BRAKE: Biometric Resilient Authenticated Key Exchange

Pia Bauspieß, Tjerand Silde, Alexandre Tullot, Anamaria Costache,
Christian Rathgeb, Jascha Kolberg, Christoph Busch, *Member, IEEE*

***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 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 biometric resilient authenticated key exchange that fulfils the above requirements of biometric information protection compliant with ISO/IEC 24745. The protocol is based on established improved fuzzy vault schemes and validated with good recognition performance. We build our protocol from trusted primitives for password-authenticated key exchange using oblivious pseudo-random functions. Our protocol is independent of the biometric modality and can be implemented based on the security of discrete logarithms as well as 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 which achieves real-time efficiency 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**

I. Introduction

Biometric characteristics provide accurate and non-reputable 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 characteristic, further referred to as a biometric 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 ageing, 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 would consequently be 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

This research work has been funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE. *(Corresponding author: Pia Bauspieß)*

Pia Bauspieß and Christoph Busch are with the da/sec Biometrics and Security Research Group, Hochschule Darmstadt, 64295 Darmstadt, Germany, and with the Department of Information Security and Communication Technology, NTNU – Norwegian University of Science and Technology, 7034 Trondheim, Norway.

Tjerand Silde and Anamaria Costache are with the Department of Information Security and Communication Technology, NTNU – Norwegian University of Science and Technology, 7034 Trondheim, Norway.

Christian Rathgeb and Jascha Kolberg are with the da/sec Biometrics and Security Research Group, Hochschule Darmstadt, 64295 Darmstadt, Germany.

Alexandre Tullot is with the National Higher French Institute of Aeronautics and Space, 31055 Toulouse, France. This work was done during an internship with the da/sec Biometrics and Security Research Group, Hochschule Darmstadt, 64295 Darmstadt, Germany.

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 with classical security assumptions. 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 provide 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 (fPAKE) [9] and asymmetric fuzzy PAKE (fuzzy aPAKE) [10] have been proposed. However, with regard to biometrics, they have the following shortcomings: fPAKE [9] does not achieve protection of the biometric data, which is shared with the server in plaintext. Fuzzy aPAKE [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 simple distance function.

A. Contribution

In this work, we present a protocol for Biometric Resilient Authenticated Key Exchange (BRAKE) that addresses the deficiencies of previous works [7], [9], [10]. Our BRAKE protocol 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 on commodity hardware from the biometric capture to the completed key exchange, including communication cost. More precisely, we contribute:

- Secure biometrics-authenticated key exchange from trusted 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 a lattice-based OPRF [13] and KEM [14].
- Interchangeability of biometric modalities: our protocol can be instantiated with different fuzzy vault schemes that have been designed for 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 [15]. 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. Through our protocol's compatibility with lattice-based primitives, which are assumed to be post-quantum secure, we further achieve long-term protection of the underlying biometric data.
- 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://github.com/dasec/BRAKE. We show that our protocol achieves real-time efficiency with transaction times of under one second from the fingerprint image capture at the sensor to the completed key exchange.

B. State-of-the-Art

We briefly discuss 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 authors of [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 as long as the sample quality is high. It is worth noting that the state-of-the-art in iris recognition is based on samples captured under near-infrared light, and therefore requires designated capture devices, i.e., near-infrared sensors. Such specific sensors are however not part of most personal communications devices such as smartphones. The use of classical iris recognition in the Signal [19] messaging protocol as motivated by [7] is therefore not meaningful. In such a scenario, iris recognition in the visual spectrum would need to be considered, which is a more challenging task and provides, as of today, lower accuracy [20].

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 check if it corresponds to the public key in hand, without interacting with another party. In such an attack, the adversary does not have to guess an exact biometric feature representation, but

TABLE 1
COMPARISON OF OUR PROTOCOL TO RELATED WORK.

Scheme	Asymmetric	Efficient	Accurate	Compliant with ISO/IEC 24745
fPAKE [9]	✗	✓	✓	✗
fuzzy aPAKE [10]	✓	✗	✓	✗
BAKE [7]	✓	✓	✗	✗
BRAKE *(ours)*	✓	✓	✓	✓

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 biometric accuracy leads to a low effort in an offline search attack.

Even with assumed high biometric accuracy, offline attacks expose biometric data to high risks. Therefore, we construct our protocol such that interaction is required for every adversarial guess, which allows for rate-limiting that can be enforced as long as at least one party remains honest. The concept of enforcing interaction through a third party OPRF service in itself is not new [21]. However, the construction previously presented by [21] is neither trivially compatible with fuzzy secrets such as biometric features, nor with lattice-based primitives as our proposed protocol. In particular, no lattice-based partially OPRF as required for the protocol given in [21] is known as of today, and its construction lies outside of the scope of this work.

An overview of how our proposed scheme compares to the works discussed above can be found in Table 1. Other related works have been directed on extracting uniformly distributed cryptographic keys directly from biometric templates without running an interactive protocol [22]. 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 [23]. More recently, [24] 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 specific hardness assumptions.

C. Technical Overview

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

Improved Fuzzy Vault: The improved fuzzy vault scheme was first proposed by [22] and builds on error-correcting codes, more specifically, Reed-Solomon codes [25]. First, a random polynomial f of degree $\tau - 1$ is generated, where τ is the biometric verification threshold. Then, the elements of a biometric feature set t are encoded into a polynomial $V(x) = f(x) + \prod_{a \in t}(x - a)$. The polynomial $V(x)$ is further referred to as a locked fuzzy vault, which in the original scheme is stored on a server together with a hash $H(f)$ for further reference.

For verification, a biometric probe feature set t' is captured. From the locked fuzzy vault $V(x)$, a set of points $\{b, V(b) \mid b \in t'\}$ is computed. Using Lagrange interpolation, a polynomial f' is reconstructed from the point in this unlocking set. Again, τ refers to the biometric decision threshold, which is chosen as the correction capacity of the Reed-Solomon code. Therefore it holds that if the intersection of t and t' is larger than τ, then $H(f') = H(f)$, and the verification is successful.

It has to be emphasized that this construction is an improvement upon the original construction by [11], which has been found to be insecure against correlation attacks between different locked fuzzy vault records [15]. Instead of encoding biometric features directly into polynomials, the original scheme uses large point clouds to hide the biometric features. However, next to the large memory requirements, finding random points that hide the secret polynomial f truly is a hard problem. Therefore, improved schemes that work as introduced above have been developed by [15], [26], [27], and these are the constructions we use in our protocol.

Nevertheless, a persisting point of attack even in the improved fuzzy vault schemes is provided by the checksum $H(f)$ stored alongside the vault $V(x)$. 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, which enforces interaction during each attempted key exchange. The computation that takes place obliviously is an evaluation using a secret key k held by a third party we call the evaluator. 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 never sees any biometric information. From the OPRF evaluation, the client receives an evaluation of f without learning the evaluation key k, and the evaluator does not learn f. Based on this evaluation, 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 attempted guess of a stored biometric reference template t. 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 $\mathtt{pk}_t$, an evaluation needs to be obtained.

TABLE 2
SELECTED BIOMETRIC FEATURE REPRESENTATIONS.

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

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 a 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.

D. Structure of Paper

The rest of this paper is structured as follows: In Section II, background information and definitions required for the construction of our protocol are presented. As our main contribution, Section III presents our BRAKE protocol with security definitions and proof sketches. Section IV presents the experimental evaluation of the protocol and practical comparison with related work, before we outline our conclusions in Section V.

II. PRELIMINARIES

The framework for automated and interoperable biometric recognition has been standardised in ISO/IEC 19794-1 [28], 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.

A. Fingerprint Recognition

The representation extracted from a fingerprint sample to be used for biometric recognition are its ridge lines, which can be captured both with capacitive, optical, or contactless capture devices. 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 [29], 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 [30] 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. These functions produce dissimilarity scores, such that a verification attempt is accepted when the comparison score is below the threshold, and rejected otherwise.

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 [31] and [32]. 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 [33].

Minutiae-based comparators are more complex due to the problem of finding an accurate mapping between two unordered, noisy sets containing 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 [34]. Despite their differences and individual shortcomings, they share one common aim: a number of minutiae points higher than the pre-defined threshold need to be mapped uniquely between the two sets, such that each pairing is considered a matching minutiae pair.

B. Biometric Performance Metrics

Biometric performance testing and reporting is standardised in ISO/IEC 19795-1 [35] 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 capture subject 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 impostors were able to be verified. In most 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% [36]. 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 [36].

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 [37], [38]. Furthermore, any feature transformation such as binarisation may impact the accuracy of the system.

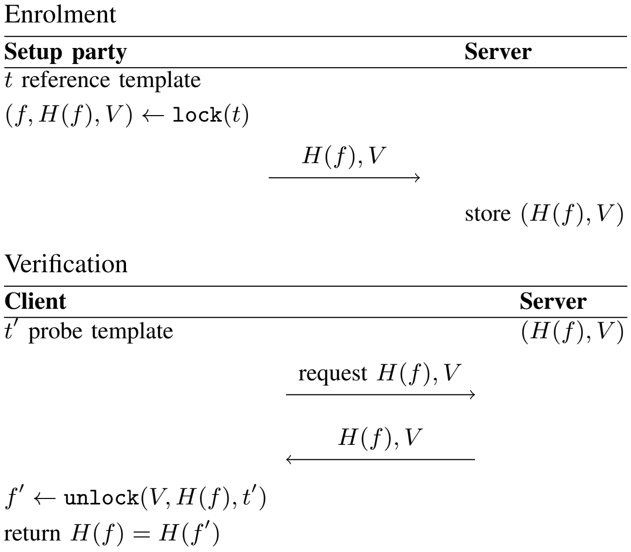
C. Fuzzy Vault

The concept of fuzzy vaults was first introduced by [11], who propose a scheme that allows to *lock* a biometric feature secret set t with a secret polynomial 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, $H : \mathcal{C} \to \{0,1\}^{2\lambda}$, for security parameter λ, be a cryptographic hash function H, and let τ a biometric comparison threshold. Then, a *fuzzy vault scheme* is a set of the following algorithms:

- $(f, H(f), V) \leftarrow \texttt{lock}(t)$: On input of a biometric feature set t, the algorithm samples a random secret $f \in \mathcal{C}$ and outputs a locked fuzzy vault V together with the hash digest $H(f)$.
- $f' \leftarrow \texttt{unlock}(V, H(f), t')$: 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 unlocking can be verified by comparing $H(f)$ to $H(f')$.

A basic authentication protocol based on the fuzzy vault scheme is given in Figure 1.


Enrolment

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

Verification

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

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

Instantiation for Fingerprint: The original schemes by [11] and a similar scheme by [39] 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 [40]. Therefore, [15] presented an improved scheme to mitigate correlation attacks (see [15], Section 1.2.3), building on the initial proposal by [22]. These improved fuzzy vault schemes fulfil 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 $(f, H(f), V) \leftarrow \texttt{lock}(t)$ be a commitment to a polynomial $f \in \mathbb{F}_p[x]$ with minutiae set t, and $f' \leftarrow \texttt{unlock}(V, H(f), t')$ 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 [26] and face [27] recognition, which we refer the reader to for full details.

D. 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 [41], a minutiae-based fingerprint representation at 82 bits

[42], and an iris at 249 bits [43]. 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 appropriate 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.

E. Cryptographic Primitives

Definition 2 (Pseudo-Random Function, [44])**.** A family of functions $f_k : \{0,1\}^\lambda \times \{0,1\}^m \to \{0,1\}^n$, with key $k \in \{0,1\}^\lambda$, are called Pseudo-Random Functions (PRFs) 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 n.

Definition 3 (Oblivious Pseudo-Random Function, [45])**.** 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, [46])**.** Let $\mathbb{G}$ be a cyclic group of prime order p, $x \in \{0,1\}^*$ the client input, $k \in \mathbb{Z}_q$ the evaluator's secret key, $H_\mathbb{G} : \{0,1\}^* \to \mathbb{G}$ and $H_{\mathbb{Z}_q} : \{0,1\}^* \to \mathbb{Z}_q$ cryptographic hash functions that output values in $\mathbb{G}$ and $\mathbb{Z}_q$, respectively. The protocol HashDH consists of the following algorithms:

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

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 evaluator learning the input x nor the output U.

Definition 5 (Key Encapsulation Mechanism, [47])**.** A Key Encapsulation Mechanism (KEM) is a scheme with three algorithms KeyGen, encap and decap, where

- $(\texttt{pk}, \texttt{sk}) \leftarrow \texttt{KeyGen}(1^\lambda)$: takes as input the security parameter λ and outputs a public key pk and a secret key sk.
- $(\texttt{ctx}, \gamma) \leftarrow \texttt{encap}(\texttt{pk})$: takes as input a public key pk, samples a session pre-key γ, and outputs γ and an encapsulation ctx of γ under the public key pk.
- $\gamma' \leftarrow \texttt{decap}(\texttt{ctx}, \texttt{sk})$: takes as input an encapsulated session pre-key ctx and a secret key sk and outputs a decapsulated session pre-key γ'.

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

A KEM can, e.g., be instantiated with (Elliptic Curve) Diffie Hellman [48], RSA [49] or CRYSTALS-Kyber [14].

III. Biometric Resilient Authenticated Key Exchange

In this section, we introduce our protocol for Biometric Resilient Authenticated Key Exchange (BRAKE) built from a fuzzy vault scheme, an OPRF, and a KEM.

A. 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 an evaluator which is in possession of a secret OPRF key. In practice, the evaluator 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 evaluator, but work under the weaker assumption that all communication between client and evaluator is seen by the server. This is a common practice in biometric information protection [50], as it allows for enhanced network security choices that protect the party handling secret key material. For example, the evaluator can be set up in a local area network that does not have to be accessible over the internet. In other applications, the separation of trust allows for scenarios where several servers may connect to an independent trusted third party service, thus increasing trust in the entire system [21].

Furthermore, we assume that authenticated channels are established between all participating parties, e.g, using standard public-key infrastructure to make server public key material accessible to the client. Depending on the context, this can also be achieved by certificate-pinning during setup.

B. Threat Model

The goal of an adversary taking control over one or more of the parties participating in the BRAKE 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) [51] can be applied. In reality, it is a hard problem to construct a presentation attack instrument, 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 evaluator, such that brute-force attempts can be detected and denied. In the setting where the evaluator enforces rate-limiting, this must be done either with respect to the server (which might have many users and, hence, the limit must be quite big) or with respect to specific users (which enforces user-specific evaluation keys). The former may allow a malicious server to get many attempts to brute-force the biometric sample of a single user before reaching the limit, while the latter requires that the user identity is sent along with the blinded value to the evaluator, and the correctness must be verified with respect to this identity.

For offline brute-force searches, a secondary attack mechanism of an adversary is to obtain the secret OPRF key held by the evaluator 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 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.

C. *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 [52]. In all three fuzzy vault schemes [15], [26], [27] discussed in our work, the algorithm of [52] 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 [39] 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. In both cases, no non-mated comparisons yield matching candidate polynomials within the list decoder threshold. Therefore, the polynomial that occurs with the highest frequency is also not a matching candidate polynomial. Consequently, the FMR is not affected by the change from hash-verified decoding to highest-frequency decoding. Instead, only changes in the FNMR or convenience of the overall system can be expected. A degradation in terms of the FNMR occurs in the case where 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 correct 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.

In addition, the frequency pattern found in a mated comparison does not give an attacker an advantage in terms of an offline-brute force attack. Through the additional roots of the randomly generated secret polynomial f, a number of seemingly correct polynomials of degree $\tau - 1$ could be interpolated by an attacker that is not in possession of a mated feature set. Therefore, a brute-force attack on a locked vault alone, without the confirmation of $H(f)$ or a successful key exchange, corresponds to a non-mated comparison attempt with no clear frequency pattern.

D. *Protocol*

In this Section, we give the formal definition of our proposed protocol for biometric resilient authenticated key exchange.

Definition 6 (Biometric Resilient Authenticated Key Exchange)**.** A three-party protocol BRAKE between a client, a server and an evaluator is a Biometric Resilient Authenticated

Key Exchange, if BRAKE privately realizes the following functionalities:

- Enrolment: A trusted setup party inputs a biometric reference template t and corresponding identifier $\texttt{id}$. The setup party computes a locked vault (f, V) based on t. The evaluator inputs a key k. Then the parties jointly compute a client public key $\texttt{cpk}_t$ derived from f. The server outputs $(V, \texttt{cpk}_t = \texttt{eval}(f, k), \texttt{id})$ and the other parties outputs nothing. The enrolment protocol is detailed in Figure 2.
- Verification: The client inputs a biometric probe feature set t' and a biometric claim $\texttt{id}$, the server inputs $(V, \texttt{cpk}_t, \texttt{id})$ and the evaluator inputs k. The client requests the locked vault V for $\texttt{id}$ and interpolates a polynomial f' from t'. The parties jointly compute a key exchange on input f'. The server outputs a session key ρ and the client outputs a session key ρ' and a bit indicating if $H(\rho) = H(\rho')$. The verification protocol is detailed in Figure 3.

Here, the client will output the bit 1 if and only if $|t \cap t'| \geq \tau$ for τ the biometric verification threshold. For the algorithms defined in Definition 6, we require the following building blocks:

Definition 7 (Building blocks)**.** We define the following building blocks for the BAKE protocol:

- $\texttt{pp} \leftarrow \texttt{setup}(1^\lambda)$: 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 defines an error-correcting code $\mathcal{C}$ with correction capacity τ. These are incorporated in the public parameters $\texttt{pp}$ and all following algorithms implicitly inherit $\texttt{pp}$.
- $(f, V) \leftarrow \texttt{lock}(t)$: 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. Note that the fuzzy vault scheme do not include the hash digest $H(f)$.
- $f' \leftarrow \texttt{unlock}(V, t')$: The algorithm takes as input a biometric probe feature vector t' and locked fuzzy vault V, and outputs an opening polynomial f'.
- $(B, r) \leftarrow \texttt{blind}(f)$: The algorithm samples a random element $r \in \mathcal{R}$ and outputs an element $B \in \mathcal{P}$.
- $S \leftarrow \texttt{eval}(B, k)$: On input $B \in \mathcal{P}$ and key $k \in \mathcal{K}$, the server outputs an evaluation $S \in \mathcal{P}$.
- $\texttt{sk} \leftarrow \texttt{unblind}(S, r)$: On input $S \in \mathcal{P}$ and $r \in \mathcal{R}$, the algorithm outputs an evaluation t U that can further be used as (or to generate) a client secret key $\texttt{csk} \in \mathcal{K}$.
- $(\texttt{sk}, \texttt{pk}) \leftarrow \texttt{KeyGen}(1^\lambda)$: The algorithm outputs a secret key $\texttt{sk} \in \mathcal{K}$ and a public key $\texttt{pk} \in \mathcal{P}$.
- $\texttt{pk} \leftarrow \texttt{pkGen}(\texttt{sk})$: The algorithm takes as input a secret key $\texttt{sk} \in \mathcal{K}$ and outputs a public key $\texttt{pk} \in \mathcal{P}$.
- $(\texttt{ctx}, \gamma) \leftarrow \texttt{encap}(\texttt{cpk})$: The algorithm takes as input a client public key $\texttt{cpk}$, samples a session pre-key γ and outputs γ and an encapsulation $\texttt{ctx}$ of γ under the public key $\texttt{cpk}$.
- $\gamma' \leftarrow \texttt{decap}(\texttt{ctx}, \texttt{csk})$: The algorithm takes as input an encapsulated session pre-key $\texttt{ctx}$ and a client secret key $\texttt{csk}$ and outputs a decapsulated session pre-key γ'.
- $\rho \leftarrow \texttt{KDF}(\texttt{cpk}, \texttt{spk}, \texttt{cpk}_e, \texttt{spk}_e, \gamma)$: The key derivation function KDF takes as input the client and server static and ephemeral public keys $\texttt{cpk}, \texttt{spk}, \texttt{cpk}_e, \texttt{spk}_e$ as well as a session pre-key γ and outputs a session key $\rho \in \{0,1\}^{2\lambda}$.

The detailed functioning of the BRAKE protocol can be seen in Figures 2 and 3. We also give a short semantic description in the following. During enrolment (Figure 2), a client public key $\texttt{cpk}_t$ is derived from a biometric reference template t and the OPRF key k, and is 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. Note that now, the fuzzy vault scheme no longer includes the hash digest $H(f)$ of the secret polynomial sampled during locking. Then, the client initiates the OPRF evaluation on input f. The evaluator evaluates the blinded input B using the OPRF key k, and the client is able to unblind and obtain its secret key $\texttt{csk}_t$, from which it computes the corresponding public key $\texttt{cpk}_t$. To conclude the enrolment step, the client sends the tuple $(V, \texttt{cpk}_t, \texttt{id})$ 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 $\texttt{id}$, 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 $(\texttt{ssk}, \texttt{spk})$ generated during setup that is not derived from any biometric information. For the key exchange, we assume that the client has access to the static server public key $\texttt{spk}$ as discussed above. Once all keys have been generated, the server encapsulates a session pre-key γ using the client's public key $\texttt{cpk}_t$. The client can decapsulate γ 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. Finally, the session key ρ is derived from γ using the client and server static and ephemeral public keys $\texttt{cpk}, \texttt{spk}, \texttt{cpk}_e, \texttt{spk}_e$ in the key derivation function KDF.

E. Security Definitions

Following the definition of the BRAKE protocol in Figures 2 and 3, we give formal definitions of the security of the protocol. For simplicity, we implicitly model the use of identifiers within the enrolment database. 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.

Notation. 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 evaluator, and let $\ell_\mathcal{A}$ be the brute-force capacity of the attacker $\mathcal{A}$.

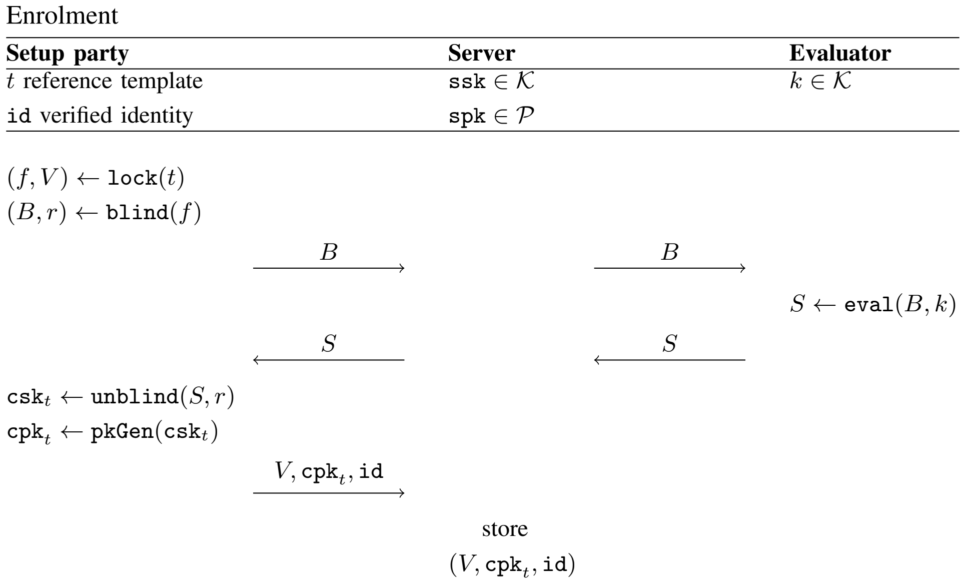

Enrolment
Setup party
Server
Evaluator
t reference template
$\texttt{ssk} \in \mathcal{K}$
$k \in \mathcal{K}$
$\texttt{id}$ verified identity
$\texttt{spk} \in \mathcal{P}$
$(f, V) \leftarrow \texttt{lock}(t)$
$(B, r) \leftarrow \texttt{blind}(f)$
$\xrightarrow{B}$
$\xrightarrow{B}$
$S \leftarrow \texttt{eval}(B, k)$
$\xleftarrow{S}$
$\xleftarrow{S}$
$\texttt{csk}_t \leftarrow \texttt{unblind}(S, r)$
$\texttt{cpk}_t \leftarrow \texttt{pkGen}(\texttt{csk}_t)$
$\xrightarrow{V, \texttt{cpk}_t, \texttt{id}}$
store
$(V, \texttt{cpk}_t, \texttt{id})$

Fig. 2. BRAKE enrolment protocol.

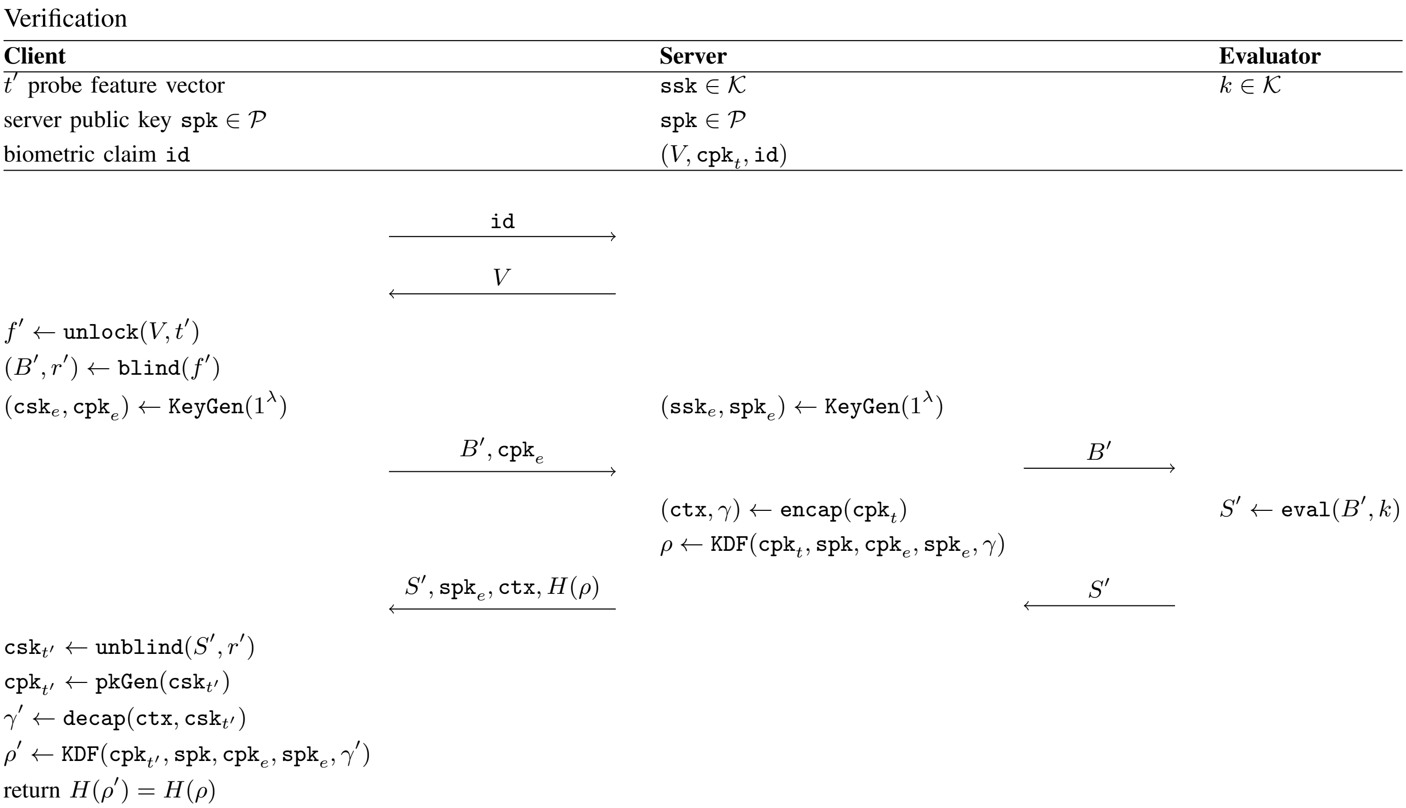

Verification
Client
Server
Evaluator
t' probe feature vector
$\texttt{ssk} \in \mathcal{K}$
$k \in \mathcal{K}$
server public key $\texttt{spk} \in \mathcal{P}$
$\texttt{spk} \in \mathcal{P}$
biometric claim $\texttt{id}$
$(V, \texttt{cpk}_t, \texttt{id})$
$\xrightarrow{\texttt{id}}$
$\xleftarrow{V}$
$f' \leftarrow \texttt{unlock}(V, t')$
$(B', r') \leftarrow \texttt{blind}(f')$
$(\texttt{csk}_e, \texttt{cpk}_e) \leftarrow \texttt{KeyGen}(1^\lambda)$
$(\texttt{ssk}_e, \texttt{spk}_e) \leftarrow \texttt{KeyGen}(1^\lambda)$
$\xrightarrow{B', \texttt{cpk}_e}$
$\xrightarrow{B'}$
$(\texttt{ctx}, \gamma) \leftarrow \texttt{encap}(\texttt{cpk}_t)$
$S' \leftarrow \texttt{eval}(B', k)$
$\rho \leftarrow \texttt{KDF}(\texttt{cpk}_t, \texttt{spk}, \texttt{cpk}_e, \texttt{spk}_e, \gamma)$
$\xleftarrow{S', \texttt{spk}_e, \texttt{ctx}, H(\rho)}$
$\xleftarrow{S'}$
$\texttt{csk}_{t'} \leftarrow \texttt{unblind}(S', r')$
$\texttt{cpk}_{t'} \leftarrow \texttt{pkGen}(\texttt{csk}_{t'})$
$\gamma' \leftarrow \texttt{decap}(\texttt{ctx}, \texttt{csk}_{t'})$
$\rho' \leftarrow \texttt{KDF}(\texttt{cpk}_{t'}, \texttt{spk}, \texttt{cpk}_e, \texttt{spk}_e, \gamma')$
return $H(\rho') = H(\rho)$

Fig. 3. BRAKE verification protocol.

Definition 8. (Correctness) We say that a `BRAKE` 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 $|t \cap t'| \geq \tau$ for a fixed biometric verification threshold τ, except with negligible probability.

Definition 9. (Client Privacy) We say that a `BRAKE` 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:

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

Definition 10. (Server Privacy) We say that a `BRAKE` 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:

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

If client and server run the protocol `BRAKE` honestly, the evaluator only sees the blinded element, which is information-theoretically secure, and hence, independent of the biometric template. We therefore do not model evaluator 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 evaluated element S'. However, as discussed above, the evaluator cannot distinguish between evaluation requests for different biometric feature vectors corresponding to mated authentication attempts, or repeated evaluation requests for a single identity aimed at running a brute-force search. Therefore, rate-limiting at the evaluator can be enforced by user-specific OPRF keys. This way, the evaluator 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 evaluator 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 an evaluation by use of the evaluation key, and compare the resulting key against the obtained one. Therefore, we realistically model an adversary controlling both the client and the evaluator 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 evaluator colluding.

Definition 11. (Client-Evaluator Privacy) We say that a `BRAKE` protocol has client-evaluator 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:

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

Definition 12. (Server-Evaluator Privacy) We say that a `BRAKE` protocol has server-evaluator privacy if an adversary $\mathcal{A}$ controlling both the server and the evaluator 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:

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

F. 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 prime order of $\mathbb{G}$. Further, we also define two hash functions $H_{\mathbb{G}} : \{0,1\}^* \to \mathbb{G}$ and $H_{\mathbb{Z}_q} : \{0,1\}^* \to \mathbb{Z}_q$.

Building on these foundations, the respective algorithms of Definition 7 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-BRAKE. We note that in the setting where the evaluator rate-limits the number of evaluations per user, the protocol can trivially be updated to send the identity of the user (or a fixed pseudonym) together with the blinded value, and the evaluator evaluates a partially oblivious PRF where the identity is a public input to the function together with the secret evaluation key. Implementing the techniques from [53], [54] allows us to perform this slightly different evaluation without (noticeable) increased computation nor communication compared to the protocol we have described.

G. 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

Enrolment (DL instantiation)

Setup party	Server	Evaluator
t reference template	$\mathtt{ssk} \in \mathbb{Z}_q$	$k \in \mathbb{Z}_q$
	$\mathtt{spk} \in \mathbb{G}$	

$f \leftarrow\!\!\$\; \mathbb{F}_p[x] : \deg(f) = \tau - 1$				
$V(x) = f(x) + \prod_{a \in t}(x - a)$				
$r \leftarrow\!\!\$\; \mathbb{Z}_q$				
$B = [r]H_{\mathbb{G}}(f)$				
	$\xrightarrow{B}$		$\xrightarrow{B}$	
				$S = [k]B$
	$\xleftarrow{S}$		$\xleftarrow{S}$	
$U = [r^{-1}]S = [k]H_{\mathbb{G}}(x)$				
$\mathtt{csk}_t \leftarrow H_{\mathbb{Z}_q}(U)$				
$\mathtt{cpk}_t = [\mathtt{csk}_t]G$				
	$\xrightarrow{V, \mathtt{cpk}_t, \mathtt{id}}$			
		store $(V, \mathtt{cpk}_t, \mathtt{id})$		

Fig. 4. BRAKE enrolment protocol instantiated with discrete-logarithm OPRF and Diffie-Hellman key exchange.

Verification (DL instantiation)

Client	Server	Evaluator
t' probe feature vector	$\mathtt{ssk} \in \mathbb{Z}_q$	$k \in \mathbb{Z}_q$
$\mathtt{spk} \in \mathbb{G}$	$\mathtt{spk} \in \mathbb{G}$	
	$(V, \mathtt{cpk}_t, \mathtt{id})$	

	$\xrightarrow{\mathtt{id}}$			
	$\xleftarrow{V}$			
find $\{(b, V(b)) : b \in t'\}$				
and decode to $f' \in \mathbb{F}_p[x]$				
$r' \leftarrow\!\!\$\; \mathbb{Z}_q$				
$B' = [r']H_{\mathbb{G}}(f')$				
$\mathtt{csk}_e \leftarrow\!\!\$\; \mathbb{Z}_q$		$\mathtt{ssk}_e \leftarrow\!\!\$\; \mathbb{Z}_q$		
$\mathtt{cpk}_e = [\mathtt{csk}_e]G$		$\mathtt{spk}_e = [\mathtt{ssk}_e]G$		
	$\xrightarrow{B', \mathtt{cpk}_e}$		$\xrightarrow{B'}$	
		$\mathtt{ck}_s \leftarrow \mathtt{KDF}([\mathtt{ssk}_e]\mathtt{cpk}_e,$ $[\mathtt{ssk}]\mathtt{cpk}_e, [\mathtt{ssk}_e]\mathtt{cpk}_t,$ $\mathtt{cpk}_e, \mathtt{spk}_e, \mathtt{cpk}_t, \mathtt{spk})$		$S' = [k]B'$
	$\xleftarrow{S', \mathtt{spk}_e, H(\mathtt{ck}_s)}$		$\xleftarrow{S'}$	
$U' = [r'^{-1}]S' = [k]H_{\mathbb{G}}(x')$				
$\mathtt{csk}_{t'} \leftarrow H_{\mathbb{Z}_q}(U')$				
$\mathtt{cpk}_{t'} = [\mathtt{csk}_{t'}]G$				
$\mathtt{ck}_c \leftarrow \mathtt{KDF}([\mathtt{csk}_e]\mathtt{spk}_e,$ $[\mathtt{csk}_e]\mathtt{spk}, [\mathtt{csk}_{t'}]\mathtt{spk}_e,$ $\mathtt{cpk}_e, \mathtt{spk}_e, \mathtt{cpk}_{t'}, \mathtt{spk})$				
return $H(\mathtt{ck}_c) = H(\mathtt{ck}_s)$				

Fig. 5. BRAKE verification protocol instantiated with discrete-logarithm OPRF and Diffie-Hellman key exchange.

template $t_{\mathtt{id}}$ for some registered identity id. Then the DL-BRAKE protocol in Figure 5 is correct.

Proof sketch. This follows directly from the construction. If the comparison result of the probe feature set t' to a biometric template t_id is within the the verification threshold τ for some registered identity 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. If the distance between probe and reference feature set is more than τ points, by correctness of Lagrange interpolation, two different polynomials will be reconstructed, and, but for a collision in the hash function, the key exchange will fail. □

Theorem 2 (Client Privacy)**.** Let $\mathcal{A}_0$ be an adversary against *client privacy* in the DL-BRAKE 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 evaluated 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 in the DL-BRAKE 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-Evaluator Privacy)**.** Let $\mathcal{A}_0$ be an adversary against client-evaluator privacy in the DL-BRAKE protocol in Figure 5 with advantage ϵ_0 controlling both the client and the evaluator. 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 evaluator. 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 evaluator'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-Evaluator Privacy)**.** Let $\mathcal{A}_0$ be an adversary against server-evaluator privacy in the DL-BRAKE protocol in Figure 5 with advantage ϵ_0 controlling both the server and the evaluator. 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 evaluator. Then $\mathcal{A}_0$ can guess a biometric probe, interpolate to get a polynomial f' and execute the OPRF on input f' using the evaluator'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. □

H. Improved Security using NIZK

The protocol can be further secured by the addition of non-interactive zero-knowledge proofs (NIZKs). We show how to apply a standard Chaum-Pedersen zero-knowledge proof [55] using a Fiat-Shamir transform [56] yielding a non-interactive proof.

The NIZK is added to prove the honest evaluation of the OPRF. Thereby, a client can verify that the evaluator computed the evaluation 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 evaluator. 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.

I. 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 [57]. 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 [58], 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 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 otherwise, see [59] for more details. This specific PRF construction is put in place to enable verifiability and security against active adversaries.

However, the zero-knowledge proof appended to the lattice-based PRF for active security are not practical for real-world situations due to proof sizes of several gigabytes [13]. Therefore, we only look at the case of passive security against dishonest clients 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 [13], continuing to omit one of the zero-knowledge proofs. The hashed input is blinded with an R-LWE sample and sent to the evaluator to obtain an evaluation by a secret OPRF key k. The evaluator computes the evaluation as another R-LWE sample, and the evaluated 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 session pre-key using the client's stored reference public key. The client can only decapsulate the pre-key if its 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.

IV. 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://github.com/dasec/BRAKE and includes automated installation scripts with all dependencies in order to support the reproducibility of our work.

For the fingerprint fuzzy vault instantiation, we used the open-source implementation provided by [15] with all original parameter settings, in particular, the minutiae quantisation and encoding into a product of finite field $\mathbb{F}_{2^{18}} \times \mathbb{F}_{2^{18}}$ which accommodates a unique encoding of at most $t_{max} = 44$ genuine minutiae as described in [15]. Keeping the parameter choices evaluated in the work of [15] ensures perfect replaceability with other state-of-the-art fuzzy vault instantiations, such as [26] for iris and [27] for face. In particular, we run our implementation on the same fingerprint database MCYT-330 [33] 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 [26], [27].

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 III-C). 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 [60] as the elliptic curve and SHA-256 as the hash function.

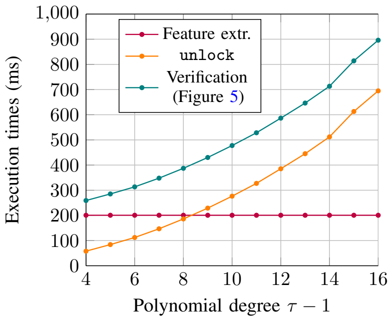


Fig. 6. Execution times in milliseconds for the DL-BRAKE protocol instantiated with fingerprint fuzzy vault [15].

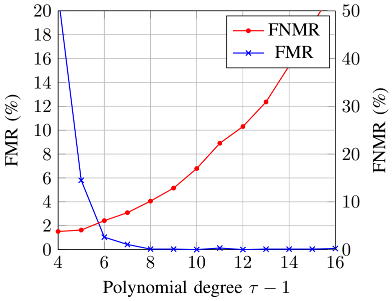


Fig. 7. Biometric performance for the DL-BRAKE protocol instantiated with fingerprint fuzzy vault [15].

To begin, we give a more detailed comparison of our work with closely related work in Table 3 by extending 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. In order to prevent offline attacks, a minimum number of two rounds of communication is necessary. Therefore, [9], [10], and our protocol can be considered optimal in terms of number of rounds. As [7] constructed a one-round protocol, this leaves them open to offline attacks. 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 6 as well as in terms of biometric performance shown in Figure 7. In comparison, fPAKE [9] does not achieve irreversibility as templates are disclosed to the server in plaintext, fuzzy aPAKE [10] does not achieve computational efficiency, and [7] does not achieve an acceptable biometric performance, as we show in Appendix B.

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-BRAKE given in Figure 5 takes 387.15

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

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 6 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 when setting up the system, and does not affect verification performance.

Accordingly, Figure 7 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 32 bytes for any point on the elliptic curve P-256 [60] (i.e., $\mathtt{cpk}_e, \mathtt{spk}_e, B'$ and S'), 99 bytes for a locked fuzzy vault of degree at most 43 and coefficients in $\mathbb{F}_{2^{18}}$, and 32 bytes for the hash digest.

V. Conclusions

In this work, we constructed secure biometrics-authenticated key exchange from fuzzy vaults and proved its security in compliance with ISO/IEC 24745. Our BRAKE 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 attacks. 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 evaluated 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 works may focus on 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 four-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 *Proceedings of the International Conference 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 Transactions 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 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.

TABLE 3
Summary of our protocol compared to previous published protocols as described in Table 1 of [7].

Scheme	Technique	Rounds	Communication cost	Compatibility	ISO/IEC 24745 [6]
fPAKE-1 [9]	Garbled Circuits	5	N/A	iris, fixed-length fingerprint	✗
fPAKE-2 [9]	PAKE + Secret Sharing	2	N/A		✗
fuzzy aPAKE-1 [10]	Secret Sharing + OT	2	$\sim$ 700 KB		✗
fuzzy aPAKE-2 [10]	aPAKE	2	$\sim$ 1 MB		✗
BAKE-1 [7]	Random Linear Codes	1	5-8.4 KB	minutiae-based fingerprint	✗
BAKE-2 [7]	Secret Sharing + Polynomial Interpolation	1	1.7-96.6 KB	iris	✗
DL-BRAKE (ours)	Fuzzy Vault + OPRF + KEM	2	0.3 KB	minutiae-based fingerprint, iris, face	✓

TABLE 4
Execution times in milliseconds for the DL-BRAKE 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	—

[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 *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 International Conference on Pattern Recognition (ICPR)*, 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 Transactions on Pattern Analysis and Machine Intelligence*, March 2010.

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

[19] S. Foundation, "Technical information - specifications and software libraries for developers," 2022, https://signal.org/docs/.

[20] H. Proença, "Unconstrained iris recognition in visible wavelengths," in *Handbook of Iris Recognition*. Springer, 2016, pp. 321–358.

[21] A. Everspaugh, R. Chaterjee, S. Scott, A. Juels, and T. Ristenpart, "The pythia PRF service," in *USENIX Security Symposium*, 2015, pp. 547–562.

[22] 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.

[23] 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.

[24] 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.

[25] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for ŁIndustrial and ŁApplied ŁMathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[26] 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.

[27] 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.

[28] 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.

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

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

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

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

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

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

[35] 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.

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

[37] M. Olsen, V. Šmida, and C. Busch, "Finger image quality assessment features - definitions and evaluation," *IET Biometrics*, vol. 5, no. 2, pp. 47–64, June 2016.

[38] E. Tabassi, M. Olsen, O. Bausinger, C. Busch, A. Figlarz, G. Fiumara, O. Henniger, J. Merkle, T. Ruhland, C. Schiel, and M. Schwaiger, "NIST interagency report 8382," National Institute of Standards and Technology, NIST Interagency Report 8382, July 2021.

[39] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcard-based fingerprint authentication," in *Proceedings of the ACM SIGMM Workshop on Biometrics Methods and Applications*, 2003, pp. 45–52.

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

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

[42] 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.

[43] 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.

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

[45] 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.

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

[47] 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.

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

[49] 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.

[50] 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.

[51] S. Marcel, M. S. Nixon, J. Fierrez, and N. Evans, *Handbook of biometric anti-spoofing: Presentation attack detection*. Springer, 2019, vol. 2.

[52] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," in *Proceedings Annual Symposium on Foundations of Computer Science*. IEEE, 1998, pp. 28–37.

[53] T. Silde and M. Strand, "Anonymous tokens with public metadata and applications to private contact tracing," in *Financial Cryptography and Data Security*, I. Eyal and J. Garay, Eds. Cham: Springer International Publishing, 2022, pp. 179–199.

[54] N. Tyagi, S. Celi, T. Ristenpart, N. Sullivan, S. Tessaro, and C. A. Wood, "A fast and simple partially oblivious PRF, with applications," in *EUROCRYPT 2022, Part II*, ser. LNCS, O. Dunkelman and S. Dziembowski, Eds., vol. 13276. Springer, Heidelberg, May / Jun. 2022, pp. 674–705.

[55] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *Annual International Cryptology Conference*. Springer, 1992, pp. 89–105.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

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

[61] 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 the additional NIZK as discussed in Section III-G. 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. 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 [61], 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 [34]. 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 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% [36] 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% using the SourceAFIS comparison algorithm[2]. 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 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.

[2] https://sourceafis.machinezoo.com/

[3] http://biometrics.idealtest.org

Verification (DL instantiation with NIZK)

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

Fig. 8. BRAKE verification protocol with discrete-logarithm NIZK.