



Yale University
Department of Computer Science

**Efficient and Privacy-Preserving Biometric
Authentication**

Ewa Syta David Wolinsky Michael J. Fischer
Abraham Silberschatz Bryan Ford
Gina Gallegos-García

YALEU/DCS/TR-1469
November 14, 2012

Efficient and Privacy-Preserving Biometric Authentication*

Ewa Syta[†] David Wolinsky[†] Michael Fischer[†] Abraham Silberschatz[†]
Bryan Ford[†] Gina Gallegos-García[‡]

Abstract

Biometric authentication offers many benefits ranging from strong security guarantees to user convenience, but remote authentication poses unique challenges to the security and privacy of biometric templates. Current schemes use either unprotected biometric templates, making them directly available to the verifying party, or protected templates lack adequate privacy protection, revocability, or use techniques which impact the recognition performance.

We propose an efficient remote authentication protocol that combines possession-based authentication and biometrics to protect users' privacy. Since users cannot change their biometric characteristics, as they can easily do in case of passwords or PINs, our protocol protects biometric templates by never directly storing or transmitting them during authentication. The protocol uses a token to store the user's blinded biometric template which changes with each authentication, rendering the information stored on the token useless if stolen. A user successfully authenticates if the verifying party confirms that the difference between the blinded template and a fresh template as computed by the token is sufficiently close to 0.

Our approach offers benefits such as protection of biometric data, revocability of templates, and privacy-protection with respect to users' biometric identities as well as actions performed using those identities. Furthermore, our protocol adds negligible overhead and maintains the recognition performance of the underlying recognition algorithm, which we validate using the CASIA Iris Image Database and two different open source iris recognition libraries.

1 Introduction

The fast pace of technological advances bringing faster and cheaper computing devices and nearly constant Internet access have changed the ways in which people utilize online services. To provide security in this ever connected world, almost every transaction performed requires identity verification to ensure only legitimate access to protected resources. There have been a number of authentication solutions proposed, which can be categorized as knowledge-based, possession-based, or biometrics depending on the factor used to verify

*This work builds on the work reported in TR1455 by Ewa Syta, Michael J. Fischer, Abraham Silberschatz, Gina Gallegos-García and Bryan Ford [60].

[†]Department of Computer Science, Yale University, CT

[‡]Department of Computer Science, National Polytechnic Institute of Mexico

the claim of identity [6]. While all three categories of methods have been extensively employed over the years, biometrics offer exceptional benefits including high level of assurance that users are who they claim to be, non-repudiation and ease of use. Because biometric data cannot be lost or forgotten and is constantly available, and biometric characteristics are unique to a person, they provide an excellent way to define user’s identity. However, the uniqueness of biometric data poses serious security and privacy concerns if it is ever compromised.

Many biometric authentication protocols exist. However, frequently the protection of biometric data is achieved by assuming a trusted verified party or with performance and complexity costs [51, 5, 55, 29]. This paper proposes a novel remote biometric authentication protocol, which limits the exposure of biometric data, is theft-resistant with respect to tokens storing authentication information, privacy-preserving with respect to users’ biometric identities and the actions performed using those identities, as well as very efficient. The protocol retains its security properties under a complete compromise of either the proving or verifying parties, though not both simultaneously. The novelty of this protocol lies in the way biometric data is handled. Biometric templates are *never* directly stored, transmitted or made available to the verifying party. This approach relaxes the trust model typically required for biometric protocols, which fully entrust the verifying party with biometric templates.

The protocol builds upon two different kinds of authentication methods, possession-based authentication and biometrics, to take advantage of the benefits of two factor authentication. During the enrollment phase, the proving and verifying parties exchange a secret seed, later used to create the same sequence of blinding factors using a *backtracking resistant* pseudorandom bit generator. The blinding factors secure the biometric reference template with the resulting blinded template stored on the user’s token. During verification, a user obtains a new template and combines it with her blinded reference template in a way that results in their difference. The verifying party makes his decision based on this difference, rather than on the templates themselves. The smaller the difference, the more likely the user is who she claims to be. Since the user’s identity consists of biometric data combined with with blinding factors, multiple, independent identities called *personas* can be securely established based on the same biometric identity.

Our protocol provides for efficient authentication. We validated this claim using two different open source iris recognition libraries and compared the recognition performance of our protocol against using unprotected iris-based templates. Our approach produces identical comparison results and induces negligible overhead in comparison to the cost of producing a biometric template. As a result, our protocol has nearly the same efficiency as protocols offering no template protection.

The rest of the paper is structured as follows. Section 2 gives an overview of security and privacy issues in biometric systems, and summarizes other solutions to biometric authentication. Section 3 details our protocols and Section 4 analyzes its security properties. Section 6 provides both implementation details and evaluates the performance of our protocol. Section 7 concludes.

2 Background and Related Work

2.1 Security and Privacy Issues

The perception and acceptance of biometric systems significantly depends on the security of biometric data [2, 21]. However, the uniqueness of biometric data, a cherished feature of biometrics, is also the source of security and privacy concerns. A biometric template derives from characteristics, which uniquely identify an individual, and unlike passwords and other knowledge-based factors cannot be changed. The template has the user’s identity “embedded” into it and therefore there are limited defenses in case of compromise [52]. Typically, the proving party makes the biometric template available to the verifying party for the purpose of comparison. In case of remote authentication, this poses a risk of serious attacks in which biometric data is intercepted during transmission, stolen from the verifying party or even misused by the verifying party.

From the security point of view, once compromised, biometric data has limited utility for authentication purposes as it might be used for identity theft. The privacy issues are two fold. Firstly, a biometric template, in addition to defining a user’s identity, carries considerable personal information, which often include race, gender and certain medical conditions [46]. Secondly, a biometric template can be used to identify an individual and successfully track and link his or her activities performed using the same biometric identity across different verifying parties.

For these reasons, biometric templates security has become a crucial issue resulting in a high level of awareness and concern [50]. Users expect that the verifying parties protect their biometric data and use them only for the purpose provided [5], in order to prevent identity theft, information linkage across different providers, and secondary uses of supplied information [45].

2.2 Related Work

Our protocol is a hybrid approach which merges two-factor authentication (possession-based and biometrics) and a template transformation technique which belong to a class of template protection systems.

2.2.1 Template Protection Systems

There are two main categories of schemes for protecting templates: biometric cryptosystems and template transformation [29, 47].

Biometric cryptosystems (BC) [19] use a template as well as helper data to extract a cryptographic key, with the resulting key validated by verifying its correctness. Helper data generally consists of a biometric template (secure sketches and fuzzy extractors [19, 18]) and optionally an external key (fuzzy vaults [33] and fuzzy commitments [34]). The helper data in BC systems, however, unavoidably leaks data [26, 20]. BC techniques heavily rely on error correction codes limiting their recognition performance to the error-correcting capability of employed code [29, 55]. Furthermore, BC has not been designed with reusability and revocability in mind [29, 47]. Attacks on multiple records in BC may lead to a full recovery of the secret key and/or the biometric template [10, 57, 58]. To achieve reusability and

revocability, BC schemes must be strengthened by adding auxiliary information, for example passwords [49, 3]. This adds to their complexity, limits user convenience, and in some cases may still be insufficient [25]. While BC offers additional features such as reliable cryptographic key generation, they come at the cost of performance and complexity.

Template transformation schemes use a transformation function, either invertible (Bio-Hashing [32]) or non-invertible (cancelable biometrics [54]), and applies it to biometric data during the enrollment phase. Verification applies the same transformation and compares the resulting template against the reference template. In case of invertible transformations, users need to supply, and therefore remember or keep secure, a password or a key which impacts user’s convenience. A compromise of this additional information can yield further vulnerabilities [37, 43]. In case of non-invertible transformations, the recognition performance is affected because the matching is applied to transformed templates [55], however, revocability and unlinkability can be achieved [29, 47]. Finally, it has been shown that in some cases it is possible to recover biometric data from transformed biometric templates [24, 1, 56]. Additionally, both schemes are vulnerable to intrusion and linkage attacks using information recovered from transformed templates [48].

2.2.2 Two-factor Schemes

Combining biometric and possession-based authentication is a popular approach to remote biometric authentication. Some schemes, in addition to biometrics and a smart card, require additional knowledge-based authentication factors. An early scheme combined biometrics with a smart card and a password [38]. However, this scheme succumbed to masquerade [42] and conspiring [11] attacks. A later scheme [42] remained vulnerable to server spoofing attacks [36]. The scheme was further improved by [39] but it requires to keep a secret system’s key. Another, more efficient scheme [40] enabled users to change their passwords and removed the requirement of a synchronized clock between the proving and verifying parties. The scheme, however, did not to provide proper authentication and was susceptible to the man-in-the-middle attacks [41]. The resulting scheme was broken and then again improved upon by [31].

3 Protocol Description

In this section we describe the trust model, detail our protocol and specify the required cryptographic primitives.

3.1 Trust Model

The authentication process is performed between a proving party (Peggy, the user) and a verifying party (Victor, the authentication server). Peggy and Victor interact over a network possibly in the presence of a computationally bounded adversary (Mallory, the malicious adversary).

Peggy’s goal is to convince Victor of her identity by providing *sufficient* information to prove her claim of identity. In a typical case of a biometric remote authentication protocol, the verifying party needs to know the reference template in order to compare it with the

template submitted during each authentication request. For this reason it must be assumed that templates are transmitted over a network. Furthermore, the verifying party must be fully trusted to adequately store, protect, and not to misuse the templates.

Our goal is to relax this assumption. In our protocol, Victor does not need nor have access to an unprotected template at any point. During each authentication he receives sufficient amount of information about Peggy’s biometric data required to make a valid decision about her identity. Since Victor does not have direct access to Peggy’s biometric template, he need not be fully trusted. However, he needs to protect his internal authentication state because it can be used to impersonate Peggy to gain access to his system. This is a reasonable assumption as Victor is ultimately responsible for protecting access to his own resources. We further assume that Victor successfully authenticates Peggy whenever she sufficiently proves her identity.

3.2 Enrollment Phase

A biometric authentication protocol consists of two phases, the *enrollment* phase and the *verification* phase [14]. The enrollment phase is a one-time process while the verification phase occurs each time Peggy wants to prove her identity to Victor.

During the enrollment phase (Figure 1) Peggy and Victor must cooperate to create Peggy’s credentials and establish the shared authentication information. The public information includes the choice of a biometric characteristic, a feature extractor and an appropriate matching metric. Our protocol makes use of biometric template in binary form that uses an exclusive-OR (XOR) for matching two templates with the Hamming distance between the two to produce a difference score: the closer to 0 the more likely the two templates match. Section 6 discusses our iris-based implementation of the protocol which meets these requirements.

Our protocol is a two-factor scheme. Therefore, in addition to her biometric sample, Peggy needs to obtain a token which she will use to store her protected biometric template and authentication information. Section 5.4 discusses the issue of obtaining tokens. Peggy and Victor need to decide on an appropriate pseudorandom bit generator G and securely establish a shared secret z of the length m to seed the generator. The generator consists of a tuple $G = (m, n, S, \iota, \delta, \rho)$, which defines the length of the seed, the length of the output sequence, the finite set of states of the generator, the initial-state function, the next-state function, and the output function respectively. We assume a cryptographically secure and backtracking resistant incremental pseudorandom bit generator. Details and requirements for the generator as well as possible solutions for a secure seed exchange are given in Section 3.5.

After agreeing on the secret seed z , Peggy and Victor initialize their generators using the seed. Peggy commits to her token by providing a biometric sample to generate a reference template (Section 5.2 discusses template generation). Then, she blinds the reference template with the first blinding factor generated using G . This process binds Peggy’s biometric identity to the secret established with Victor. Peggy can obtain a biometric sample using an external sensor or a sensor built into the token depending on the kind of token she chose as described in Section 5.4. Algorithm 1 shows the steps of the enrollment process in detail. We assume the enrollment phase concludes with Peggy in possession of a

Algorithm 1 Enrollment Phase

1. Peggy obtains a token. Peggy and Victor agree on non-secret authentication information: the choice of $G = (m, n, S, \iota, \delta, \rho)$.
2. Peggy and Victor securely exchange a random seed $z \in \{0, 1\}^m$.
3. Peggy and Victor initialize their generator G and obtain the initial state $s_0 = \iota(z)$.
4. Victor sets $R_0 = r_0$, where $r_0 = \rho(s_0)$ and keeps $s_1 = \delta(s_0)$, the next state of G .
5. Peggy obtains a biometric template P and creates a blinded template $T_0 = P \oplus r_0$, where $|P| = |r_0|$, $r_0 = \rho(s_0)$ and \oplus is a bit-wise exclusive-OR operation.
6. Peggy securely erases z , P and r_0 , and keeps the blinded template T_0 and the next state of G , $s_1 = \delta(s_0)$. Victor securely erases z .

After the enrollment phase:

- Peggy's token stores $T_0 = P \oplus r_0$ and s_1 .
 - Victor retains $R_0 = r_0$ and s_1 .
-

token bound to her biometric identity. The token stores her blinded biometric template, the current state of the generator as well as other non-secret authentication information (for example, Peggy's identifier, information required by the underlying biometric recognition protocol, etc.). Additionally, we assume that the secret seed z is securely erased on both sides, Peggy's unprotected template and the first blinding factor r_0 are securely erased as well, and none of the secret information (z, P, r_0) have been compromised.

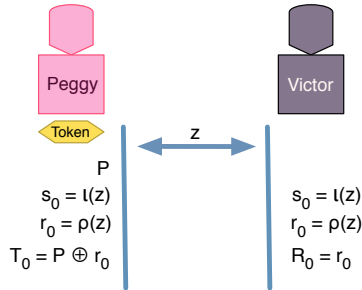


Figure 1: Enrollment Phase

3.3 Verification Phase

Our protocol requires two authentication factors in order to prove a claim of identity. Therefore, Peggy must be in possession of the token issued to her upon the enrollment into the system and obtain a fresh biometric sample. A feature extractor can use the biometric sam-

ple to produce a biometric template for verification. We assume that the produced template is of good quality and is a representation of an appropriate biometric characteristic. More specifically, we discard every template which does not meet these requirements to ensure that only an actual template is used and not, for example, an empty or random binary vector.

In order to authenticate, Peggy obtains a new biometric sample and generates a fresh template P'_i . Then, she uses her token to recover the blinded reference template T_{i-1} and calculates an authentication message $W_i = P'_i \oplus T_{i-1}$. The process can continue up to k times, where k depends on n , the number of bits that can be safely generated using G . Peggy sends W_i , the blinded difference between the two biometric templates, to Victor for verification. After each authentication attempt, successful or not, she updates the blinded verification template stored on the token with a newly generated blinding factor to ensure that the same sequence of blinding factors is never used twice. Algorithm 2 details the steps Peggy performs during verification.

Algorithm 2 Verification Phase: Steps performed by Peggy

1. Peggy obtains a biometric sample and generates a fresh biometric template P'_i .
 2. Peggy calculates calculate $W_i = T_{i-1} \oplus P'_i$ and sends W_i to Victor.
 3. Peggy updates the blinded reference template T : $T_i = T_{i-1} \oplus r_i$, where $r_i = \rho(s_{i-1})$.
 4. Peggy securely erases P', W_i, T_{i-1} and r_i .
-

Upon receiving W_i from Peggy, Victor unblinds the difference between the two templates using R_{i-1} . Victor computes $V_i = R_{i-1} \oplus W_i$ checks if $V_i \approx 0$. If the authentication attempts is successful, Victor updates his R_{i-1} by adding the next blinding factor generated by G . Algorithm 3 specifics Victor's steps during verification.

Assuming that Peggy's and Victor's generators are in sync,

$$\begin{aligned}
 R_{i-1} &= r_0 \oplus \dots \oplus r_{i-1} \\
 T_{i-1} &= r_0 \oplus \dots \oplus r_{i-1} \oplus P \\
 W_i &= T_{i-1} \oplus P'_i \\
 V_i &= R_{i-1} \oplus W_i \\
 V_i &= r_0 \oplus \dots \oplus r_i \oplus P \oplus r_0 \oplus \dots \oplus r_{i-1} \oplus P'_i = P \oplus P'_i, \\
 V_i &= \Delta(P'_i, P).
 \end{aligned}$$

If at any point Victor and Peggy get out of sync, Victor's generator can catch up to Peggy's using a simple approach sketched in Section 5.3.

We write $\Delta(P, P'_i)$ to denote the difference between P and P'_i using the difference function Δ . While Victor learns $\Delta(P, P'_i)$ the individual templates are never available to Victor. Because V_i is a binary vector, Δ can be expressed as the Hamming distance of V_i . The lower the Hamming distance, then the closer the difference is to 0.

Victor's goal is to establish whether the authentication message W_i came from Peggy. If two templates are created based on a biometric sample from the same user, they will be

very similar. Therefore, if $\Delta(P, P'_i)$ is sufficiently “small” according to a security threshold τ ($\Delta(P, P'_i) < \tau$) then authentication succeeds and Victor accepts Peggy’s claim of identity. The parameter τ defines how “small” or how close to 0 the difference between two samples should be. We write $\Delta(P, P'_i) \approx 0$ to express that the difference between P and P'_i is sufficiently close to 0 in terms of the parameter τ , that is $\Delta(P, P'_i) \approx 0$ if $\Delta(P, P'_i) < \tau$.

Section 3.4 further discusses the issue of making the verification decision.

Algorithm 3 Verification Phase: Steps performed by Victor

1. Victor calculates $V_i = W_i \oplus R_{i-1}$.
 2. Victor Verifies that $V_i \approx 0$ and if yes, accept Peggy’s claim of identity.
 3. If verification succeeded, he updates R : $R_i = R_{i-1} \oplus r_i$, where $r_i = \rho(s_{i-1})$.
 4. Victor securely erase W_i, R_{i-1} and r_i .
-

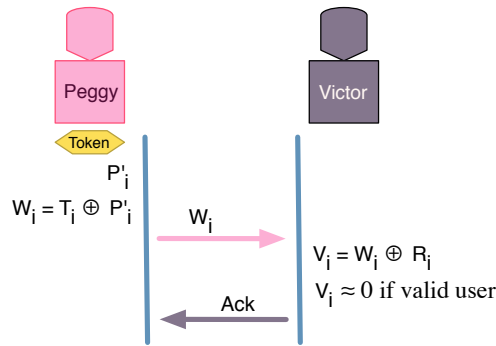


Figure 2: Verification Phase

3.4 Verification Decision

In biometric systems, a matcher and a decision module are the two components directly involved in making the verification decision [30]. A matcher takes two biometric templates created using a feature extractor, the reference template created in the enrollment phase and the freshly obtained template from the verification phase, as input. Then, it calculates a *match* score which shows how similar the two templates are [28]. The In case of our protocol, the matcher functionality is embedded into the protocol. When Victor unblinds Peggy’s authentication message, he receives a difference of two templates under an appropriate difference function Δ which in our case is an exclusive-OR. Therefore, Victor can apply a proper metric to $V_i = \Delta(P, P'_i)$ to evaluate the difference between two biometric templates. Because V_i is a binary vector and δ is an exclusive-OR operation, the Hamming distance (or alternatively a fractional Hamming distance) is a proper metric to obtain the *difference score* between two binary templates.

In other biometric authentication protocols, the authentication decision is based on the match score while in our it is based on the difference score. To express the difference score in terms of the match score we can say that the smaller the difference of V_i , the higher the match score is.

The decision module takes a match score (in our case a difference score) as input and based on a predefined threshold parameter τ decides whether the two templates were created based on biometric samples from the same person. If the match score is greater than a predefined threshold τ , user's identity is verified. In our protocol, if the difference score is lower than τ , then the two templates are accepted as coming from the same user.

Choosing a proper value for τ is a challenging task. To have a high level of confidence that two templates were created based on samples from the same user, the difference should be very low. Hence, the value chosen for τ should reflect the desired level of security as well as the sensor and feature extractor's capabilities to create accurate templates. The goal is to balance the false rejection (FRR) and false acceptance (FAR) rates while ensuring a proper level of security and user experience.

In case of our protocol, the template comparison function is an exclusive-OR operation as the templates are binary vectors. Therefore, a low difference score will correspond to a low Hamming distance of the resulting vector V_i . Section 5.2 contains a more detailed discussion of the meaning of different values of the difference score for a concrete example of iris-based templates.

3.5 Cryptographic Primitives

Secure seed exchange

The protocol relies on the parties' ability to establish a secret seed for pseudorandom bit generation. The seed needs to be established in a secure manner in order to keep the sequences of blinding factors secret and the blinded template secure.

The secret seed can be exchanged using one of the schemes to establish a shared secret, for example a key agreement protocol [23]. Alternatively, the seed can be sent through a secure channel.

Pseudorandom number generation

A pseudorandom bit generator (PRBG) (sometimes called a deterministic random bit generator (DRBG) [4]), is a deterministic polynomial-time algorithm G that maps a random seed z of length m to an output string r of length $n > m$.

In practice, pseudorandom bits are generated on demand and the output string is built incrementally.

An *incremental PRBG (iPRBG)* G is defined by a tuple $(m, n, S, \iota, \delta, \rho)$. m is the length of the seed, n is the length of the output sequence, S is a finite set of *states* of the generator, and ι , δ , and ρ are functions. The *initial-state function* ι maps a seed to an initial state $s_0 \in S$. The *next-state function* δ is a permutation on S . The output function ρ maps S to $\{0, 1\}$. Starting with a seed $z \in \{0, 1\}^m$, G computes a sequence of states s_0, s_1, \dots, s_n and a sequence of bits $r_0 r_1 \dots r_{n-1}$, where $s_0 = \iota(z)$, $s_k = \delta(s_{k-1})$ for $1 \leq k \leq n$, and $r_k = \rho(s_k)$ for $0 \leq k \leq n - 1$. The output $G(z) = r_0 r_1 \dots r_{n-1}$.

In case of an incremental PRBG, its state can be explicitly defined, saved, and later on retrieved to resume the generation of random bits. Any pseudorandom bit generator can be easily expressed as an iPRBG. For example, a Blum-Blum-Shub (B.B.S.) [7] which is an elegant and provably secure pseudorandom bit generator. Algorithm 4 shows how it can be adapted as an iPRBG.

Algorithm 4 Blum-Blum-Shub iPRBG

Input: Random seed $s \in_R [1, N - 1]$, where N is a product of two sufficiently large Blum primes.

Output: $r_0, r_1, r_2 \dots, r_{n-2}$

1. Initial state: $s_0 = \iota(s) = s^2 \bmod N$.
 2. Next-state: $s_i = \delta(s_{i-1}) = s_{i-1}^2 \bmod N$.
 3. Output bit: $r_i = \rho(s_i) = \text{parity}(x_i)$.
-

We require the pseudorandom bit generator to be *cryptographically secure* and *backtracking resistant*.

To be *cryptographically secure*, the ensemble of output strings $G(U_m)$ should be computationally indistinguishable from U_n , where U_m and U_n are the uniform distributions over strings of length m and n , respectively. The notion of computational indistinguishability, introduced by Yao [63], means that any probabilistic polynomial-time algorithm behaves essentially the same whether supplied with inputs from the one distribution or the other. See Goldreich [22] for further details.

A *backtracking attack* applies to an iPRBG G whose internal state has been compromised. We assume that an adversary compromises the internal state of G at stage i after r_{i-1} has been produced and the internal state has been replaced by s_i . Informally, we say that G is *backtracking resistant* if the bit string $r_0 \dots r_{i-1}$ is judge-indistinguishable from a truly random string of the same length. This implies that a polynomially-bounded adversary has only a negligible advantage at guessing any of the bits produced by G before the attack.

Backtracking resistance implies that knowing a state of the generator gives the adversary no useful information about the previous output bits. This property is needed to prevent an adversary who gains access to the token at stage i from recovering the sequence of blinding factors (specifically the last blinding factor r_{i-1}) that protects the blinded reference template P . The adversary gains all of the information stored on the card at the time of the attack, including the state s_i of G . We also have to assume that the adversary might have obtained r_0, \dots, r_{i-2} from observing and using the values W_i going over the channel. This information might be used for a backtracking attack to be carried out in an attempt to recover r_{i-1} .

While it seems likely that many cryptographically strong pseudorandom number generators are resistant to a previous-outputs backtracking attack, we are not aware of any such generator that has been proven to enjoy this property.

4 Security Properties

Our main goal and concern is the security of biometric data, not only under normal use of the protocol, but also in case of complete compromise of either Peggy's token or Victor's entire internal state. In addition, our protocol prevents an attacker who compromises Peggy's token from impersonating her to Victor.

We note that if an attacker compromises both Peggy and Victor then Peggy's biometric template is easily obtained. Peggy's token contains her blinded template. Victor has the information needed to unblind the difference between Peggy's reference and sample templates, but this same information will also unblind the template stored on the token.

4.1 Assumptions

We assume that all communication occurs over an unsecured channel and Mallory can record all messages sent between Peggy and Victor. Therefore, after k authentication attempts, Mallory will see a series of authentication messages W_1, \dots, W_k , which are blinded differences between pairs of biometric templates. Thus, Mallory has the following information.

$$\begin{aligned} W_1 &= P'_1 \oplus T_1 = P'_1 \oplus P \oplus r_0 \\ W_2 &= P'_2 \oplus T_2 = P'_2 \oplus P \oplus r_0 \oplus r_1 \\ W_3 &= P'_3 \oplus T_3 = P'_3 \oplus P \oplus r_0 \oplus r_1 \oplus r_2 \\ &\dots = \\ W_k &= P'_k \oplus T_k = P'_k \oplus P \oplus r_0 \oplus \dots \oplus r_{k-1} \end{aligned}$$

The security of our protocol depends critically on the pseudorandom bit generator. We assume that the secret seed z and the original unprotected template P are securely erased after enrollment and are not available to Mallory. We assume the generator is cryptographically secure and backtracking resistant, and that Peggy and Victor never use more than n pseudorandom bits, where n is the maximum length of the output sequence of G . Thus, Peggy never reuses the same blinding factors.

Furthermore, we assume that the sensor Peggy uses to obtain biometric samples does not directly reveal her biometric data to Mallory, prior to his possible compromise of Peggy's token. We also assume that Peggy does not use her token after it has been compromised. Similarly, we assume that the communication channel between the sensor and the token is trusted.

We assume that at the time of compromise, Victor is in possession of only the blinding factor R_i and the next state of the generator s_{i+1} . Similarly, Peggy's token stores only her blinded reference template T_i and the next state of the generator s_{i+1} . We further assume that Peggy's token is not compromised at the moment she is using it, since for a brief interval, the token contains her unprotected biometric template as well as data from both stage $i - 1$ and stage i .

4.2 Security of Biometric Templates

4.2.1 Mallory compromises Victor

If Mallory compromises Victor, she gets the current blinding factor R_i and the next state of the generator s_{i+1} in addition to the sequence of messages sent up to this point W_1, \dots, W_{i-1}

and future messages W_i, \dots, W_k . Knowing the next state of G , Mallory can obtain the future blinding factors too $r_{i+1}, r_{i+2}, \dots, r_{k-1}$.

$$\begin{aligned}
W_i &= P' \oplus T_{i-1} = P' \oplus P \oplus r_0 \oplus \dots \oplus r_{i-1} \\
W_{i+1} &= P' \oplus T_i = P' \oplus P \oplus r_0 \oplus \dots \oplus r_{i-1} \oplus r_i \\
W_{i+2} &= P' \oplus T_{i+1} = P' \oplus P \oplus r_0 \oplus \dots \oplus r_{i-1} \oplus r_i \oplus r_{i+1} \\
\dots &= \\
W_k &= P' \oplus T_{n-1} = P' \oplus P \oplus r_0 \oplus \dots \oplus r_i \oplus \dots \oplus r_{k-1}
\end{aligned}$$

By knowing the future blinding factors, Mallory will be able to unblind messages W_{i+2}, \dots, W_k . This way Mallory recovers exact same information that Victor receives from Peggy, the difference between two templates.

4.2.2 Mallory compromises Peggy

When Mallory obtains access to Peggy's token, she learns the current blinded reference template

$$T_i = P \oplus r_0 \oplus r_1 \oplus \dots \oplus r_{i-1} \oplus r_i$$

and the next state of the generator s_{i+1} . This new information is in addition to all authentication messages W_1, \dots, W_i sent up to that point which we assume she already knew.

T_i looks random to Mallory because of the blinding factor r_i , which he does not know. It was securely erased from the token when T_i was updated, and it was never included in any of the messages sent. Additionally, r_i cannot be recovered using the stored state s_{i+1} of G and r_0, \dots, r_{i-1} (assuming they are known) since G is backtracking resistant and cryptographically secure. Thus, neither T_i nor s_{i+1} give Mallory any additional useful information about P .

4.3 Impersonation

4.3.1 Mallory compromises Victor

As before, when Mallory compromises Victor, she gets R_i and s_{i+1} . R_i is the sequence of all blinding factors needed to unblind the next authentication message. Therefore, Mallory can prepare a fake message W'_{i+1} that is close to R_i so that verification will succeed from Victor's point of view as follows. Assume that $W' \approx R_i$. Then, $V_{i+1} = R_i \oplus W'_{i+1} \approx 0$ and Victor will successfully authenticate Mallory.

4.3.2 Mallory compromises Peggy

In order to impersonate Peggy, Mallory must send a message W'_{i+1} that Victor will accept as valid. Looking at Victor's protocol, this will happen exactly when $W'_{i+1} \approx R_i$.

If Mallory has also compromised Peggy's token, he knows T_i and s_{i+1} . Since $T_i = P \oplus R_i$, this would enable him to find Peggy's unblinded template P . But we argued in Section 4.2 that Mallory's compromise of Peggy's token is insufficient for him to obtain her template, so it must also be insufficient for him to find a good approximation of the blinding factor R_i .

4.4 Leakage of Information

During each authentication attempt, Victor receives a difference between two biometric templates. If he has been compromised, then Mallory also receives this information. Unlike the case of a compromise of Peggy, we assume that the compromise of Victor might be undetected so that Mallory can collect data over time from legitimate authentication requests.

After a number of such authentications, Mallory has a set of differences between Peggy's templates. Those differences are binary vectors of differences between Peggy's reference template and the sample template used on a given authentication. A difference bit of 1 indicates a discrepancy between the reference and the sample templates. In a perfect biometric system, the differences would always be zero.

The frequency of 1's in any given bit position represents the unreliability in that position of the template. A low-frequency position indicate a reliable bit; a high frequency position means that little useful information is being carried by that bit. Mallory can compute these frequencies and thereby learn about the reliability of each bit in the template. What information these frequencies carry about the actual reference template or Peggy's raw biometric data depends in detailed properties of the sensor as well as the feature extraction algorithm. Analyzing this kind of information leakage for any particular sensor and feature extractor is beyond the scope of this paper, it is well to keep in mind this possibility in designing biometric systems.

Low-frequency bits can arise equally well from 0's in both reference and sample templates or from 1's in both, so knowing that it is low frequency says little about the actual template bit. With a good feature extractor, we expect most difference bits to be low-frequency, so information leakage would seem to be minimized with good quality biometric systems.

5 Usage Considerations

5.1 Suitable Biometrics

There are two main categories of biometric characteristics used in biometric systems: physiological (e.g., a fingerprint or iris pattern) and behavioral (e.g., voice print or signature). Characteristics must be universal (everyone has it), unique (different for every person), permanent (it does not change with time), and collectable (it can be quantitatively measured) [14]. In practice, fingerprints, face geometry, and iris patterns have been popular choices as they can be obtained easily and non-intrusively using a simple camera. Fingerprints tend to be prone to spoofing, however, and the accuracy of facial recognition may be impacted by pose, expression, or lighting [16, 28]. An iris, on the other hand, exhibits many highly desirable properties. Its pattern varies greatly among different people, even identical twins, and persists over a lifetime. The iris tend to be relative easy to locate and isolate to create a size-invariant representation, making them well-suited for biometric systems. Iris-based recognition systems have been widely deployed by many organizations including British Telecom, US Sandia Labs, UK National Physical Lab, NBTC, Panasonic, LG, Oki, EyeTicket, and IBM SchipholGroup [15, 9].

5.2 Template Generation and Matching

A feature extractor obtains a biometric sample in order to generate a biometric template. The feature extraction process identifies the key features of the biometric sample and encodes them in the form of a template. The template quality directly impacts the performance of a biometric system [30]. Our protocol preserves the the recognition performance of the underlying feature extractor. Therefore feature extractors should produce templates of high quality, such that sample from the same individual should be “sufficiently” similar to be suitable for authentication purposes, an acceptable false rejection rate (FRR). Likewise, two templates created from two different users should be “sufficiently” different, ensuring a low false acceptance rate (FAR).

An iris produces a template using an iris pattern represented as a binary vector. Fingerprint templates use the fingerprint texture as a real-valued fixed length vector. Finally facial features use facial features represented again as a real-valued fixed length vector. A match can be performed by calculating the hamming distance (or alternatively a fractional Hamming distance) for binary vectors, while a Euclidian distance for real-valued vectors with the points defined by the set difference. While our current protocol assumes binary biometric templates, a binarization technique [35, 12, 53] can convert data into a binary vector.

We chose to use an iris-based template for our implementation, described in Section 6. These templates typically consist of 2048 bits to represent the iris pattern with any bit equally likely to be either 1 or 0. The templates also include a mask, which can be used during the matching phase to exclude bits from calculating the score. On average half of all the bits will disagree between the templates of two different people. Therefore, the difference score expressed as a fractional Hamming distance (a fraction of all disagreeing bits to all bits) is expected to be around 0.5. [15] showed the average fractional Hamming distance of $p = 0.499$ with standard deviation $\sigma = 0.0317$ with the minimum of $p = 0.329$ and maximum of 0.546 based on 9.1 million comparisons between different pairings of iris images. [15] concludes that it is extremely improbable that two different irises might disagree fewer than at least a third of their bits. In other words, if two irises disagree on more than 30% of all bits, they likely come from two different people. Table 1 shows the relationship between a fractional Hamming distance (the difference score) and a probability of a false match. Consequently, if a difference score is less than 0.32, then a positive match is statistically “guaranteed”.

5.3 Resynchronization

Victor’s ability to verify Peggy’s identity depends on his ability to “unblind” the difference between her two templates, which requires the two pseudorandom number generators to be in sync. Unfortunately, desynchronizations can occur as a result of either a poor feature extraction by Peggy, Viktor’s not storing the updated template before going offline, or as a result of intentional malicious attempts from Mallory.

Because Peggy updates her blinded template T_u after each authentication attempt and Victor does so only after a successful authentication, if the generators are out of sync, Peggy’s generator will be ahead of Victor’s. The simple solution is for Victor to search forward in the sequence produced by G for some predefined distance n looking for a value

Difference Score	False Match
0.26	1 in 10^{13}
0.27	1 in 10^{12}
0.28	1 in 10^{11}
0.29	1 in 13 billion
0.30	1 in 1.5 billion
0.31	1 in 185 million
0.32	1 in 26 million
0.33	1 in 4 million
0.34	1 in 690,000
0.35	1 in 133,000

Table 1: The relation between the difference score and odds of a false match [15]

of T_s that is the blinding factor. After finding the correct value of T_s , both generators will be in sync. To make the process easier, Peggy can keep the stage her generator is at and send it to Victor along with her authentication message.

5.4 Tokens

There are four typical locations for storing biometric templates: portable tokens, central databases, sensors, and individual workstations [51], with the former two being the most popular. A token allows users to secure their biometric templates physically and gives them a sense of control over their personal data. However, issues arise when tokens are lost or stolen and their content is unsecured. A central database makes it possible for users to authenticate from multiple locations easily as templates are constantly available for verification. On the other hand, the database may become a target of attacks because of its valuable content. Furthermore, central storage of templates raises privacy concerns because all authentication attempts go through a single point, allowing the verifying party to track and link users' actions.

Since our protocol has been designed with privacy protection in mind, we use tokens to store biometric templates but ensure that their content is protected in case of loss or theft. There are two different approaches to utilizing tokens depending on the token's ability to obtain biometric samples.

A token with a built-in sensor

Obtaining a biometric sample is a crucial part of the authentication process. Ideally, a token has a built-in sensor, removing security concerns related to the sensor and the communication channel between the token and the sensor.

Mobile devices are an obvious choice for such tokens. Most modern phones, PDAs, or tablets, are equipped with a high resolution camera capable of capturing images suitable for authentication using several biometric characteristics such as a fingerprint, facial geometry, or iris pattern [28]. Additionally, mobile devices make it possible to take advantage of less frequently utilized characteristics like voiceprint, keystroke or handwriting patterns, service utilization [13] or even gait [17].

A token without a sensor

In this case, the token is only used to store authentication information and to perform computations. It must be paired with an external sensor to obtain a biometric sample. This implies certain level of trust that the sensor is not compromised. However, this approach makes it easy to utilize virtually any biometric characteristic.

Smart cards are the most obvious choice for such tokens. They have been extensively used for authentication and are relatively cheap, small, and convenient to use [59]. Most smart cards offer enough computational power to perform the required operations, given that the computational requirements of our protocol are modest.

5.4.1 Obtaining and Managing Tokens

The main drawback of possession-based authentication is the need to obtain and manage tokens. Eddie, an enrolling agent, can be responsible for issuing tokens and performing the enrollment phase, ensuring a successful bond between a token and a biometric identity. Depending on the application-specific security requirements, Eddie can be an independent, trusted enrollment center, Victor can assume Eddie's role, or Eddie's role can be delegated to users. In the first scenario, Eddie's services can be offered by an organization such as VeriSign [61]. This approach would provide a good way to issue and manage a variety of tokens. VeriSign already provides similar services and issues security credentials (VIP Security Token and VIP Security Card [62]).

5.5 Personas

Privacy protection stems from how the user's identity is established with the verifying party. The identity created is based on user's unprotected biometric template with respect to the blinding factors known to the verifying party. This results in two benefits. First, the verifying party does not need to have direct access to biometric data. Second, a user can create multiple identities based on the same biometric template and different blinding factors. This approach allows users to create multiple, fully independent *personas*. Each persona is based on the same biometric identity but a different secret shared with a verifying party. Therefore, a persona represents user's unique identity as seen by the verifying party. Users can create different personas to deal with multiple verifying parties, or use personas for different transactions with the same verifying party. This creates a separation and *unlinkability* of biometric identities and transaction performed using those identities. The user must perform the enrollment process once for each persona, so the policies and procedures controlling this enrollment process determine how many and what type of personas a user may acquire.

6 Evaluation

In this section, we analyze our prototype implementation to observe the performance characteristics of our protocol in comparison to using unprotected templates. We then analyze the behavior of the feature extraction libraries to determine their usefulness and potential

overheads in this scheme. We used the CASIA Iris Image Database [27] as input into our system.

6.1 Implementation

We have implemented our biometric protocol system in C++ using the Qt framework and Crypto++ cryptographic libraries. For feature extraction or obtaining biometric templates, we have employed two different Iris recognition libraries: Project Iris [8] and Libor Masek’s Iris Recognition [44] which both utilize John Daugman’s approach [15] for extracting an iris template. Project Iris also uses C++ and the Qt framework; however, for Masek’s library, we constructed a C++ to Octave¹ interface.

During enrollment, the proving party, Peggy, provides the verifying party, Victor, with a Diffie-Hellman public key and commits to a feature extraction scheme, the size of the template, and the number of samples to be transmitted during each authentication attempt. Victor obtains his shared secret with Peggy by completing the Diffie-Hellman protocol, and uses the result as input into a deterministic pseudorandom bit generator (PRBG) (AES256-CBC) to obtain the inputs into a Blum Blum Shub incremental pseudorandom bit generator (iPRBG). Furthermore, Victor computes the first blinding factor, R_0 , by considering the size of the feature extraction scheme and the number of parallel attempts. Upon storing this to a SQLite database along with a randomly generated unique identifier, Victor responds to Peggy with her unique identifier and his Diffie-Hellman public key. Peggy completes enrollment by obtaining a set of templates using the feature extraction scheme, constructing her iPRBG as Victor did and then using it to blind her template, T_0 . Finally she stores the iPRBG state, her blinded template, and her unique identifier to her smart card.

During authentication, Peggy uses the feature extractor to obtain a new template, P'_i . Using the protocol described earlier and elaborated on later in this section, she obtains the authentication message W_i , which she sends along with her unique identifier to Victor. At which point, she updates the state on her smart card including an updated blinded template, T_i . Victor unblinds her W_i using R_i , producing V_i , and calculates its difference score (fractional hamming distance). If it suggests Peggy has correctly authenticated, by scoring a difference score of .32, Victor updates his unblinding factor and sends Peggy a positive response. Otherwise Victor leaves his unblinding factor alone and sends Peggy a negative response.

In iris recognition, a single template may need to be rotated up to 8° in both the left and right directions in order to obtain an acceptable difference score. Because Victor cannot perform such operations, Peggy must do this for him by doing the actions locally and embedding multiple attempts within a single V_i . This leads to an important design decision: Peggy can send less data in V_i in order to reduce communication and computation cost at the risk of having to perform several authentication attempts. In our system, we preprocess a single template to produce these rotations. Thus Peggy’s new template need only be XORed to each of these independent blinded templates in order to obtain the same effect as a traditional template comparison would. By rotating the images during enrollment, we tradeoff a slightly larger demand for storage capacity (on the order of kilobytes) in favor of performing a rotation for each new template, P'_i .

¹Open-source Matlab compatible system

During our evaluation, we discovered some interesting behavior for both feature extraction libraries. Project Iris only supports version 1 of the CASIA database, in which images have been preprocessed by replacing the pupils with a black (constant intensity) circle. Masek’s iris recognition library handles CASIA database version 2, however, had trouble parsing approximately 4% of the images, though had no issue in database version 1. The libraries also differ in the resolution of their extracted features. While Project Iris extracts a 2048-bit template like Daugman [15], Masek extracts a 9600-bit template, the motivation for which is not clear. In evaluation figures, we denote Project Iris as C++ and Masek’s library as Octave.

All CASIA databases have already been converted to gray scale images. CASIA database version 1 contains 108 individuals with 7 images each. The 7 images were obtained in two separate sessions with 3 images in the first session and 4 in the second.

We ran the evaluations on a workstation computer equipped with a 4 core Intel Core i7-2600 with 4 cores, 16 GB of memory, and a Crucial 256GB SSD hard drive. Though our software ran in single threaded mode and never eclipsed 12 MB of used memory, the typical amount for a minimal Qt application on said machine.

6.2 System Performance

To evaluate the enrollment phase, we created a client, the prover, for each image in the CASIA database version 1 and used a single server, the verifier. Clients, in no particular order, enrolled one after another. The enrollment occurred within the same process, as a result the evaluation focuses on data processing and message serialization, i.e., CPU time. Our results can be found in Figure 3.

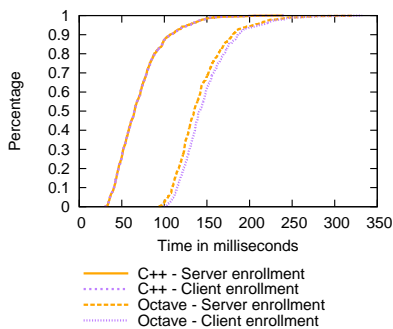


Figure 3: CPU time for enrollment

The clients enrollment time includes both the initial enrollment request and the subsequent processing for a successful enrollment, both are represented as a single, summed value. The client enrollment times is still negligibly larger than the servers enrollment time. The major factor in performance appears to be the size of the stored template(s). The iPRBG causes the performance degradation between the two approaches. While Octave, Masek’s library, uses 17 9600-bit template with 17 masks resulting in 40.8 KBs of iPRBG work, C++, Project Iris, uses only 8.7 KBs.

To evaluate authentication time, we set a minimal difference score of .32 as passing. We then had each image in the database tested against every client for a total of 571,536 authentication attempts or 756 attempts per client. We separated the results, in Figure 4, into valid and invalid client and server authentications, those that our system processed, and compared them against the time a traditional template comparison would take.

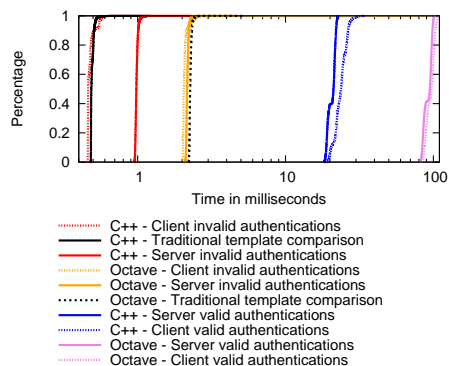


Figure 4: CPU time for authentication

The traditional template comparison takes two templates XORs them together and uses the masks to remove unhelpful bits. This process is repeated 16 more times by rotating one template both left and right 8 times. In contrast, a client and server, in our scheme, each perform only the XOR process as the rotations have been performed during enrollment.

Invalid authentications limit clients to retrieving their previous personal data from disk and performing the XOR with a newly acquired template. Valid authentications complete this process by updating the templates with new blinds. Similarly, invalid authentications on the server involve querying the database for the client’s information and performing an XOR. The server only updates the blind information upon successful authentication. At this point, we have yet to consider the impact of an honest client failing to authenticate and requiring additional attempts, which would result in his blinding factor becoming out of sync with the server.

The results indicate that an invalid authentication attempt has negligible effect on a server, in fact, it is nearly the same time to perform a template comparison. As a result, the protocol should be resilient to denial of service attacks by brute force attempts to break into the system, which could be further enforced by rate limiting a client’s authentication attempts. By comparing the invalid client and server authentication attempts as well as template comparison, it would appear that the database interaction plays a roll in server performance especially for smaller template sizes. Successful authentication attempts take orders of magnitude longer than a template comparison, but still within the realm of latencies on the Internet and thus should not be easily perceived by individuals.

6.3 Feature Extraction Reliability

To evaluate the ability of the feature extraction libraries, we computed the difference scores for two images extracted from the same individual as well as different individuals and then

processed them using our system. The results, as expected, were identical, though the time to do so was different, as shown earlier in Figure 4. Therefore, in this section, the evaluation primarily focuses on the abilities of the feature extraction libraries to recognize images from the same eye and differentiate those that are not, as shown in Figure 5.

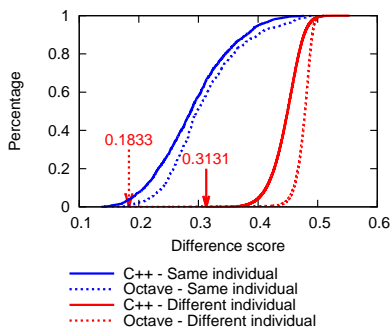


Figure 5: Difference scores using different libraries

While both libraries were able to identify common individuals relatively well (low FRR), we were surprised to see different participants had such low difference scores in both systems, in particular Masek’s. This was a surprising result given Daugman’s earlier analysis [15]. Regardless of the reason for the discrepancy, we are satisfied that our system works equally well as the underlying feature extraction and unprotected matching scheme.

6.4 Feature Extraction Timing

While our system has good response time on the orders of 10s to 100s of milliseconds, depending on the sampling size, we discovered that feature extraction has significant performance overheads as shown in Figure 6. This along with the ability to capture the image play a critical role during the clients authentication process.

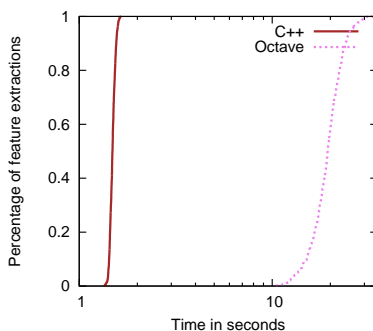


Figure 6: Time for feature extraction

On average, the Project Iris processes images in less 2 seconds; however, the Masek’s library took an order of magnitude longer. Considering these costs, our protocol has nearly

negligible overhead in comparison to the cost of feature extraction alone. However, in comparison to Daugman’s [15] results on a significantly older computer, these libraries seem to perform poorly. Project Iris runs nearly an order of magnitude slower with Masek’s library two orders slow. Using these libraries run time, our protocol overhead is negligible and even using Daugman’s numbers we are still within the same magnitude.

7 Conclusions

Remote biometric authentication faces significant challenges related to the sensitive nature of biometric data. While biometrics are exceptionally suitable for authentication purposes, biometric templates carry sensitive information as user’s identity is embedded into them. Consequently, protection of biometric data is of utmost importance.

The protocol we proposed defines a new approach to remote biometric authentication. It combines biometric- and possession-based authentication in a way that:

- **Protects biometric data.** Our protocol handles biometric data in a way that results in minimal disclosure. This approach guards against attacks exploiting the communication channel and attacks based on compromised proving or verifying party. However, our protocol does not protect the biometric data in case of a simultaneous compromise of both parties.
- **Is theft-resistant.** Peggy uses a token to store her authentication information, including a biometric template secured by a sequence of blinding factors, and the current state of a pseudorandom bit generator. If Peggy loses her token and Mallory uses it or obtains its contents, Peggy’s biometric template cannot be stolen assuming a backtracking resistant generator.
- **Is privacy-preserving.** Our protocol enables biometric data to be used in a way that allows Peggy to create different personas. While each persona derives from the same biometric identity, the blinding factor establishes a unique identity. Peggy can use a different persona for each verifying party to ensure that her actions remain unlinkable across those parties. This approach also allows to easily revoke and replace the blinded reference template if it is ever compromised, for example, if Peggy loses her token. In such cases, Peggy can disregard the compromised template as it does not reveal information about the underlying biometric template and on its own cannot be used for authentication. A replacement token can easily be created by re-enrolling with the verifier.
- **Offers good performance.** Our protocol provides the same recognition performance as straight-forward unprotected comparison of templates, in terms of FAR/FRR, while only adding a little overhead in computation time. In fact, the time and computation delay in feature extraction negates the overhead in our protocol, thus offering a good user experience in contrast to existing protocols.

Acknowledgment

This material is based upon work supported by the Defense Advanced Research Agency (DARPA) and SPAWAR Systems Center Pacific, Contract No. N66001-11-C-4018.

References

- [1] A. Adler. Can images be regenerated from biometric templates? In *Biometrics Consortium Conference*, 2003.
- [2] A. Albrecht. Understanding the issues behind user acceptance. *Biometric Technology Today*, 2001.
- [3] L. Ballard, S. Kamara, F. Monrose, and M. K. Reiter. Towards practical biometric key generation with randomized biometric templates. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, 2008.
- [4] E. Barker and J. Kelsey. Recommendation for random number generation using deterministic random bit generators. Technical report, National Institute of Standards and Technology, 2012.
- [5] O. Bernecker. Biometrics: Security: An end user perspective. *Information Security Technical Report*, 2006.
- [6] M. Bishop. *Introduction to Computer Security*. Addison-Wesley Professional, 2004.
- [7] L. Blum, M. Blum, and M. Shub. A simple unpredictable pseudo random number generator. *SIAM J. Comput.*, 1986.
- [8] M. Boyd, D. Carmaciu, F. Giannaros, T. Payne, and W. Snell. Iris recognition (project iris). <http://projectiris.co.uk/>, March 2010.
- [9] M. Boyd, D. Carmaciu, F. Giannaros, T. Payne, W. Snell, and D. Gillies. Iris recognition. Technical report, Department of Computer Science, Imperial College London, 2010.
- [10] X. Boyen. Reusable cryptographic fuzzy extractors. In *ACM CCS 2004*, ACM, 2004.
- [11] C.-C. Chang and I.-C. Lin. Remarks on fingerprint-based remote user authentication scheme using smart cards. *SIGOPS Oper. Syst. Rev.*, 2004.
- [12] C. Chen and R. Veldhuis. Binary biometric representation through pairwise polar quantization. In *Advances in Biometrics*. Springer Verlag, 2009.
- [13] N. Clarke and S. Furnell. Authentication of users on mobile telephones - a survey of attitudes and practices. *Computers and Security*, 2005.
- [14] R. Clarke. Human identification in information systems: Management challenges and public policy issues. *Information Technology & People*, 1994.

- [15] J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 2002.
- [16] K. Delac and M. Grgic. A survey of biometric recognition methods. In *Electronics in Marine, 2004. Proceedings Elmar 2004. 46th International Symposium*, 2004.
- [17] M. Derawi, C. Nickel, P. Bours, and C. Busch. Unobtrusive user-authentication on mobile phones using biometric gait recognition. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*, 2010.
- [18] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 2008.
- [19] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in cryptology-Eurocrypt 2004*, 2004.
- [20] Y. Dodis and A. Smith. Correcting errors without leaking partial information. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, 2005.
- [21] M. El-Abed, R. Giot, B. Hemery, and C. Rosenberger. A study of users' acceptance and satisfaction of biometric systems. In *Security Technology (ICCST), 2010 IEEE International Carnahan Conference on*, 2010.
- [22] O. Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001.
- [23] S. Goldwasser and M. Bellare. Lecture notes in cryptography, 2001.
- [24] C. Hill. Risk of masquerade arising from the storage of biometrics. *Master's thesis, Australian National University*, 2001.
- [25] S. Hong, W. Jeon, S. Kim, D. Won, and C. Park. The vulnerabilities analysis of fuzzy vault using password. In *Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking - Volume 03*, 2008.
- [26] T. Ignatenko and F. Willems. Information leakage in fuzzy commitment schemes. *Information Forensics and Security, IEEE Transactions on*, 2010.
- [27] Institute of Automation, Chinese Academy of Sciences CAISA. Iris image databases. <http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp>, 11 2012.
- [28] A. Jain, A. Ross, and K. Nandakumar. *Introduction to Biometrics*. Springer, 2011.
- [29] A. K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP J. Adv. Signal Process*, 2008.
- [30] A. K. Jain, A. Ross, and S. Pankanti. Biometrics: A tool for information security. *IEEE Transactions On Information Forensics and Security*, 2006.

- [31] S. Q. Jian-Zhu Lu, Shaoyuan Zhang. Enhanced biometrics-based remote user authentication scheme using smart cards. Cryptology ePrint Archive, Report 2011/676, 2011.
- [32] A. Jin, D. Ling, and A. Goh. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern recognition*, 2004.
- [33] A. Juels and M. Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 2006.
- [34] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and Communications Security*, 1999.
- [35] T. Kevenaar, G. Schrijen, M. van der Veen, A. Akkermans, and F. Zuo. Face recognition with renewable and privacy preserving binary templates. In *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*, 2005.
- [36] M. K. Khan and J. Zhang. Improving the security of 'a flexible biometrics remote user authentication scheme'. *Comput. Stand. Interfaces*, 2007.
- [37] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You. An analysis of biohashing and its variants. *Pattern Recogn.*, 2006.
- [38] J. K. Lee, S. R. Ryu, and K. Y. Yoo. Fingerprint-based remote user authentication scheme using smart cards. *Electronics Letters*, 2002.
- [39] Y. Lee and T. Kwon. An improved fingerprint-based remote user authentication scheme using smart cards. In *Proceedings of the 2006 international conference on Computational Science and Its Applications - Volume Part II*, 2006.
- [40] C.-T. Li and M.-S. Hwang. An efficient biometrics-based remote user authentication scheme using smart cards. *J. Netw. Comput. Appl.*, 2010.
- [41] X. Li, J.-W. Niu, J. Ma, W.-D. Wang, and C.-L. Liu. Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. *J. Netw. Comput. Appl.*, 2011.
- [42] C.-H. Lin and Y.-Y. Lai. A flexible biometrics remote user authentication scheme. *Computer Standards and Interfaces*, 2004.
- [43] A. Lumini and L. Nanni. An improved biohashing for human authentication. *Pattern Recogn.*, 2007.
- [44] L. Masek and P. Kovesi. Matlab source code for a biometric identification system based on iris patterns. Technical report, University of Western Australia, 2003.
- [45] L. Millett and S. Holden. Authentication and its privacy effects. *Internet Computing, IEEE*, 2003.
- [46] E. Mordini and S. Massari. Body, biometrics and identity. *Bioethics*, 2008.
- [47] A. Nagar. *Biometric Template Security*. Dissertation, Michigan State University, 2012.

- [48] A. Nagar, K. Nandakumar, and A. Jain. Biometric template transformation: a security analysis. *Proc. SPIE, Electronic Imaging, Media Forensics and Security XII*, 2010.
- [49] K. Nandakumar, A. Nagar, and A. K. Jain. Hardening fingerprint fuzzy vault using password. In *Proceedings of the 2007 international conference on Advances in Biometrics*, 2007.
- [50] S. Natasha. When a palm reader knows more than your life line. *The New York Times*, 2012.
- [51] A. Patric. Usability and acceptability of biometric security systems. In *Proceedings of the Financial Cryptography Conference (FC04)*, 2004.
- [52] S. Prabhakar, S. Pankanti, and A. K. Jain. Biometric recognition: Security and privacy concerns. *IEEE Security and Privacy*, 2003.
- [53] S. Rane, A. Nagar, and A. Vetro. Method and system for binarization of biometric data, Jan. 15 2010. US Patent App. 12/688,089.
- [54] N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Syst. J.*, 2001.
- [55] C. Rathgeb and A. Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011.
- [56] A. Ross, J. Shah, and A. Jain. From template to image: Reconstructing fingerprints from minutiae points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2007.
- [57] W. J. Scheirer and T. E. Boult. Cracking fuzzy vaults and biometric encryption. In *in Proceedings of Biometrics Symposium*, pages 1–6, 2007.
- [58] K. Simoons, P. Tuyls, and B. Preneel. Privacy weaknesses in biometric sketches. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, 2009.
- [59] Smart cards and biometrics. A Smart Card Alliance Physical Access Council White Paper, March 2011. Publication Number: PAC-11002.
- [60] E. Syta, M. J. Fischer, A. Silberschatz, G. G. García, and B. Ford. Strong theft-proof privacy-preserving biometric authentication. Technical Report TR1455, Department of Computer Science, Yale University, May 2012.
- [61] VeriSign. Symantec Corporation. <http://www.verisign.com>, November 2012.
- [62] VeriSign. Symantec Corporation. Verisign ID protection center: Validation & ID protection (VIP), <http://idprotect.verisign.com>, November 2012.
- [63] A. C. Yao. Theory and application of trapdoor functions. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, 1982.