Biometric-Based Non-transferable Anonymous Credentials*

Marina Blanton[1] and William M.P. Hudelson[2]

[1] Department of Computer Science and Engineering, University of Notre Dame
mblanton@cse.nd.edu
[2] Mathematics Department, Pennsylvania State University
phil.hudelson@gmail.com

Abstract. This work explores the problem of using biometric data to achieve non-transferability of anonymous credentials; that is, sharing of anonymous credentials, which allow one to anonymously authenticate, can be severely limited if their use requires possession of the credential owner's biometric. We target to provide strong security guarantees using minimal trust assumptions, namely that a fresh reading of a biometric is enforced on each use of the credentials. Furthermore, no biometric or other information is compromised if an adversary obtains full access to all credential-related data. Our solution relies on constructions for fuzzy extractors that allow one to extract and reproduce a random string from noisy biometric images. We first examine security requirements of biometric key generators, and then show how they can be integrated with anonymous credentials to achieve a high degree of non-transferability and security.

1 Introduction

Biometric-based authentication is becoming more prevalent today. This is well justified by the advances in biometric recognition techniques and a higher degree of security of biometric-based authentication compared to some alternative authentication mechanisms. Biometric-based authentication is also viewed as convenient to users, as it does not require them to remember passwords or carry authentication tokens or keys. Biometric data, however, requires very careful handling and protection, since, once captured, it can uniquely identify an individual and cannot be revoked. For that reason, a lot of research in the recent years has been dedicated to designing mechanisms that minimize the impact of (accidental or intentional) leakage of biometric data stored in databases for authentication purposes. Instead of working on protecting privacy of biometric data, the direction explored in this work comes from the (opposite) idea of using biometrics to aid privacy and anonymity. In particular, biometric data can be incorporated into privacy-preserving tools to limit abuse of anonymity, and thus enable a safer deployment of such techniques on a wider scale.

Anonymous authentication allows the prover to convince the verifier that she has a certain property or credential without revealing any other information (and without

* Portions of this work were sponsored by grant AFOSR-FA9550-09-1-0223. This work was performed while the second author was at the University of Notre Dame.

S. Qing, C.J. Mitchell, and G. Wang (Eds.): ICICS 2009, LNCS 5927, pp. 165–180, 2009.

the ability of two authentication instances to be linked to the same individual by the verifier). In the case of theft or voluntary sharing of such credentials by duplication, these credentials can be freely used by others without the ability to find out the identity of their owner. Incorporating biometrics in the authentication process then results in a high degree of assurance that credentials cannot be transferred and that access is performed by credential bearers only. We arrive at a setting where biometrics is used to ensure non-transferability of anonymous credentials realizing the concept of accountable anonymity. The benefits of such an approach are clearly seen in numerous applications where non-transferability is essential. For example, in an organization where digital locks are used to enter buildings and facilities, it is extremely easy to track one's movements, and the employer that respects individual privacy can be willing to implement anonymous access to its facilities, i.e., the same as conventional keys provide.[1] Digital keys can then be issued in the form of anonymous credentials that encode access privileges of an employee, and should not be used by any other individual. Other important applications of non-transferable anonymous credentials include certification that the credential-bearer is a U.S. citizen, guaranteeing that the insurance company will pay for an anonymous HIV test of the credential-bearer, and others.

When biometry is integrated into the authentication process, to achieve unlinkability of protocol executions, biometrics can no longer be scanned by the device that performs access control enforcement. This means that biometrics are captured on a device that each user carries and which is trusted by the system. Then during the enrollment phase, user credentials are placed on that device, and during authentication the device is trusted to capture a biometric and use the scan in combination with the user credentials to authenticate anonymously. While anonymous credentials schemes where biometrics is used to achieve non-transferability have been proposed in the past [5,24], the key difference between our and prior work is in the trust requirements we place on the tamper-resistant device. In our case these requirements are minimal: the only functionality the device is required to do correctly is to enforce capturing a new biometric (and erase it afterwards). Should the integrity of the device be compromised and all information stored on it retrieved, it is not feasible to either recover the biometric data that identifies the credential-bearer or to successfully authenticate using the captured credentials (this is achieved without reliance on any additional secrets that the credential-bearer must know).

The structure and contributions of this work are as follows: We first discuss biometrics key generators, which allow one to extract cryptographic keys from biometric data, including their common constructions and security requirements. We then propose a generic way of improving security of biometric key generators with respect to privacy protection of the biometric from which such keys are derived. In addition to being useful for biometric key generators, this construction has a direct application to our anonymous credentials scheme where biometrics are used to achieve non-transferability. The next part of this work is dedicated to constructing a general solution for such anonymous credentials, to which we refer as biometric-based non-transferable credentials.

[1] Note that anonymous access will not be suitable for certain organizations where access to information or restricted-access facilities must be monitored by law or other provisions, but in most environments anonymous access to facilities is natural and harmless if access control is properly implemented.

The only requirement that we pose to ensure correct operation of the system is that biometric readings are enforced on each use of the credentials (and erased afterwards). Once again, this offers protection of credentials in cases of both voluntary duplication of anonymous credentials and when such credentials are stolen.

2 The Model and Preliminaries

2.1 The Model

The operation of our system proceeds in a standard way: First, an authority A sets up the system, which includes generation of a public-private key pair (pk_A, sk_A), where the private key will allow it to issue user credentials and the public key can be used to verify them. When a user joins the system, her biometric is recorded and credentials are issued by the authority in accordance with user privileges. During the authentication protocol, the user's biometric is captured by her device and the credentials are verified by the authority (or a different entity on behalf of the authority) in an anonymous way. More precisely, an authentication scheme consists of the following components:

- AC-Setup is an algorithm that, on input a security parameter κ, sets up the system including A's public-private key pair (pk_A, sk_A). Its output consists of public parameters pub that include pk_A, and A's secret key sk_A.
- AC-Enroll is a procedure during which, given system's parameters pub, a user $\mathcal{U}$'s input consists of her biometric, the authority's input consists of its secret key sk_A and $\mathcal{U}$'s privileges. It results in $\mathcal{U}$ obtaining credentials cred that are tied to her biometric and specify her access privileges.
- AC-Auth is a protocol between a user $\mathcal{U}$ and server $\mathcal{S}$, in which $\mathcal{U}$'s input consists of her fresh biometric reading and credentials cred, and the server's input consists of public parameters pub. Authentication is successful if the server could successfully verify $\mathcal{U}$'s credentials as authentic and meeting the access control policy.

A secure anonymous biometric-based authentication scheme must satisfy the following:

Completeness: Every honest user should be able to successfully authenticate using her biometric and credentials.

Soundness: A person without proper credentials should not be able to successfully authenticate with more than negligible (in κ) probability after observing any (polynomial) number of successful authentication protocols. Furthermore, any coalition of valid users should not be able to gain access to more resources than what they can already legitimately access with more than negligible probability.

Unlinkability: To satisfy anonymity requirements, we require that one should not be able to determine with more than negligible probability whether two executions of the authentication protocol correspond to the same user or different users.

Privacy of biometric: We require that in the case of compromise of a user's device, the information stored on it does not allow one to learn the credentials owner's biometric or successfully authenticate without the knowledge of it.

These properties are defined formally in section 4.2.

2.2 Signatures with Protocols

We use signature schemes due to Camenisch and Lysyanskaya [12,13], which have two protocols associated with them: (i) they allow a user to obtain a signature on a committed value without revealing that value to the signer; and (ii) they enable a user to convince a third party that she possesses a signature on a certain value. The commitment scheme used is the Pedersen commitment [32], in which the public parameters are a group G_q of prime order q such that the discrete logarithm problem is hard in G_q and generators $g_0, g_1, \ldots, g_\ell$. To compute a commitment to $x_1, \ldots, x_\ell \in \mathbb{Z}_q$, we randomly choose $r \in \mathbb{Z}_q$ and set $com(x_1, \ldots, x_\ell; r) = g_0^r \prod_{i=1}^{\ell} g_i^{x_i}$. This commitment is unconditionally hiding (i.e., $com(x_1, \ldots, x_\ell; r)$ reveals no information about $x_1, \ldots, x_\ell$) and is computationally binding (assuming that the discrete logarithm problem is hard in G_q, the sender cannot open the commitment to values other than $x_1, \ldots, x_\ell$).

Then given a commitment $com(x_1, \ldots, x_\ell; r)$, it is possible obtain the signer's Camenisch-Lysyanskaya (CL) signature $\sigma(x_1, \ldots, x_\ell)$ without revealing any information about the values $x_1, \ldots, x_\ell$ to the signer. Furthermore, possession of $\sigma(x_1, \ldots, x_\ell)$ allows its owner to use commitments to $x_1, \ldots, x_\ell$ to prove to other parties that she has the signer's signature on the values included in the commitments without revealing additional information about the signed values themselves. If this protocol is combined with a zero-knowledge proof that the values included in these commitments satisfy certain properties, it becomes possible to convince a third party that the prover possesses a CL signature that meets these conditions without disclosing additional information about the signed values. The signature scheme [12] relies on the Strong RSA assumption for its security. The scheme [13] relies on LRSW assumption in groups with bilinear maps.

2.3 Zero-Knowledge Proofs of Knowledge

Zero-knowledge proofs of knowledge (ZKPKs) allow one one party, the prover, to prove to another, the verifier, the veracity of some statement without revealing to the verifier any information besides the fact that it is valid. Prior literature provides efficient ZKPKs for a variety of statements, with many efficient proofs being based on the discrete logarithm problem (see, e.g., [16,15,14,6,9]). ZKPKs used in our protocols are a proof of knowledge of the discrete logarithm representation, equality of discrete logarithms, and conjunction of two or more statements [15], solutions to which are well known. Verification of context-specific user privileges might include other techniques such as, e.g., a proof of knowledge that a committed values lies in an interval and others.

3 Biometric Key Generators

Before proceeding with our anonymous credentials scheme, we give a brief description of a *biometric key generator* (BKG), which is a system that allows one to produce a cryptographic key from a biometric and later reconstruct the key using the same (noisy) biometric. Constructions for biometric key generators will be directly used in our biometric-based anonymous authentication scheme, and the results presented in this section have uses in both biometric key generation and our construction of non-transferable anonymous credentials. In what follows, we assume that the system is first

initialized using a security parameter κ, which is used to determine other parameters and algorithms' configurations (i.e., BKG-Setup algorithm is implicit).

BKG-Enroll: a probabilistic algorithm that, on input a user $\mathcal{U}$'s identity and a biometric representation W, outputs a cryptographic key K and *public information* or *helper data* P that will aid key recovery. If the input does not meet certain criteria, the algorithm might output a special failure symbol $\perp$.

BKG-KeyRec: a deterministic algorithm that, on input a biometric representation W' and helper data P, outputs either a cryptographic key K or a failure symbol $\perp$ if it is unable to compute a key from its input.

Normally, during the enrollment phase, a cryptographic key is chosen anew and locked with the biometric, or is derived from the biometric. The purpose of the helper data is, given a noisy biometric image W', to correct the errors and permit unlocking or derivation of the key. Recovery of the correct key is possible only if the biometric representations W and W' are close enough, for some definition of distance specific to the type of biometrics and BKG construction used. Helper data P is designed to leak as little information about the biometric as possible, so that it can be assumed to be non-private data.

Recent work of Ballard et al. [3] lists security properties that must hold for a BKG. We define them next with the difference that we additionally require privacy of the biometric to hold when a user $\mathcal{U}$ is enrolled in the system more than once using the same (noisy) biometric and the same or a similar key generation mechanism. In other words, we want biometric information remain protected when $\mathcal{U}$'s key is lost or compromised and it re-enrolls, when it legitimately assumes more than one role within the system, or when it is enrolled at more than one system that use similar biometric key generators.

- *Key randomness* (REQ-KR): A key K contains sufficient amount of randomness (based on the security parameter κ) and appears random to any adversary with access to the helper data P used in deriving K and any auxiliary information.
- *Weak biometric privacy* (REQ-WBP): Given helper data P and any auxiliary information, any adversary does not learn useful information about the biometric W used in generating P. Often the difficulty of recovering the entire biometric W can be sufficiently well measured, but it is desirable that learning parts of the biometric is also difficult (i.e., we might desire to prevent an adversary from learning any function of the biometric).
- *Strong biometric privacy* (REQ-SBP): Given helper data P, any auxiliary information, and the key K itself, any adversary does not learn any useful information about the biometric W used in generating P. Similar to the above, we might require that an adversary is unable to compute any function of the biometric.

In the above, the auxiliary information refers to any additional information that can surround the system. Such information available to an adversary can contain biometric data, corresponding helper data, and keys, not associated with the user in question. It can also contain information about distributions of biometric data, implementation decisions, and other information from the environment that can potentially weaken the security of the keys or biometrics.

In recent years, a large number of proposals for biometric key generators have been developed. Examples of thoroughly analyzed and well studied constructions include fuzzy vaults [25] (and work that builds on them [17,36,34,37,29,28,27,30]), secure sketches and fuzzy extractors [21,7,26,8,20] (and work that builds on them [23,2,10,1]). Many of them were designed and adapted to work on real data before the security requirements were defined in the current form. For that reason, many existing constructions and implementations would fail to achieve the strong biometric privacy requirement (and some proposals can fail other requirements as shown in [3]). Here we show that property REQ-SBP is not hard to achieve if a BKG construction can meet the REQ-WBP requirement.

Before we present our generic conversion from REQ-WBP to REQ-SBP, we first need to provide more detail about how BKGs normally work. Secure sketches [21,20] were introduced as a mechanism of correcting errors in noisy secrets (e.g., biometrics) by releasing a helper string S that does not reveal a lot of information about the secret. A secure sketch is generated using a "sketch" procedure SS that, given W, produces a string S; and can be "recovered" using Rec that given W' and S, outputs W if the distance between W and W', $\mathsf{dist}(W, W')$, in the appropriate metric space is at most t. Secure sketches have been constructed for different types of metric spaces including the Hamming distance (applicable to iris codes, which are represented as binary strings) and set intersection (applicable to fingerprints represented as a set of points in a two-dimensional plane). Security of a secure sketch is evaluated in terms of leakage of biometric information associated with the release of helper data, i.e., the difference between the "worst-case" entropy of W and the average minimum entropy of W after the release of S.[2] Since secure sketches are normally built using error-correcting codes, the larger the number of errors that need to be corrected, the more redundancy needs to be included in the code, and the less effective it is at protecting information about the biometric stored in the helper data.

Fuzzy extractors allow one to extract randomness from W (for use in cryptographic constructions) and later reproduce it exactly using different W' close to the original W. A fuzzy extractor is generated by Gen that, on input W, outputs extracted random string R and a helper string P; and can be reproduced by Rep that on input W' and P outputs R that was generated using $\mathsf{Gen}(W)$ if $\mathsf{dist}(W, W') \leq t$. The security requirement is such that, for any W with sufficient entropy, R is sufficiently close to a uniformly chosen random string, even after observing the helper data P. A fuzzy extractor can be built from a secure sketch as follows [21]: on input W, Gen executes $S \leftarrow \mathsf{SS}(W)$ and applies a strong extractor Ext to W to extract a random string R. S and random coins used by Ext form the helper data P. Let r_1 denote the random coins used by SS and r_2 random coins used by Ext (i.e., execution is of the form $\mathsf{SS}(W; r_1)$ and $\mathsf{Ext}(W; r_2)$). We obtain $P = (S, r_2)$. Algorithm $\mathsf{Rep}(W', P)$ uses S from P to recover the original W (given that $\mathsf{dist}(W, W') \leq t$) and extracts R by computing $\mathsf{Ext}(W, r_2)$. Fuzzy vault is a secure sketch construction for the set intersection metric, so we will use secure sketch and fuzzy extractor terminology in the rest of this paper.

One complaint regarding the theoretical work on biometric key generation is that proposed solutions cannot tolerate realistic variations in the biometric signal such as

[2] We refer the reader to [21] for precise definitions.

variable-length representations, unordered or unaligned representations [35]. In fact, the unlocking algorithm in the fuzzy vault implementation given by Uludag et al. [34] produces many potential keys, one of which must be selected as authentic. Uludag et al. [34] address this issue by adding some structure (i.e., redundancy) to the secret being locked, thus weakening the hiding properties of the construction. Such variations in the biometric, however, are not arbitrary and often key recovery still can be performed using the proposed schemes. Thus, to aid recovery of the exact secret among several candidates, we propose to ship helper data with a verification value that will permit confirmation of the correct key. Adding a verification value computed using a one-way function can cleanly avoid additional information leakage. Now we are ready to describe our construction.

Assume that we are given a BKG $(\mathsf{BKG\text{-}Enroll}_0, \mathsf{BKG\text{-}KeyRec}_0)$ that achieves REQ-WBP, but $\mathsf{BKG\text{-}KeyRec}_0$ might output several key candidates rather than a single key. Let f be a one-way function that hides all information about its inputs and let $||$ denote string concatenation. We then simultaneously address key confirmation and strong biometric privacy using the following construction:

$\mathsf{BKG\text{-}Enroll}(W)$

1. Run $(K_0, P_0) \leftarrow \mathsf{BKG\text{-}Enroll}_0(W)$.
2. Set $P = (P_0, f(K_0||\text{“0”}))$ and $K = f(K_0||\text{“1”})$.
3. Output (K, P).

$\mathsf{BKG\text{-}KeyRec}(W', P = \{P_1, P_2\})$

1. Run $\mathsf{BKG\text{-}KeyRec}_0(W', P_1)$ to find a set $\mathcal{K}$ of candidate keys for K_0.
2. For each $k \in \mathcal{K}$, if $f(k||\text{“0”}) = P_2$, set $k = K_0$ and output $K = f(K_0||\text{“1”})$.

Theorem 1. *Given a BKG* $(\mathsf{BKG\text{-}Enroll}_0, \mathsf{BKG\text{-}KeyRec}_0)$ *that satisfies* REQ-WBP *and a one-way function* f *that hides information about its inputs, the above construction for* $(\mathsf{BKG\text{-}Enroll}, \mathsf{BKG\text{-}KeyRec})$ *satisfies* REQ-SBP.

Proof. The proof is straightforward. Given $P = \{P_1, P_2\}$, K and auxiliary information aux, we need to show that an adversary $\mathcal{A}$ does not learn useful information about W. By the assumption that $(\mathsf{BKG\text{-}Enroll}_0, \mathsf{BKG\text{-}KeyRec}_0)$ satisfies REQ-WBP, $\mathcal{A}$ does not learn information about W from P_1 and aux. The only other information available to $\mathcal{A}$ is P_2 and K, which, by our assumption, were produced using a one-way function that hides information about its inputs. This means that $\mathcal{A}$ cannot gain information about K_0 or W from these data. □

4 Non-transferable Anonymous Credentials via Biometrics

Now we proceed with combining biometric key generators with anonymous credentials to result in anonymous biometric-based authentication with non-transferable credentials. Recall that we want a user's credentials to encode a cryptographically strong value derived from the user's biometric. This value is not stored with the credential (and cannot be recovered from the credential) and must be recomputed from the biometric on each use. To ensure that the value was actually derived from the biometric, we want

the derivation procedure to be one-way, and a proof of derivation should be provided at authentication time in zero knowledge.

In our construction of non-transferable anonymous credentials we crucially rely on the following fact that holds true for known secure sketch constructions and allows us to build a simple and efficient solution.

Fact 1. *Let* $P \leftarrow \mathsf{SS}(W; r_1)$*, where* r_1 *denotes the random choices the algorithm* SS *makes. Then knowledge of* W'*, such that* $\mathsf{dist}(W, W') \leq t$*, implies knowledge of* r_1 *and* W*. Furthermore, knowledge of* r_1 *implies knowledge of* W *and therefore some* W'' *with* $\mathsf{dist}(W, W'') \leq t$*.*

Since the trust requirement we place on the device is to enforce a fresh biometric reading, it would be necessary to provide assurance that the key was actually derived from W'. Using the above fact, however, we have that knowledge of one of W, r_1, and W' is equivalent in presence of P. Thus, if an adversary is able to recover r_1 or W from P and then compromise the integrity of the device to avoid enforcing a new biometric scan, it will be able to successfully produce W' and pass the verification procedures of the protocol. The above fact then mitigates the need to provide expensive zero-knowledge proofs of correct computation of W from W' (procedure Rec) and lets us enforce only a proof that the key was derived from the original biometric W (and, as mentioned above, we want the derivation process to be one-way). Fuzzy extractors were not explicitly designed to be one-way and in fact are not guaranteed to be one-way (i.e., the underlying randomness extractors do not have this property). Thus, we enforce the one-way requirement of key derivation using the construction for REQ-SBP given in the previous section. Note that in this case the REQ-SBP property is not as crucial as in the case of BKGs, because with anonymous biometric-based authentication the derived key does not leave the client; instead, one-wayness of the computation ensures the verifier that the necessary value was actually derived from biometric data.

In our construction, we assume that fuzzy extractor parameters are chosen in such a way as to output strings indistinguishable (to a polynomial-time distinguisher) from strings chosen uniformly at random (for the appropriate choice of the security parameter). That is, using the terminology of randomness extractors, we assume that the fuzzy extractor is configured to output strings ϵ-close to uniform, where ϵ is a negligible function of the security parameter. See [31] for more detail. This makes the output suitable for use in cryptographic protocols.

4.1 The Scheme

To be able to prove to the server that the computation was performed as prescribed, the application of one-way function f must be verifiable in zero-knowledge. To achieve this, we use the verifiable random function (VRF) of Dodis and Yampolskiy [22] to implement f. This function is computed as $y = f_{sk}(x) = g^{1/(sk+x)}$ over groups of prime order with bilinear maps, where g is the group generator, sk is the secret key, and x is the input to the function (which can be public); its security is based on q-DHI and q-DBDHI assumptions (see [22] for more information). In particular, we will assume that the verification value and the extracted key are computed as $f_{sk}(\text{“0”})$ and $f_{sk}(\text{“1”})$, where sk is derived from W' and the fuzzy extractor helper data (and must meet the requirements for f).

AC-Setup: On input security parameter 1^κ, authority A sets up the group parameters and its public-private key pair (pk_A, sk_A) for the CL signature scheme. All values except the signing key sk_A are stored in pub.

AC-Enroll:

1. Biometric W of user $\mathcal{U}$ is measured with small error, BKG-Enroll(W) is executed to produce (P, K), where $P = (P', V)$ such that (P', K') is the output of the fuzzy extractor Gen algorithm and $V = f_{K'}(\text{“0”})$ denotes the verification data.
2. $\mathcal{U}$ chooses z_1 at random and sends commitment $Z_1 = g^{z_1}$ to A.
3. A verifies the validity of the computation in step 1 and computes $K = f_{K'}(\text{“1”})$.
4. A chooses random z_2 and uses Z_1 to compute commitment $com(K, priv; z)$, where $z = z_1 + z_2$ (i.e., both A and $\mathcal{U}$ contribute randomness, but the value of z will be know only to the user).
5. A uses sk_A to produces signature $\sigma_A(K, priv)$ and sends it and z_2 to $\mathcal{U}$.
6. $\mathcal{U}$ computes $z = z_1 + z_2$ and verifies the validity of the signature.
7. User $\mathcal{U}$'s credentials cred $= (P, priv, \sigma_A(K, priv))$ are stored at his device.

Technically speaking, the signature here is a signature on values K, $priv$, and random value z, and the value of z will be necessary for showing the validity of the signature. Thus, it is implicitly assumed that this value is also stored with the user's credentials. With this setup, authority A learns biometrics and keys of users and is expected to erase such information after the enrollment protocol. This does not permit A to distinguish between different users at authentication time, but it might be desirable to prevent A from learning user biometrics at all; we leave this as a direction for future work.

AC-Auth: A user $\mathcal{U}$ with a device holding credentials cred $= (P, priv, \sigma_A(K, priv))$ engages in interaction with a server $\mathcal{S}$ as follows:

1. The device scans the user's biometric W', recovers K' using P' and confirms it using V stored in cred. It then forms commitment $C_1 = com(K'; z_1)$.
2. The device computes $K = f_{K'}(\text{“1”})$ and a commitment to it $C_2 = com(K; z_2)$.
3. The device computes commitment $C_3 = com(priv; z_3)$ using $priv$ from $\mathcal{U}$'s credentials.
4. The device sends to $\mathcal{S}$ C_1, C_2, C_3, and performs the following ZKPKs:
   (a) the opening of C_2 corresponds to the result of applying function f to the opening of commitment C_1 and string “1”;
   (b) $\mathcal{U}$ possesses A's signature on the opening of C_2 and C_3 (more precisely, only the parts of $priv$ relevant for obtaining access to the resource);
   (c) the opening of C_3 satisfies the access control rules imposed by $\mathcal{S}$.
5. $\mathcal{S}$ verifies all proofs in step 4, and if they pass, grants user $\mathcal{U}$ access to the resource.
6. The device erases all information captured and computed during the authentication process (in particular, it is important that the device erases W' and all information derived from it).

4.2 Security Analysis

In this section, we give a more detailed description of the security games and prove security of the scheme.

We model soundness as a game in which adversary $\mathcal{A}$ is allowed to corrupt users, obtain access to their credentials cred, and engage in authentication protocols with the server, as well as monitor authentication protocols of other (honest) users in the system. Let $\mathcal{U}_1, \ldots, \mathcal{U}_n$ denote the set of users that $\mathcal{A}$ controls and let $\mathsf{cred}_{\mathcal{U}_1}, \ldots, \mathsf{cred}_{\mathcal{U}_n}$ denote their corresponding credentials. $\mathcal{A}$ wins the game if it is able to successfully authenticate to the server by forging credentials $\mathsf{cred}'_{\mathcal{U}}$ of a user $\mathcal{U}'$ it does not control or by gaining access to more resources than what the corrupted users can already access (and possibly authenticating as one of the users it controls). An authentication scheme is sound if $\mathcal{A}$ has at most negligible advantage in winning this game.

One way to model unlinkability is to have a simulator Sim that can simulate a valid execution of the authentication protocol without access to user information. Then no information about the user is leaked if the server's view after interacting with a valid user is indistinguishable from its view after interacting with the simulator. The unlinkability game proceeds as follows: the adversary $\mathcal{A}$ represents the authority A colluding with the server $\mathcal{S}$. It creates the authority's private and public keys, enrolls all users, and possibly corrupts some users obtaining full access to their credential information (including random choices made at enrollment time). After that, $\mathcal{A}$ engages in a challenge authentication protocol with either a valid uncorrupted user or a simulator. We say that unlinkability is achieved if, for any adversary $\mathcal{A}$, it is able to correctly guess whether it was communicating with a real user or a simulator with the probability at most negligibly larger than $1/2$.

To ensure biometric privacy, we let $\mathcal{A}$ obtain access to user credentials. Then $\mathcal{A}$ who is in possession of credentials $\mathsf{cred}_{\mathcal{U}_1}, \ldots, \mathsf{cred}_{\mathcal{U}_n}$ should be unable to extract biometric information of any of the users with high probability and successfully authenticate (on behalf of one of them or as a non-existing user) without proper biometric data. This property is required to hold even if $\mathcal{A}$ colludes with $\mathcal{S}$.

Before showing security of the overall scheme, we provide a supplemental result. Let $\sigma_A(m_1, \ldots, m_\ell; r)$ denote CL-signature with A's key on values $m_1, \ldots, m_\ell$ using randomness r. That is, we make the random value used for hiding the messages explicit in the representation of the signature.

Lemma 1. *There exists a CL-signature scheme over group G such that, given a CL-signature $\sigma_A(m_1, \ldots, m_\ell; r)$ and values $m_1, \ldots, m_{j-1}, m_{j+1}, \ldots, m_\ell$ for some $1 \leq j \leq \ell$ and r, it is infeasible for a polynomial-time adversary to determine m_j, where m_j is drawn from a distribution indistinguishable from uniform to a polynomial-time adversary, assuming that the discrete logarithm problem is hard. This holds even if the signing key sk_A is known.*

We omit the proof due to space considerations.

Theorem 2. *Assuming the security of the CL-signature scheme and the verifiable random function f, the scheme presented above is a secure anonymous biometric-based authentication scheme in groups where the discrete logarithm problem is hard.*

Proof. Completeness: this property can be shown to hold by examination.

Soundness: As previously described, we let adversary $\mathcal{A}$ enroll in the system on behalf of users and corrupt existing users. For the set of users $\mathcal{U}_1, \ldots, \mathcal{U}_n$ that $\mathcal{A}$ controls, we

assume that not only $\mathcal{A}$ has access to their credential information $cred_{\mathcal{U}_i}$, but can also obtain access to information not stored in the credentials. In particular, we assume that $\mathcal{A}$ can have access to key K for each user it controls. Let $K^{(i)}$ denote the key belonging to user $\mathcal{U}_i$. To violate soundness of the scheme, $\mathcal{A}$ then will either attempt to authenticate using invalid K' or using one of valid $K^{(i)}$ with privileges $\mathcal{U}_i$ does not have.

In the first case, $\mathcal{A}$ has to produce a proof that either it knows the authority's signature on openings of C_2 and C_3 or it has to correctly form the zero-knowledge proof in step 4(a) of the AC-Auth using K' of $\mathcal{A}$'s choice. The former can happen only with a negligible probability assuming that CL-signatures are secure, and the latter can also happen only with a negligible probability assuming that the function f is secure.

In the second case, $\mathcal{A}$ attempts to authenticate using $K^{(i)}$ and privileges $\mathcal{U}_i$ does not have. The only way for $\mathcal{A}$ to achieve this is to produce a proof that it knows the authority's signature on openings of C_2 and C_3, which can happen with at most negligible probability.

Unlinkability: Let $\mathcal{A}$ be the adversary defined for the unlinkability game that represents the colluding A and $\mathcal{S}$ (and thus has access to enrollment information). During the challenge, $\mathcal{A}$ is asked to engage in an authentication protocol with either a real user $\mathcal{U}$ or a simulator Sim without access to any user information. Our simulator engages in the authentication protocol by performing the following steps:

1. Sim chooses K' at random, computes $K = f_{K'}(\text{“1”})$, and produces commitments $C_1 = com(K'; z_1)$ and $C_2 = com(K; z_2)$.
2. Sim selects privileges $priv$ and computes commitment $C_3 = (priv, z_3)$.
3. Sim produces a zero-knowledge proof that the opening of C_2 corresponds to the result of applying f to the opening of C_1, which we denote by π_1.
4. Sim produces a simulated proof of knowledge, π_2, of a CL-signature from the authority on the openings of C_2 and C_3. This requires usage of the corresponding simulator of CL-signatures.
5. Sim produces a proof π_3 that the opening of C_3 satisfies the access control rules.

At the end of this interaction, $\mathcal{A}$ obtains $(C_1, C_2, C_3, \pi_1, \pi_2, \pi_3)$. We next argue that $\mathcal{A}$'s view during interaction with a simulator is indistinguishable from its view when interacting with a real user.

The commitments C_1, C_2, and C_3 information-theoretically hide the values encoded in them, and therefore the values Sim chooses are indistinguishable from those chosen by real users. The proofs π_1 and π_3 produced by the simulator are real proofs of knowledge and thus are indistinguishable from a user's proofs. Finally, the (simulated) proof π_2 differs from a real proof of knowledge of a CL-signature, but due to the security of CL-signatures, $\mathcal{A}$ can distinguish between a real and simulated proofs only with a negligible probability. Therefore, $\mathcal{A}$ can distinguish between real and simulated protocol executions with at most negligible probability, as required.

Privacy of biometric: Let $\mathcal{A}$ be in possession of credentials $\mathsf{cred}_{\mathcal{U}_1}, \ldots, \mathsf{cred}_{\mathcal{U}_n}$. Since biometric information encoded in each user credential is independent of information included in credentials of other users, $\mathcal{A}$ does not gain additional advantage in impersonating a user or recovering her biometrics by using information in other credentials, and we consider attacking a user by using only her own credentials. We have

$\mathsf{cred}_{\mathcal{U}} = (\sigma_A(K, priv; z), z, P', V, priv)$. For $\mathcal{A}$ to impersonate the user, it has to either recover K from $\mathsf{cred}_{\mathcal{U}}$ or authenticate without knowledge of K. Lemma 1 states that the former is infeasible when K is indistinguishable from a random value (which in our case is true due to pseudo-randomness of output of function f that we use to produce K), and the soundness property states that the latter is infeasible. To be able to recover $\mathcal{U}$'s biometric from the credentials, the only way for $\mathcal{A}$ to do this is to recover the biometric (or other information that leads to recovery of the biometric) from P'. Assuming that a secure biometric generator is used to produce P', $\mathcal{A}$ can be successful only with a small probability. □

5 Practical Considerations

The purpose of this section is to assess the feasibility of applying theoretical cryptographic techniques to empirical biometric data, as proposed in this work. We use iris codes as an example biometric. The two main questions we would like to address is (i) whether biometric key generation from iris codes is feasible, and (ii) whether key material extracted from iris codes can satisfy the requirements of the cryptographic tools.

In regards to the first question, the work of Hao et al. [23] was able to achieve a notable step toward biometric key generation by constructing a secure sketch for iris data. The construction used nested error-correcting codes and was able to achieve excellent key recovery rates. In particular, it used Hamming distance over binary strings as the metric with two types of error-correcting codes. A disadvantage of any secure sketch based approach is that correcting a large number of errors can cause the public data to potentially leak a lot of information about the biometric. In particular, as described in section 3, the current techniques give only worst-case information leakage analysis, which is measured as the loss of entropy after the release of the public data. Then to correct 10% of errors in a 2048-bit iris code, up to 411 bits of entropy can potentially be leaked. An iris code, however, is estimated to have about 250 degrees-of-freedom [18], and a noticeably higher error rate for authentic codes must be tolerated.

A natural way to lower the entropy loss in this case is to attempt to reduce the noise. A pioneering work on biometric-based authentication of Davida et al. [19] gave the idea of reducing the noise by acquiring multiple samples and performing majority decoding to create a single image with low noise. That is, during both the enrollment and key recovery stages, the algorithms take a number of biometric readings $W_1, W_2, \ldots, W_m$ rather than a single one, and create a single representation W using majority computation that more accurately represents the corresponding biometric. This technique is believed to be effective, and was recently empirically evaluated on iris codes in [4]. For a realistic bound of error tolerance of 30% [35,23], the theoretical approach significantly reduces the error (e.g., to 5% if 15 scans were used), while in practice the error was shown not to go lower than 16–18% [4], which is still a significant improvement. Other techniques for reducing noise also exist (e.g., scanning both eyes instead of a single one for iris recognition), which can be combined to result in even lower error rates. This means that the entropy loss will become tolerable when the error rate is reduced to a rather small value. We emphasize that the entropy-based analysis is only an upper

bound on the information leakage that we can compute rather than a close estimate, and the latter is likely to be significantly lower.

Now with respect to the second question raised above, we have already seen that iris codes can be assumed to have at least 250 bits of entropy. Currently known constructions of extractors extract all of the randomness from a source [31] assuming a sufficient number of additional truly random bits (which in our case are supplied as a part of the public information in sufficient quantity). This means that we can extract a 250-bit random string from an iris code. Our construction is built using groups over elliptic curves with bilinear pairings for which using a 160–190-bit modulus is sufficient. Thus, a random string extracted from a biometric has sufficient amount of randomness, and we will be able to construct the key exactly as prescribed by the protocol.

In addition to the above concerns, recent work of Simoens et al. [33] shows that existing constructions of secure sketches and the keys produced by fuzzy extractors that rely on them might not meet security requirements sought of encryption keys. In particular, the release of public data might allow one to link together ciphertexts belonging to the same individual (i.e., generated using keys produced from related biometrics). This threat, however, does not exist in our framework because the public data always stays with the client and the authentication protocol does not leak any information that can be linked to the individual.

6 Related Work

Related work on biometric-based key generation is very extensive, especially publications at biometric-related venues, and its survey is beyond the scope of this work. Anonymous credentials where biometric data are used for non-transferability are known in prior literature. They were first introduced by Bleumer in [5], and later expanded upon by Impagliazzo and More in [24]. The construction of Bleumer is based on the "Wallet with Observer" protocol originally constructed by Chaum and Pedersen for achieving the properties of non-transferability and privacy-protection. The second paper extends and formalizes these results, and adds the feature of revoking credentials. Such "Wallet with Observer" architecture, which is used in both of these works, however, requires non-trivial tamper-resistant hardware that runs trusted processes and executes parts of cryptographic protocols.

Other approaches to achieving non-transferability include encoding external sensitive information, e.g., credit card numbers, into the credentials (see, e.g., [11]). Then, when such credentials are used, the owner must prove knowledge of said information to the verifier. While such information may be shared by close relatives or friends who will be able to use the credentials on behalf of the owner, such sharing is often acceptable to service providers, and their goal is to prevent large-scale sharing of credentials (by, e.g., posting credentials on a Web page). The above approach assumes that the credential issuer will have access to a large quantity of external or otherwise sensitive information about the user that can be included in the credential. When, however, such information is not readily available, alternative solutions must be sought. The current framework provides such an alternative.

7 Conclusions

This work examines most researched techniques for extracting keys from biometric data and their use in cryptographic applications. We first provide a generic mechanism of enhancing biometric privacy protection of biometric key generators. We then build on biometric key generators to construct anonymous credentials, where biometric is used to ensure non-transferability of user credentials. Unlike previous proposals, we target at making minimum trust assumptions on the execution environment: we only require a fresh biometric to be captured on each use of such credentials. Then even if tampering with the user device results in full access to the information stored on it, this does not lead to weakening the security guarantees nor compromises the biometric, even in the event of collusion of multiple participants.

The scope of this work could not cover all schemes for biometric key generation and could not provide their thorough analysis with respect to security and re-usability. Thus, we leave it to future work to analyze the security of different BKG constructions and different metric spaces and determining which constructions would provide the best security guarantees for a particular type of biometric data.

Acknowledgments

This work benefited from discussion regarding biometrics with Karen Hollingsworth, Patrick Flynn, and Kevin Bowyer.

References

1. Arakala, A., Jeffers, J., Horadam, K.: Fuzzy extractors for minutiae-based fingerprint authentication. In: Lee, S.-W., Li, S.Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 760–769. Springer, Heidelberg (2007)
2. Bakhtiari, A., Shirazi, A., Zamanlooy, B.: An efficient biocryptosystem based on the iris biometrics. In: Mery, D., Rueda, L. (eds.) PSIVT 2007. LNCS, vol. 4872, pp. 334–345. Springer, Heidelberg (2007)
3. Ballard, L., Kamara, S., Reiter, M.: The practical subtleties of biometric key generation. In: USENIX Security Symposium, pp. 61–74 (2008)
4. Blanton, M., Aliasgari, M.: Secure computation of biometric matching. Technical Report 2009–03, Department of Computer Science & Engineering, University of Notre Dame (2009)
5. Bleumer, G.: Biometric yet privacy protecting person authentication. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 99–110. Springer, Heidelberg (1998)
6. Boudot, F.: Efficient proofs that a committed number lies in an interval. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 431–444. Springer, Heidelberg (2000)
7. Boyen, X.: Reusable cryptographic fuzzy extractors. In: ACM Conference on Computer and Communications Security (CCS 2004), pp. 82–91 (2004)
8. Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., Smith, A.: Secure remote authentication using biometric data. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 147–163. Springer, Heidelberg (2005)
9. Bresson, E., Stern, J.: Proofs of knowledge for non-monotone discrete-log formulae and applications. In: Chan, A.H., Gligor, V.D. (eds.) ISC 2002. LNCS, vol. 2433, pp. 272–288. Springer, Heidelberg (2002)

10. Bringer, J., Chabanne, H., Cohen, G., Kindarji, B., Zemor, G.: Optimal iris fuzzy sketches. In: IEEE BTAS, pp. 1–6 (2007)
11. Camenisch, J.L., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)
12. Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer, Heidelberg (2002)
13. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
14. Camenisch, J., Michels, M.: Proving in zero-knowledge that a number is the product of two safe primes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 107–122. Springer, Heidelberg (1999)
15. Camenisch, J., Stadler, M.: Proof systems for general statements about discrete logarithms. Technical Report No. 260, ETH Zurich (1997)
16. Chaum, D., Evertse, J.-H., van de Graaf, J.: An improved protocol for demonstrating possession of discrete logarithms and some generalizations. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 127–141. Springer, Heidelberg (1988)
17. Clancy, T., Kiyavash, N., Lin, D.: Secure smartcard-based fingerprint authentication. In: ACM SIGMM Workshop on Biometrics Methods and Applications, pp. 45–52 (2003)
18. Daugman, J.: How iris recognition works. IEEE Transactions on Circuits and Systems for Video Technology 14(1), 21–30 (2004)
19. Davida, G., Frankel, Y., Matt, B.: On enabling secure applications through off-line biometric identification. In: IEEE Symposium on Security and Privacy, pp. 148–157 (1998)
20. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM Journal of Computing 38(1), 97–139 (2008)
21. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004)
22. Dodis, Y., Yampolskiy, A.: A verifiable random function with short proofs and keys. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 416–431. Springer, Heidelberg (2005)
23. Hao, F., Anderson, R., Daugman, J.: Combining crypto with biometrics effectively. IEEE Transactions on Computers 55(9), 1081–1088 (2006)
24. Impagliazzo, R., Miner More, S.: Anonymous credentials with biometrically-enforced non-transferability. In: ACM Workshop in Privacy in the Electronic Society (WPES 2003), pp. 60–71 (2003)
25. Juels, A., Sudan, M.: A fuzzy vault scheme. In: International Symposium on Information Theory (2002)
26. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: ACM Conference on Computer and Communications Security, pp. 28–36 (1999)
27. Lee, S., Moon, D., Jung, S., Chung, Y.: Protecting secret keys with fuzzy fingerprint vault based on a 3d geometric hash table. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4432, pp. 432–439. Springer, Heidelberg (2007)
28. Lee, Y.J., Bae, K., Lee, S.J., Park, K.R., Kim, J.: Biometric key binding: Fuzzy vault based on iris images. In: Lee, S.-W., Li, S.Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 800–808. Springer, Heidelberg (2007)

29. Nagar, A., Chaudhury, S.: Biometrics based asymmetric cryptosystem design using modified fuzzy vault scheme. In: International Conference on Pattern Recognition (ICPR 2006), pp. 537–540 (2006)
30. Nandakumar, K., Jain, A., Pankanti, S.: Fingerprint-based fuzzy vault: Implementation and performance. IEEE Transactions on Information Forensics and Security 2(4), 744–757 (2007)
31. Nisan, N., Ta-Shma, A.: Extracting randomness: A survey and new constructions. Journal of Computer and System Sciences 58, 148–173 (1999)
32. Pedersen, T.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)
33. Simoens, K., Tuyls, P., Preneel, B.: Privacy weaknesses of biometric sketches. In: IEEE Symposium on Security and Privacy, pp. 188–203 (2009)
34. Uludag, U., Pankanti, S., Jain, A.K.: Fuzzy vault for fingerprints. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 310–319. Springer, Heidelberg (2005)
35. Uludag, U., Pankanti, S., Prabhakar, S., Jain, A.: Biometric cryptosystems: Issues and challenges. Proceedings of the IEEE 92(6), 948–960 (2004)
36. Yang, S.: Secure fuzzy vault based fingerprint verification system. In: Asilomar Conference on Signals, Systems, and Computers, vol. 1, pp. 577–581 (2004)
37. Yang, S., Verbauwhede, I.: Automatic secure fingerprint verification system based on fuzzy vault scheme. In: ICASSP, pp. 609–612 (2005)