret Sharing + Polynomial Interpolation	1	iris	✗
DL-BAKE (ours)	Fuzzy Vault + OPRFs	2	minutiae-based fingerprint, iris, face	✓

Table 1 in [7] with our protocol. In terms of round efficiency, our protocol compares well to [9] and [10] with two rounds of communication, whereas [7] constructed a one-round protocol. In terms of the protection of the biometric data compliant with ISO/IEC 24745 [6], our protocol is the only compliant one: we inherit unlinkability, renewability and irreversibility from the fuzzy vault schemes. Moreover, we show that our protocol is efficient in terms of execution times in Table 4 and Figure 7 as well as in terms of biometric performance shown in Figure 6. In comparison, fPAKE [9] does not achieve irreversibility as templates are disclosed to the server in cleartext, fuzzy aPAKE [10] does not achieve computational efficiency, and [7] do not achieve an acceptable biometric performance, as we show in Appendix B.

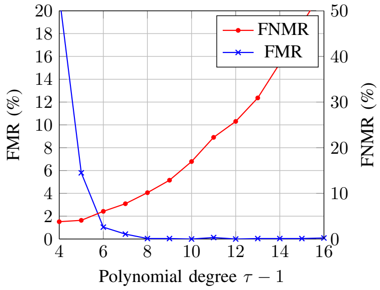


Figure 6. Biometric performance for the DL-BAKE protocol instantiated with fingerprint fuzzy vault [15].

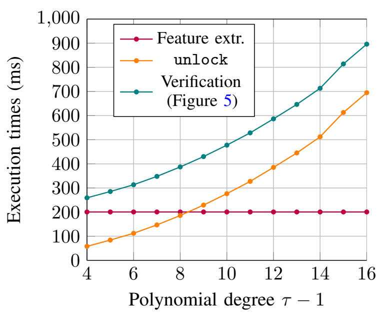


Figure 7. Execution times in milliseconds for the DL-BAKE protocol instantiated with fingerprint fuzzy vault [15].

TABLE 4. EXECUTION TIMES IN MILLISECONDS FOR THE DL-BAKE PROTOCOL INSTANTIATED WITH FINGERPRINT FUZZY VAULT [15].

	Polynomial degree $\tau - 1$					
	6	8	10	12	14	16
Feature extraction and preprocessing	200.59					
`lock`	2.38					
`unlock`	112.24	185.99	276.37	385.26	511.91	694.87
`OPRF`	0.21					
`KeyGen`, `pkGen`	0.05					
`encap`	0.16					
`decap`	0.15					
Verification (Figure 5)	313.4	387.15	477.53	586.42	713.07	896.03
FMR (%)	1.04%	0.04%	0.00%	0.00%	0.04%	0.09%
1− FNMR (%)	92.88%	88.79%	81.97%	73.18%	60.45%	44.09%
Estimated security in bits [15]	17	23	29	36	44	—

Regarding the computational performance and recognition accuracy of our protocol, we give timings for increasing polynomial degrees $\tau - 1$ in Table 4, where τ is the biometric decision threshold. At the same time, we give the biometric performance in FMR and FNMR along with the estimated false-accept security in bits as evaluated in [15]. As these security levels are derived from the FMR and our modified unlocking function does not impact the FMR, we are able to refer to the evaluation performed in [15] directly. For an acceptable recognition accuracy at $\tau - 1 = 8$, the execution of the protocol DL-BAKE given in Figure 5 takes 387.15 milliseconds. To compare, the fastest setting reported in Table 2 in [7] also achieves 387 milliseconds, but at significantly lower accuracy (see Appendix B).

The execution times are dominated by the constant cost of feature extraction (200.59 milliseconds) and the cost for unlocking, which is dependent on the polynomial degree. Figure 7 visualises these dominating costs. We note that timing for the enrolment part of the protocol given in Figure 4 is 203.23 milliseconds, where feature extraction dominates compared to the locking at 2.38 milliseconds. However, the enrolment step is a one-time effort of the setup of the system, and does not affect verification performance.

Accordingly, Figure 6 shows the trade-off between FMR and FNMR for our protocol. To conclude the efficiency evaluation of our protocol, we report that the communication cost of objects transferred between the parties during the verification step of the protocol is 65 bytes for any point on the elliptic curve (i.e., $\mathtt{cpk}_e, \mathtt{spk}_e, B'$ and S'), 152 bytes for a locked fuzzy vault for the best parameter choice of $\tau - 1 = 8$, and 32 bytes for the hash digest.

5. Conclusion

In this work, we constructed secure biometrics-authenticated key exchange from fuzzy vaults and proved its security in compliance with ISO/IEC 24745. Our BAKE protocol is efficient both in terms of execution times and biometric performance.

The combination of asymmetric, secure and efficient biometrics-authenticated key exchange has not been achieved in prior works. Related protocols are either symmetric, and thus does not provide protection of the biometric data on the server side, or inefficient in terms of computational speed due to their generality, or else insufficient in terms of recognition accuracy, allowing for zero-effort imposter attacks and low-effort brute-force security. The accuracy deficiencies of the latter can not be addresses by exchanging the biometric comparison subsystem, as the construction is specific to the imprecise comparator used.

In our protocol, we enforce communication for every adversarial guess through OPRFs. Using established and interchangable fuzzy vault schemes for different biometric modalities, the encoded secret polynomial is input to the OPRF, yielding a derived client keypair. During the key exchange, the server uses the stored keypair generated during the enrolment process, and the client uses a freshly extracted and signed keypair. Thereby, the key exchange is only successful if the two biometric samples were close.

Furthermore, we show that our protocol can be instantiated both with classical primitives, namely discrete logarithm based OPRFs and Diffie-Hellman key exchange, as well as with lattice-based OPRFs and KEMs.

Future work includes addressing the necessary pre-alignment processes of minutiae-based fingerprint representations. A promising approach both with regard to rotation and entropy is the use of 4-finger captures, where four fingerprints are captured within one image. Through the relative position of the fingers, pre-alignment can be realised more efficiently than based on minutiae, and the intra-identity independence of fingerprint patterns yield the fourfold entropy of the biometric data. Notably, the implementation of the minutiae fuzzy vault evaluated in our work includes the option of combining four fingerprints into one fuzzy vault. However, auxiliary alignment data required for pre-alignment are not yet discussed in this context.

References

[1] R. Kessler, O. Henninger, and C. Busch, "Fingerprints, forever young?" in *Proc. Intl. Conf. on Pattern Recognition (ICPR)*, 2021, pp. 8647–8654.

[2] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, "Fingerprint image reconstruction from standard templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, September 2007.

[3] J. Galbally, A. Ross, M. Gomez-Barrero, J. Fierrez, and J. Ortega-Garcia, "Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms," *Computer Vision and Image Understanding*, vol. 117, no. 10, pp. 1512–1525, 2013.

[4] G. Mai, K. Cao, P. C. Yuen, and A. K. Jain, "On the Reconstruction of Face images from Deep Face Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1188–1202, 2018.

[5] European Parliament, *EU Regulation 2016/679 of the European Parliament and of the Council (General Data Protection Regulation)*, 2016.

[6] ISO/IEC JTC1 SC27 Security Techniques, *ISO/IEC 24745:2022. Information Technology - Security Techniques - Biometric Information Protection*, International Organization for Standardization, 2022.

[7] M. Wang, K. He, J. Chen, Z. Li, W. Zhao, and R. Du, "Biometrics-authenticated key exchange for secure messaging," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 2618–2631.

[8] S. Jarecki, H. Krawczyk, and J. Xu, "OPAQUE: An asymmetric PAKE protocol secure against pre-computation attacks," in *EUROCRYPT 2018, Part III*, ser. LNCS, J. B. Nielsen and V. Rijmen, Eds., vol. 10822. Springer, Heidelberg, Apr. / May 2018, pp. 456–486.

[9] P.-A. Dupont, J. Hesse, D. Pointcheval, L. Reyzin, and S. Yakoubov, "Fuzzy password-authenticated key exchange," in *EUROCRYPT 2018, Part III*, ser. LNCS, J. B. Nielsen and V. Rijmen, Eds., vol. 10822. Springer, Heidelberg, Apr. / May 2018, pp. 393–424.

[10] A. Erwig, J. Hesse, M. Orlt, and S. Riahi, "Fuzzy asymmetric password-authenticated key exchange," in *ASIACRYPT 2020, Part II*, ser. LNCS, S. Moriai and H. Wang, Eds., vol. 12492. Springer, Heidelberg, Dec. 2020, pp. 761–784.

[11] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237–257, 2006.

[12] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, November 1976.

[13] M. R. Albrecht, A. Davidson, A. Deo, and N. P. Smart, "Round-optimal verifiable oblivious pseudorandom functions from ideal lattices," in *PKC 2021, Part II*, ser. LNCS, J. Garay, Ed., vol. 12711. Springer, Heidelberg, May 2021, pp. 261–289.

[14] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-kyber: a cca-secure module-lattice-based kem," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2018, pp. 353–367.

[15] B. Tams, "Unlinkable minutiae-based fuzzy vault for multiple fingerprints," *Iet Biometrics*, vol. 5, no. 3, pp. 170–180, 2016.

[16] X. Jiang and W.-Y. Yau, "Fingerprint minutiae matching based on the local and global structures," in *Proceedings 15th international conference on pattern recognition. ICPR-2000*, vol. 2. IEEE, 2000, pp. 1038–1041.

[17] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: a new representation and matching technique for fingerprint recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, March 2010.

[18] J. Daugman, "How iris recognition works," *IEEE Trans. on Circuits and Systems for Video Technology (TCSVT)*, vol. 14, no. 1, pp. 21–30, 2004.

[19] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2004, pp. 523–540.

[20] M. Qi, J. Chen, and Y. Chen, "A secure biometrics-based authentication key exchange protocol for multi-server tmis using ecc," *Computer methods and programs in biomedicine*, vol. 164, pp. 101–109, 2018.

[21] A. Sarkar and B. K. Singh, "A novel session key generation and secure communication establishment protocol using fingerprint biometrics," in *Handbook of Computer Networks and Cyber Security*. Springer, 2020, pp. 777–805.

[22] ISO/IEC JTC1 SC37 Biometrics, *ISO/IEC 19794-1:2011 Information Technology - Biometric Data Interchange Formats - Part 1: Framework*, International Organization for Standardization, June 2011.

[23] ——, *ISO/IEC 19794-2:2011 Information Technology - Biometric Data Interchange Formats - Part 2: Finger Minutiae Data*, International Organization for Standardization, June 2011.

[24] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Trans. on Image Processing*, vol. 9, no. 5, pp. 846–859, 2000.

[25] H. Xu, R. Veldhuis, T. Kevenaar, A. Akkermans, and A. Bazen, "Spectral minutiae: A fixed-length representation of a minutiae set," in *2008 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops*, June 2008.

[26] G. Li, C. Busch, and B. Yang, "A novel approach used for measuring fingerprint orientation of arch fingerprint," in *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2014, pp. 1309–1314.

[27] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy *et al.*, "MCYT baseline corpus: a bimodal biometric database," *Vision, Image and Signal Processing, IEEE Proc.*, vol. 150, no. 6, pp. 395–401, December 2003.

[28] R. Važan, "Sourceafis fingerprint recognition toolkit," *https://sourceafis.machinezoo.com*, 2018.

[29] ISO/IEC JTC1 SC37 Biometrics, *ISO/IEC 19795-1:2021. Information Technology – Biometric Performance Testing and Reporting – Part 1: Principles and Framework*, International Organization for Standardization, June 2021.

[30] FRONTEX, "Best practice technical guidelines for automated border control ABC systems," 2015.

[31] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcardbased fingerprint authentication," in *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications*, 2003, pp. 45–52.

[32] B. Tams, "Decodability attack against the fuzzy commitment scheme with public feature transforms," *arXiv preprint arXiv:1406.1154*, 2014.

[33] C. Rathgeb, B. Tams, J. Wagner, and C. Busch, "Unlinkable improved multi-biometric iris fuzzy vault," *EURASIP Journal on Information Security*, vol. 2016, no. 1, pp. 1–16, 2016.

[34] C. Rathgeb, J. Merkle, J. Scholz, B. Tams, and V. Nesterowicz, "Deep face fuzzy vault: Implementation and performance," *Computers & Security*, vol. 113, p. 102539, 2022.

[35] A. Adler, R. Youmaran, and S. Loyka, "Towards a measure of biometric information," February 2006.

[36] N. K. Ratha, J. H. Connell, and R. M. Bolle, "An analysis of minutiae matching strength," in *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer, 2001, pp. 223–228.

[37] J. Daugman, "Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1927–1935, 2006.

[38] S. Casacuberta, J. Hesse, and A. Lehmann, "Sok: Oblivious pseudorandom functions," *Cryptology ePrint Archive*, 2022.

[39] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword search and oblivious pseudorandom functions," in *Theory of Cryptography Conference*. Springer, 2005, pp. 303–324.

[40] W. Ford and B. S. Kaliski, "Server-assisted generation of a strong secret from a password," in *Proceedings IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2000)*. IEEE, 2000, pp. 176–180.

[41] T. Okamoto, "Authenticated key exchange and key encapsulation in the standard model (invited talk)," in *ASIACRYPT 2007*, ser. LNCS, K. Kurosawa, Ed., vol. 4833. Springer, Heidelberg, Dec. 2007, pp. 474–484.

[42] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[43] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the Association for Computing Machinery*, vol. 21, no. 2, pp. 120–126, 1978.

[44] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Packed homomorphic encryption based on ideal lattices and its application to biometrics," in *International Conference on Availability, Reliability, and Security*. Springer, 2013, pp. 55–74.

[45] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon and algebraic-geometric codes," in *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*. IEEE, 1998, pp. 28–37.

[46] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *Annual international cryptology conference*. Springer, 1992, pp. 89–105.

[47] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Conference on the theory and application of cryptographic techniques*. Springer, 1986, pp. 186–194.

[48] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter *et al.*, "Homomorphic encryption standard," in *Protecting Privacy through Homomorphic Encryption*. Springer, 2021, pp. 31–62.

[49] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *EUROCRYPT 2010*, ser. LNCS, H. Gilbert, Ed., vol. 6110. Springer, Heidelberg, May / Jun. 2010, pp. 1–23.

[50] A. Banerjee, C. Peikert, and A. Rosen, "Pseudorandom functions and lattices," in *EUROCRYPT 2012*, ser. LNCS, D. Pointcheval and T. Johansson, Eds., vol. 7237. Springer, Heidelberg, Apr. 2012, pp. 719–737.

[51] E. Barker, "Digital signature standard (dss)," 2013-07-19 2013.

[52] P. R. The Biometric Systems Lab (University of Bologna), I. P. L. M. S. University), and the Biometric Test Center (San Jose State University), "Fingerprint verification competition 2004," March 2004.

Appendix A.

In this Appendix, we give a verification protocol with NIZK as discussed in Section 3.7. The protocol is given in Figure 8.

Appendix B.

In this appendix, we give the experimental evaluation of the recent work on biometrics-authenticated key exchange proposed by [7]. Specifically, we show the biometric performance of their construction for fingerprint and discuss its shortcomings.

For this evaluation, we implemented Algorithm 2 in [7] according to the description available in the paper.[2] According to the description, we set the number of neighbours for each minutia at $\mu = 4$ and, iterating through the minutiae in the template, construct the vectors $\mathrm{v}_{j,\rho}$ from the minutia's x- and y-coordinates which are given in pixels (i.e., integers) from the upper left corner. The calculation of the Euclidean distances $d_{j,1}, ..., d_{j,4}$ therefore result in floating point numbers, whereas the angles $\phi_{j,\rho,1}, ..., d_{j,\rho,6}$ remain as integer values. In Section 6.2.2 in [7], the authors state that the number of neighbours $\mu = 4$ originates an encoding of the values $d_{j,\rho}$ and $\phi_{j,\rho,\omega}$ into $\mu = 4$ bits each. This relation is not clear to us and we were not able to satisfactorily follow the reasoning given by the authors of [7] during an email exchange. Therefore, we give the evaluation of the biometric performance for the original float and integer values, which can be considered an upper bound for the performance of a binary encoding. As comparison function, we determined the set difference by mapping minutiae based on their minimal Hamming distance.

We evaluated our implementation of Algorithm 2 in [7] on the FVC2004 DB-1 [52], which is the least challenging out of the four databases used in [7] in terms of image quality and rotation of the fingerprint images. We compare the performance against a state-of-the art rotation invariant minutiae comparator, SourceAFIS [28], in Figure 9.

From the evaluation, it becomes evident that the fingerprint comparison algorithm proposed by [7] does not have an acceptable performance. For the optimal threshold, the

2. We are happy to provide our Python code on request. In an email exchange with the authors of [7] however, they stated that they could not share their implementation due to licensing of a sponsor.

Verification (DL instantiation with NIZK)

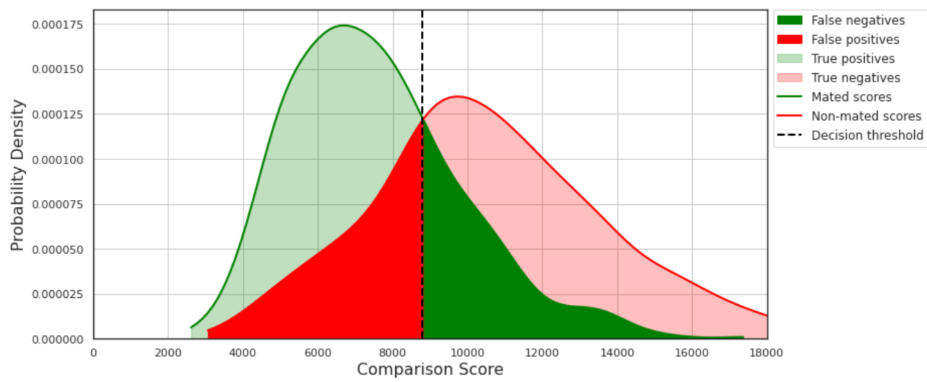
Client		**Server**		**Signer**
t' probe template		$\mathtt{ssk} \in \mathbb{Z}_q$		$k \in \mathbb{Z}_q$
$\mathtt{spk} \in \mathbb{G}$		$\mathtt{spk} \in \mathbb{G}$		
		$(V, \mathtt{cpk}_t, \mathtt{id})$		
	$\xrightarrow{\mathtt{id}}$			
	$\xleftarrow{V}$			
find $= \{(b, V(b)) : b \in t'\}$				
and decode to $f' \in \mathbb{F}_p[X]$				
$x' \leftarrow f'(0)$				
$r' \leftarrow\$ \mathbb{Z}_q$				
$B' \leftarrow [r']H_{\mathbb{G}}(x')$				
$\mathtt{csk}_e \leftarrow\$ \mathbb{Z}_q$		$\mathtt{ssk}_e \leftarrow\$ \mathbb{Z}_q$		
$\mathtt{cpk}_e \leftarrow [\mathtt{csk}_e]G$		$\mathtt{spk}_e \leftarrow [\mathtt{ssk}_e]G$		
	$\xrightarrow{B', \mathtt{cpk}_e}$		$\xrightarrow{B'}$	
		$\mathtt{ck}_s \leftarrow \mathtt{KDF}([\mathtt{ssk}_e]\mathtt{cpk}_e,$		$S' = [k]B'$
		$[\mathtt{ssk}]\mathtt{cpk}_e, [\mathtt{ssk}_e]\mathtt{cpk}_t,$		$K = [k]G$
		$\mathtt{cpk}_e, \mathtt{spk}_e, \mathtt{cpk}_t, \mathtt{spk})$		$w \leftarrow\$ \mathbb{Z}_q$
				$A = [w]B'$
				$D = [w]G$
				$c = H_{\mathbb{Z}_q}(B', S', K, G, A, D)$
				$z = w - ck$
	$\xleftarrow[K, c, z]{S', \mathtt{spk}, \mathtt{spk}_e, H(\mathtt{ck}_s)}$		$\xleftarrow[K, c, z]{S'}$	
$B' \leftarrow [r'^{-1}]S' = [k]H_{\mathbb{G}}(x')$				
$\mathtt{csk}_{t'} \leftarrow H_{\mathbb{Z}_q}(B')$				
$\mathtt{cpk}_{t'} \leftarrow [\mathtt{csk}_{t'}]G$				
$\mathtt{ck}_c \leftarrow \mathtt{KDF}([\mathtt{csk}_e]\mathtt{spk}_e,$				
$[\mathtt{csk}_e]\mathtt{spk}, [\mathtt{csk}_{t'}]\mathtt{spk}_e,$				
$\mathtt{cpk}_e, \mathtt{spk}_e, \mathtt{cpk}_{t'}, \mathtt{spk})$				
$A' = [z]B' + [c]S'$				
$D' = [z]G + [c]K$				
$c' = H(B', S', K, G, A', D')$				
return $(c = c', H(\mathtt{ck}_c) = H(\mathtt{ck}_s))$				

Figure 8. BAKE verification protocol with discrete-logarithm NIZK.

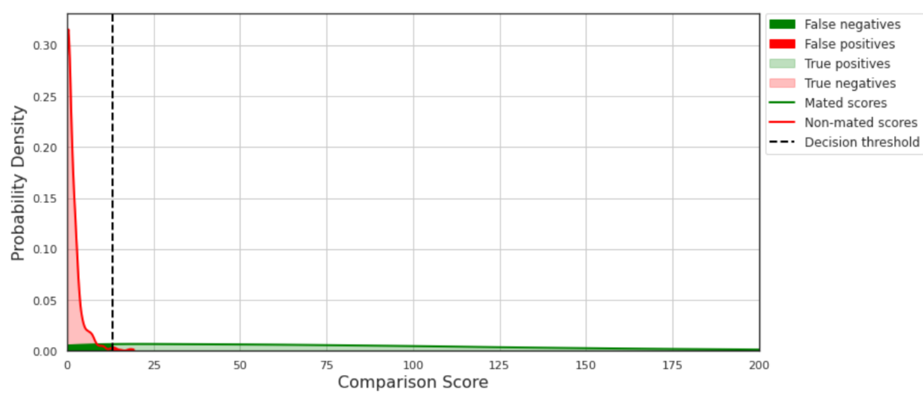
FMR is measured at 27.8% with a FNMR of 25.4%. Both of these values are not close to the required FMR of 0.1% [30] and FNMR below 5%. Compared to the state-of-the-art, the performance that can be achieved in this dataset lies at a FMR of 1.01% at FNMR of 17.29%. This shows the challenging nature of the dataset, which was collected as a fingerprint verification challenge with the goal of providing challenging fingerprint samples. Therefore, we also evaluated evaluated both algorithms on the less challenging CASIA-FPV5[3] database. However, the result are similar with a FMR of 27.6% and FNMR of 30.90% for BAKE-1 compared to a FMR of 1.13% and FNMR of 9.85% for SourceAFIS.

To conclude, the fingerprint comparison algorithm proposed for the construction in [7] is not able to distinguish between mated and non-mated comparison trials to a satisfactory degree.

3. http://biometrics.idealtest.org

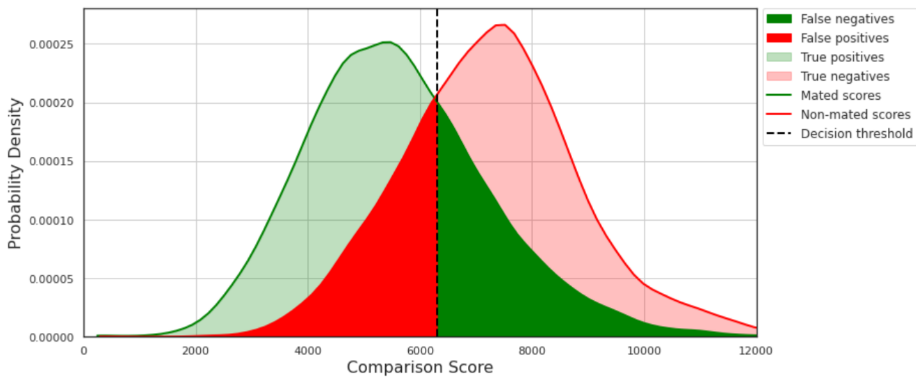


Predicted/Actual	Mated	Non-mated
Mated	74.571430	27.777780
Non-mated	25.428570	72.222220

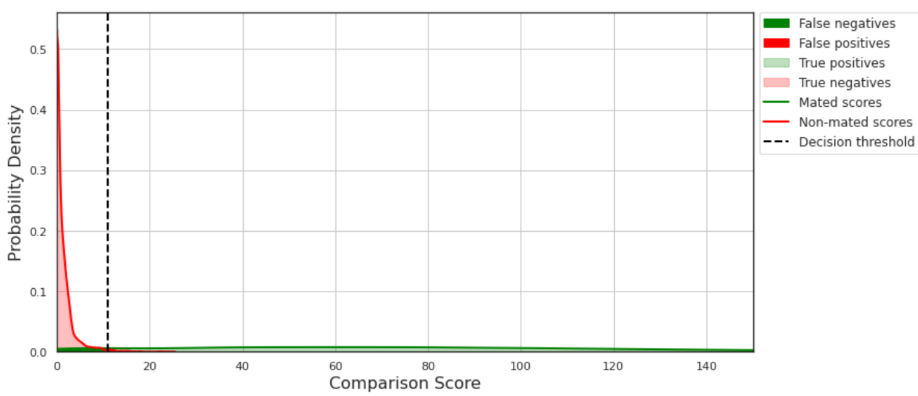


Predicted/Actual	Mated	Non-mated
Mated	82.714290	1.010100
Non-mated	17.285710	98.989900

Figure 9. Comparison score distributions for BAKE-1 [7] (top) and SourceAFIS [28] (bottom) on the FVC2004 DB-1 database [52].



Predicted/Actual	Mated	Non-mated
Mated	69.116670	27.613150
Non-mated	30.883330	72.386850



Predicted/Actual	Mated	Non-mated
Mated	90.150000	1.135600
Non-mated	9.850000	98.864400

Figure 10. Comparison score distributions for BAKE-1 [7] (top) and SourceAFIS [28] (bottom) on the CASIA-FPV5 database (http://biometrics.idealtest.org).