Private Eyes: Secure Remote Biometric Authentication

Ewa Syta[1], Michael J. Fischer[1], David Wolinsky[1], Abraham Silberschatz[1], Gina Gallegos-García[2] and Bryan Ford[1]

[1]*Department of Computer Science, Yale University, New Haven, CT, U.S.A.*

[2]*Department of Computer Science, National Polytechnic Institute of Mexico, Mexico City, Mexico*

Keywords: Authentication, Biometrics, Privacy, Security.

Abstract: We propose an efficient remote biometric authentication protocol that gives strong protection to the user's biometric data in case of two common kinds of security breaches: (1) loss or theft of the user's token (smart card, handheld device, etc.), giving the attacker full access to any secrets embedded within it; (2) total penetration of the server. Only if both client *and* server are simultaneously compromised is the user's biometric data vulnerable to exposure. The protocol works by encrypting the user's biometric template in a way that allows it to be used for authentication without being decrypted by either token or server. Further, the encrypted template never leaves the token, and only the server has the information that would enable it to be decrypted. We have implemented our protocol using two iris recognition libraries and evaluated its performance. The overall efficiency and recognition performance is essentially the same compared to an unprotected biometric system.

SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

A major problem facing the Internet is "the reliance on passwords to authenticate users" (FIDOa, 2014). The drawbacks of passwords have long been known (Kessler, 1996). Weak passwords are easily guessed; strong passwords are difficult for users to remember and to supply when required. In addition, username and password data must be somehow protected when sent over the network and stored on a server since once compromised, they are sufficient to impersonate the legitimate user.

Combining biometrics with cryptographic authentication schemes is an attractive alternative to password authentication (FIDOb, 2014), an approach supported by the FIDO (Fast IDentity Online) Alliance, a non-profit organization formed to promote easier to use and stronger authentication. FIDO works closely with dozens of prominent industry partners (e.g., Bank of America, Google, Visa, RSA) and strives to reflect the current needs and expectations of the authentication process, from clients and services alike. In their approach, a secret key, stored on the user's local device, is used with a challenge-response protocol to authenticate securely to the server. Biometrics are used to prevent the local device from being activated by any but the legitimate user. However, the user's device stores secret information, which becomes problematic when the device is compromised.

Especially sensitive in any biometric scheme is the user's biometric data which, if compromised, can subsequently be used by an attacker to impersonate the individual to whom it belongs. Unlike passwords, biometric data cannot be changed, so once compromised, it becomes useless as an authentication factor.

Many techniques exist for extracting data stored on a device's internal memory, even from so-called "tamper-resistant" devices (Anderson and Kuhn, 1996). One must assume that an attacker who steals or otherwise gains physical possession of the user's device also obtains the entire contents of the device's internal memory, including any secret keys and biometric data stored therein. Therefore, the user's biometric data must not be stored on the user's device in any form that would allow an attacker to reconstruct it. Similar considerations apply to the server, which also must protect the user's biometric data even in the face of a total compromise.

2 OUR CONTRIBUTION

We propose an efficient remote biometric authentication protocol that gives strong protection to the user's biometric data in case of two common kinds of security breaches: a full client compromise or a full server compromise. Our scheme also allows the creation of multiple unlinkable personas in much the same way as with passwords. (See Section 6.3.) Our two-factor protocol can be combined with a broad class

Syta E., J. Fischer M., Wolinsky D., Silberschatz A., Gallegos-Garcia G. and Ford B..
Private Eyes: Secure Remote Biometric Authentication.
DOI: 10.5220/0005539602430250
In *Proceedings of the 12th International Conference on Security and Cryptography* (SECRYPT-2015), pages 243-250
ISBN: 978-989-758-117-5
Copyright © 2015 SCITEPRESS (Science and Technology Publications, Lda.)

of existing biometric authentication schemes to protect the privacy of the user's biometric data and the template derived from it. It works with any biometric scheme where the result of feature extraction can be represented by a binary feature vector, and the matching criterion for two feature vectors is based on their Hamming distance.[1]

The user's device (or *token*) stores only an encrypted form of the reference biometric template, which we call the *blinded template*. This keeps the biometric data safe even if the token is compromised. The encryption is performed by computing the XOR of the biometric template with a random *blinding factor* that is stored only on the server. How this is accomplished is the heart of our protocol and is described in Section 4. Since the blinding factor is random and carries no information about the actual biometric template, the biometric data is safe even if the server is compromised. Only if *both* token *and* server are simultaneously compromised is the user's biometric data vulnerable to exposure.

The rest of the paper is structured as follows. Section 3 summarizes other solutions to biometric authentication. Section 4 presents our protocol, and Section 5 analyzes its security properties. Section 6 discusses deployment issues. Section 7 describes a prototype implementation and a performance evaluation of our protocol. Section 8 concludes.

3 RELATED WORK

Many biometric authentication protocols offer protection of biometric data. Unlike our protocol, however, those protocols frequently protect the biometric data at the cost of a degraded recognition performance, higher complexity, or a lack of mechanisms to create unlinkable personas.

Standard cryptographic solutions for protecting passwords or other secrets, such as encryption or hashing, are difficult to use for protecting biometric templates because even if two templates are generated using two samples of the same biometric characteristic, they are never exactly the same. Homomorphic encryption (Gentry, 2009) and secure two-party computation techniques (Lindell and Pinkas, 2007) offer good security and privacy guarantees but they normally come at a high performance cost.

Biometric cryptosystems (BC), such as fuzzy extractors (Dodis et al., 2008; Dodis et al., 2004), fuzzy vaults (Juels and Sudan, 2006) and fuzzy commitments (Juels and Wattenberg, 1999), use a template as well as helper data to extract a cryptographic key, with the resulting key validated by verifying its correctness. While BC offers additional features such as reliable cryptographic key generation, they come at the cost of performance and complexity. They heavily rely on error correction codes, which limits their recognition performance to the error-correcting capability of the employed code (Jain et al., 2008; Rathgeb and Uhl, 2011). Furthermore, to achieve reusability and unlinkable personas, BC schemes must be strengthened by adding auxiliary information, for example passwords (Ballard et al., 2008; Nandakumar et al., 2007). This adds to their complexity, limits user convenience, and in some cases may still be insufficient (Hong et al., 2008).

Other template protection schemes use a transformation function, either invertible (BioHashing (Jin et al., 2004)) or non-invertible (cancelable biometrics (Ratha et al., 2001)), and apply it to biometric data during the enrollment phase. For the authentication phase, they apply the same transformation and compare 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 their convenience. A compromise of this additional information can yield further vulnerabilities (Kong et al., 2006; Lumini and Nanni, 2007). This is in contrast to our protocol where a compromise of the user's token does not expose her biometric data. In the case of non-invertible transformations, the recognition performance is affected because the matching is applied to degraded transformed templates (Rathgeb and Uhl, 2011). However, unlinkable personas can be achieved (Jain et al., 2008; Nagar, 2012).

4 PROTOCOL DESCRIPTION

In this section, we give a description of our protocol. We refer the reader to our technical report (Syta et al., 2015) for interesting extensions of the protocol.

4.1 Enrollment Phase

The authentication process is performed over a network between an authenticating party (Peggy, the user) and a verifying party (Victor, the authentication server), after the user enrolls into the system.

During the enrollment phase, Peggy and Victor cooperate to create Peggy's credentials and to establish the shared authentication information (the choice of a biometric characteristic, a feature extractor that produces a biometric template, an appropriate match-

[1] The Hamming distance between two bit vectors is the number of indices in which the two vectors differ.

ing metric on templates, and an appropriate pseudorandom number generator G). Our protocol assumes the template is described by a Boolean vector, and the matching metric is a function of the difference between templates. We assume that G is cryptographically secure (Goldreich, 2001) and backtracking resistant (Bellare and Yee, 2003).

Since our protocol is a two-factor scheme, Peggy needs to obtain a token on which to store her blinded biometric template and the state of the pseudorandom number generator, as further discussed in Section 6.2.

Peggy and Victor also need to obtain a shared secret z to be used as the seed for G. 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. This can be done in person for face-to-face enrollment, using cryptographic techniques such as a key agreement protocol (Goldreich, 2001) for remote enrollment, or Victor can send the seed to Peggy over a secure channel.

To continue the enrollment, Peggy and Victor initialize G using seed z. Peggy obtains a biometric sample using a biometric sensor and generates a reference template P_{ref}. She then blinds P_{ref} with the first blinding factor $R_0 = r_0$ generated using G to produce the blinded template $T_0 = P_{\mathrm{ref}} \oplus R_0$. This process binds Peggy's biometric identity to the secret seed z established with Victor. Victor meanwhile uses G to generate the blinding factor R_0, which he stores for future use. Both Peggy and Victor store the next state s_1 of G. Finally, Peggy securely erases her raw biometric data, the unprotected template, the secret z, the first blinding factor r_0, and the first state s_0 of G. Similarly, Victor securely erases the secret z and the first state s_0 of G. See Algorithm 1 for details.

4.2 Authentication Phase

Peggy and Victor run the authentication phase each time Peggy wishes to prove her identity to Victor. At the start of phase k, Peggy's token stores $T_{k-1} = P_{\mathrm{ref}} \oplus R_{k-1}$ and s_k while Victor stores $R_{k-1} = \oplus_{j=0}^{k-1} r_j$ and s_k.

In order to authenticate, Peggy obtains a fresh biometric sample and generates a new template P_k from it. She then calculates an authentication message

$$W_k = P_k \oplus T_{k-1} = (P_k \oplus P_{\mathrm{ref}}) \oplus R_{k-1}.$$

Peggy sends W_i to Victor for verification. Without waiting for a response from Victor, she immediately updates her token. She uses G to compute $r_k = \rho(s_k)$ and $s_{k+1} = \delta(s_k)$. She then computes

$$T_k = T_{k-1} \oplus r_k = P_{\mathrm{ref}} \oplus R_{k-1} \oplus r_k = P_{\mathrm{ref}} \oplus R_k.$$

Finally, she securely replaces T_{k-1} with T_k and s_k with s_{k+1}, and she securely erases W_k and all other temporary data from memory. By updating after each authentication, successful or not, she ensures that the

Algorithm 1: Enrollment Phase.

1. Peggy obtains a token. Peggy and Victor agree on non-secret authentication information: the biometric recognition protocol and the choice of $G = (m, S, \iota, \delta, \rho, n)$, where m defines the length of the seed, S is the finite set of states of the generator, an initial-state function ι which maps a seed z to a state $s_0 \in S$, δ is the next-state function, ρ is the output function, which produces n-bit values, and n is the length of biometric templates.
2. Peggy and Victor securely exchange a random seed $z \in \{0,1\}^m$.
3. Peggy and Victor both initialize their generator G to the initial state $s_0 = \iota(z)$. They use G to generate the first random number $r_0 = \rho(s_0)$ and the next state $s_1 = \delta(s_0)$ of G.
4. Peggy obtains a biometric template P_{ref}, computes the first blinding factor $R_0 = r_0$, and creates a blinded template $T_0 = P_{\mathrm{ref}} \oplus R_0$, where $\oplus$ is the bit-wise exclusive-OR operation. She securely erases z, P_{ref}, R_0, r_0, and s_0. She keeps T_0 and s_1 on her token.
5. Victor computes the first blinding factor $R_0 = r_0$. He securely erases z, r_0, and s_0. He retains R_0 and s_1 in private storage.

To summarize, after the enrollment phase:

- Peggy's token stores $T_0 = P_{\mathrm{ref}} \oplus R_0$ and s_1.
- Victor retains $R_0 = r_0$ and s_1.

Algorithm 2: Authentication Phase.

1. Peggy obtains a biometric sample and generates a fresh biometric template P_k.
2. Peggy calculates $W_k = P_k \oplus T_{k-1}$ and sends W_i to Victor.
3. Peggy uses G to compute $r_k = \rho(s_k)$ and $s_{k+1} = \delta(s_k)$. She then computes $T_k = T_{k-1} \oplus r_k$. She securely replaces T_{k-1} on her token with T_k and s_k with s_{k+1}, and she securely erases P_k, W_k, r_k, and s_k from her token.
4. Victor, upon receiving W_k, computes the difference vector $V_k = W_k \oplus R_k$. He passes V_k to the matching algorithm and accepts Peggy's authentication attempt if the match is sufficiently good.
5. If the authentication attempt succeeds, Victor uses G to compute $r_k = \rho(s_k)$ and $s_{k+1} = \delta(s_k)$. He then computes $R_k = R_{k-1} \oplus r_k$. He securely replaces R_{k-1} with R_k and s_k with s_{k+1} in memory, and he securely erases r_k, and s_k.

To summarize, at the end of authentication phase k:

- Peggy's token stores $T_k = P_{\mathrm{ref}} \oplus R_k$ and s_{k+1}.
- Victor retains $R_k = \oplus_{j=0}^{k} r_j$ and s_{k+1}.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

same blinding factor is never used more than once.[2]

Upon receiving W_k from Peggy, Victor removes the blinding factor R_{k-1} to obtain the difference vector $V_k = P_k \oplus P_{\text{ref}}$. He applies the matching metric of the chosen biometric system to V_k in order to decide whether or not to accept Peggy's authentication attempt. If valid, he updates his stored information. Using G, he computes $r_k = \rho(s_k)$ and $s_{k+1} = \delta(s_k)$. He then computes $R_k = R_{k-1} \oplus r_k$. Finally, he securely replaces R_{k-1} with R_k and s_k with s_{k+1}, and he securely erases all temporary data from memory. See Algorithm 2 for details.

A legitimate authentication attempt might fail for many reasons, for example, because Peggy's message never reaches Victor, or because of poor feature extraction by Peggy, or because of Victor's not storing the updated blinding factor before going offline, or because of intentional malicious authentication attempts by an adversary. In such cases, Peggy advances her generator but Victor does not. This will leave Victor unable to unblind Peggy's future messages and require resynchronizing the generators to continue. Peggy updates her blinded template T_k after each authentication attempt (hence, she will never be behind) and Victor does so only after a successful authentication (hence, he can always catch up). To do so, Victor searches forward in the sequence produced by G for some limited predefined distance looking for a blinding factor $R_{k'}$ that leads to a successful authentication using Peggy's current authentication message W_k. After finding the correct value of T_k, both generators will again be in sync. Such a scheme is called a rolling code and is widely used in keyless entry systems (Waraksa et al., 1995).

5 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 blinding factor. It is needed to unblind the difference vector, but it will also unblind the template stored on the token.

We refer the reader to our technical report (Syta et al., 2015) for a fuller security analysis, including a discussion on a leakage of information from template differences and suggested solutions.

5.1 Assumptions

Peggy and Victor interact over a network, possibly in the presence of a computationally bounded adversary (Mallory, the malicious adversary). In addition to eavesdropping on all communication between Peggy and Victor, we assume that Mallory can talk directly to Victor in an attempt to impersonate Peggy. In addition, Mallory might actively attack either Peggy or Victor but not both.

In an attack on Peggy, Mallory takes possession of Peggy's token and obtains access to all of the data stored on it. Peggy detects the attack since her token is physically gone. Nevertheless, our protocol guarantees that Peggy's biometric data remains secret and Mallory cannot impersonate Peggy to Victor. In an attack on Victor, Mallory compromises Victor and gains access to all of his data. Victor does not necessarily detect the intrusion. He continues processing authentication requests, and Mallory sees everything that happens on the server. In this case, Peggy's biometric data still remains secret, but Mallory can easily impersonate Peggy to Victor. This can be prevented by using digital signatures (Syta et al., 2015).

We assume that all communication occurs over an unsecured channel, so after k authentication attempts, Mallory knows the authentication messages $W_1, \ldots, W_k$, which are blinded differences between pairs of biometric templates. Thus, Mallory has the following information.

$$\begin{aligned}
W_1 &= P_1 \oplus T_0 &&= P_1 \oplus P_{\text{ref}} \oplus r_0 \\
W_2 &= P_2 \oplus T_1 &&= P_2 \oplus P_{\text{ref}} \oplus r_0 \oplus r_1 \\
W_3 &= P_3 \oplus T_2 &&= P_3 \oplus P_{\text{ref}} \oplus r_0 \oplus r_1 \oplus r_2 \\
&\cdots && \cdots \\
W_k &= P_k \oplus T_{k-1} &&= P_k \oplus P_{\text{ref}} \oplus r_0 \oplus \cdots \oplus r_{k-1}
\end{aligned}$$

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. The security of our protocol depends critically on the pseudorandom number generator G, which we assume is cryptographically secure (Goldreich, 2001) (the sequence of outputs are computationally indistinguishable from a similar sequence

[2] This prevents an attacker from obtaining useful information from differencing two authentication messages W_i and W_j. If they both used the same blinding factor T, the blinding factor would cancel out, and an attacker could compute $W_i \oplus W_j = (P_i \oplus T) \oplus (P_j \oplus T) = P_i \oplus P_j$.

of truly random numbers) and backtracking resistant (Bellare and Yee, 2003) (it is not feasible to run G backwards from a given state to recover previously-generated values). We also assume that the secret seed z and the unprotected template P_{ref} are securely erased after enrollment and are not available to Mallory.

5.2 Security of Biometric Templates

Mallory Compromises Victor. We assume that Mallory can compromise Victor at any time and remain undetected. If she compromises him at phase k, she learns the current blinding factor R_k and the next state of the generator s_{k+1}. This enables her to compute the future random numbers $r_{k+1}, r_{k+2}, \dots$ and the future blinding factors $R_{k+1}, R_{k+2}, \dots$. Clearly, she learns the most by compromising Victor at the very beginning, in which case she recovers the exact same information that Victor receives from Peggy. From this, she can learn all of the difference vectors, $V_1, V_2, \dots$. The usefulness of the difference vectors depends on the underlying feature extractor. Ideally, we want a feature extractor that produces templates on repeated scans of the same biometric that lead to small false rejection rates. When the match function is based on the Hamming distance between two templates, small false rejection rates imply that most difference vectors approximate the zero vector. Hence, Mallory only learns vectors in the neighborhood of 0. In any case, we can say that Mallory has no other information about the template since Peggy sends nothing else besides the blinded difference vectors.

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

$$T_k = P \oplus r_0 \oplus r_1 \oplus \cdots \oplus r_{k-1} \oplus r_k$$

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

T_k looks random to Mallory because of the blinding factor r_k, which she does not know. It was securely erased from the token when T_k was updated, and it was never included in any of the messages sent. Additionally, r_k cannot be recovered using the stored state s_{k+1} of G and $r_0, \dots, r_{k-1}$ (assuming they are known) since G is backtracking resistant and cryptographically secure. Thus, Mallory cannot obtain any information about the blinding factor R_k, so neither T_k nor s_{k+1} give Mallory any information about P_{ref}.

We 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 $k-1$ and stage k.

5.3 Impersonation

Mallory Compromises Victor. As before, when Mallory compromises Victor, she gets R_k and s_{k+1}. R_k is the blinding factor needed to unblind the next authentication message. Therefore, Mallory might be able to prepare a fake message W'_k so that verification will succeed from Victor's point of view. See (Syta et al., 2015) for a practical defense against this attack.

Mallory Compromises Peggy. As before, when Mallory compromises Peggy, she gets T_k and s_{k+1}. We argued in Section 5.2 that her compromise of Peggy's token does not give her any information about P_{ref} or R_k. Hence, she gets no information that would allow her to impersonate Peggy.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

6 USAGE CONSIDERATIONS

This section discusses biometrics suitable for our protocol, tokens as well as creating personas.

6.1 Suitable Biometrics

In practice, fingerprints, face geometry, and iris patterns have been popular choices for biometric systems 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 (Jain et al., 2008). 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. Iris-based systems have been widely deployed by many organizations including British Telecom, Panasonic, LG and IBM Schiphol Group (Daugman, 2002).

For these reasons, we chose to use an iris-based template for our implementation, described in Section 7. These templates typically consist of 2048 bits to represent the iris pattern with any bit equally likely to be either 1 or 0. On average half of all the bits will disagree between the templates of two different people. A study (Daugman, 2002) based on 9.1 million comparisons between different pairings of iris images concluded that it is extremely improbable that two different irises might disagree in fewer than a third of all bits, so a difference score less than 0.32 statistically guarantees a positive match.

Our current protocol assumes binary biometric templates, which are suitable for all iris-based templates, however, a binarization technique (Rane et al., 2010) can convert other types of templates into a binary vector and make it usable in our protocol but likely trading off some recognition performance.

6.2 Enrollment Process and 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 (VeriSign, 2012). 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 or Card).

There are two different approaches to utilizing tokens depending on the token's ability to obtain biometric samples. Using a token with a built-in sensor removes 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; they 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 (Jain et al., 2011). A token without a sensor 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 and the channel between the token and sensor is secure. However, such tokens are inexpensive and make it possible to utilize virtually any biometric characteristic. Smart cards are a good choice for such tokens. They have been extensively used for authentication as they offer enough computational power and are relatively cheap, small, and convenient to use (Smart, 2011).

6.3 Personas

Many biometric authentication protocols create user's authentication credentials directly based on a biometric template. As a result, if a user chooses to enroll with multiple verifying parties using the same biometrics, then these verifying parties can identify the user and successfully track his activities performed using the same biometric credentials or credentials based on the same biometric features.

In our protocol, credentials are based on a user's unprotected biometric template but with respect to the blinding factors known to the verifying party. Hence, a user can create multiple, fully independent personas. Each persona is based on the same biometric data but on a different secret shared with a verifying party so a persona represents a 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 transactions performed using those identities. The user must perform the enrollment process once for each persona, choosing a new secret for each. Policies and procedures controlling the enrollment process would determine how many and what types of personas a user may acquire.

7 EVALUATION

We evaluated our prototype implementation to observe the performance characteristics of our protocol in comparison to using unprotected templates.

7.1 Implementation Details

We have implemented our biometric authentication system in C++ using the Qt framework and Crypto++ cryptographic libraries. For feature extraction, we have employed two different iris recognition libraries: Project Iris (Boyd et al., 2010) and Libor Masek's Iris Recognition (Masek and Kovesi, 2003), both of which utilize John Daugman's approach (Daugman, 2002) to produce an iris template. In evaluation figures, we denote the different libraries by their implementation language: Project Iris as C++ and Masek's library as Octave. We used the CASIA Iris Image Database (Institute of Automation, Chinese Academy of Sciences CASIA, 2012) as input into our system.

We use a Diffie-Hellman Key Agreement (Goldreich, 2001) to agree on a common key. We then use the agreed on key to seed a provably secure Blum-Blum-Shub generator (Blum et al., 1986) (PRBG). We used a SQLite database to store enrollment information. We set a minimal difference score of 0.32 to ensure a low probability of a false match (1 in 26 million) (Daugman, 2002). In their evaluations, Masek's scheme uses a modified Hamming distance scheme that depends on a comparison between two unencrypted templates. Our system encrypts the tem-

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

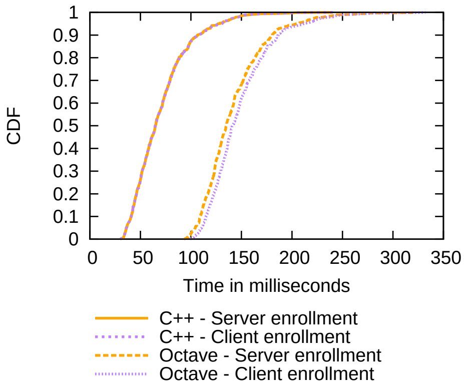

Figure 1: CPU time for enrollment.

plates, therefore, the traditional Hamming distance is more suitable. We analyzed the behavior of these libraries to determine their recognition performance, usefulness and potential overheads in our scheme. We reported the results in (Syta et al., 2015).

The two feature extraction libraries we employ use a technique to further boost the recognition performance. A single template may need to be rotated up to 8° in both directions to achieve better results. In an unprotected biometric system, the server can manipulate the template itself because it gets full access to the client's biometric template. In our system, we preprocess templates to produce these rotations and store the resulting blinded templates on the client's side. We do so because the server never gets access to unprotected biometric data and therefore cannot manipulate the templates itself. Then, during authentication, the client uses all of the saved templates as input to the protocol and forwards the result to the server. Therefore, our protocol preserves the same recognition performance as schemes employing this recognition-enhancing technique.

All CASIA database images have been converted to gray scale images. CASIA database version 1 contains 108 individuals with 7 images each. 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 (Daugman, 2002), Masek extracts a 9600-bit template.

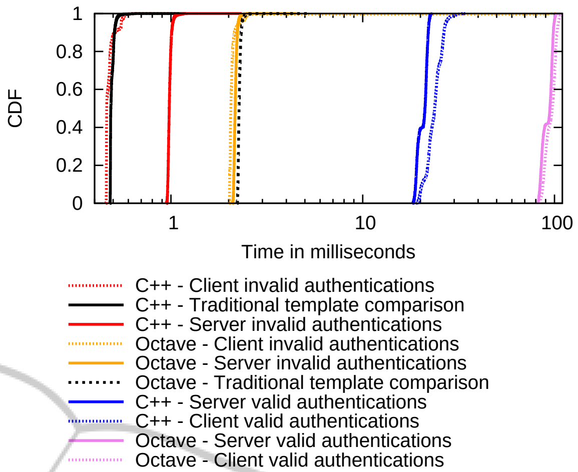

Figure 2: CPU time for authentication.

7.2 System Performance

To evaluate the enrollment phase, we created a client (Peggy) for each image in the CASIA database version 1 and used a single server (Victor). 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. Figure 1 independently plots the time for the client and server components of enrollment.

The clients enrollment time includes both the initial enrollment request and the subsequent processing for a successful enrollment, both represented as a single, summed value. Both client and server enrollment times complete well within 160 ms on average and within 300 ms together in even the worst-case scenario. The client enrollment time is negligibly larger than the server enrollment time. The major factor in performance appears to be the size of the stored template(s) and the need to generate an appropriate amount of random blinding factors. While Octave, Masek's library, uses 17 9600-bit templates with 17 masks resulting in 40.8 KBs of PRNG work, C++, Project Iris, uses only 8.7 KBs.

To evaluate authentication time, we 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 2, 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. Authentications complete on the orders of 10s of milliseconds, which we find reasonable, especially given to the average round trip delay between machines across the Internet.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

8 CONCLUSIONS

Protecting sensitive biometric data is crucially important for remote biometric authentication, because once compromised, the biometric data becomes no longer useful for distinguishing between its legitimate owner and anyone else who possess a copy.

We present a new method for adding strong biometric data protection to a wide class of existing biometric authentication protocols, making them an attractive alternative to password authentication. In particular, biometric data is never stored on the server, only on the user's token, and then only in encrypted form. The token itself requires no secure storage; the biometric data cannot be recovered even if an attacker has full and complete access to everything stored on the token. A lost token also cannot be used to impersonate its owner.

Our method is computationally efficient and has the same recognition performance as the underlying feature extraction scheme. It also allows the creation of independent personas provide enhanced privacy of users' actions across different verifying parties.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

REFERENCES

Anderson, R. and Kuhn, M. (1996). Tamper resistance-a cautionary note. In *Proceedings of the second Usenix workshop on electronic commerce*, volume 2.

Ballard, L., Kamara, S., Monrose, F., and Reiter, M. K. (2008). Towards practical biometric key generation with randomized biometric templates. In *Conference on Computer and Communications Security*.

Bellare, M. and Yee, B. (2003). Forward-security in private-key cryptography. In *Topics in Cryptology - CT RSA*.

Blum, L., Blum, M., and Shub, M. (1986). A simple unpredictable pseudo random number generator. *SIAM J. Comput.*

Boyd, M., Carmaciu, D., Giannaros, F., Payne, T., and Snell, W. (2010). Iris recognition (project iris). http://projectiris.co.uk/.

Daugman, J. (2002). How iris recognition works. *IEEE Trans. on Circuits and Systems for Video Technology*.

Dodis, Y., Ostrovsky, R., Reyzin, L., and Smith, A. (2008). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*

Dodis, Y., Reyzin, L., and Smith, A. (2004). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *EUROCRYPT*.

FIDOa (2014). FIDO alliance: Mission. http://fidoalliance.org/about.

FIDOb (2014). FIDO alliance: Specifications overview. http://fidoalliance.org/specifications.

Gentry, C. (2009). *A fully homomorphic encryption scheme*. PhD thesis, Stanford University.

Goldreich, O. (2001). *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press.

Hong, S., Jeon, W., Kim, S., Won, D., and Park, C. (2008). The vulnerabilities analysis of fuzzy vault using password. In *International Conference on Future Generation Communication and Networking*.

Institute of Automation, Chinese Academy of Sciences CASIA (2012). Iris image databases. http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp.

Jain, A., Ross, A., and Nandakumar, K. (2011). *Introduction to Biometrics*. Springer.

Jain, A. K., Nandakumar, K., and Nagar, A. (2008). Biometric template security. *EURASIP J. Adv. Signal Process*.

Jin, A., Ling, D., and Goh, A. (2004). Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition*.

Juels, A. and Sudan, M. (2006). A fuzzy vault scheme. *Designs, Codes and Cryptography*.

Juels, A. and Wattenberg, M. (1999). A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security*.

Kessler, G. C. (1996). Passwords – strengths and weaknesses. http://www.garykessler.net/library/password.html.

Kong, A., Cheung, K.-H., Zhang, D., Kamel, M., and You, J. (2006). An analysis of biohashing and its variants. *Pattern Recognition*.

Lindell, Y. and Pinkas, B. (2007). An efficient protocol for secure two-party computation in the presence of malicious adversaries. In *Advances in Cryptology-EUROCRYPT 2007*. Springer.

Lumini, A. and Nanni, L. (2007). An improved biohashing for human authentication. *Pattern Recognition*.

Masek, L. and Kovesi, P. (2003). Matlab source code for a biometric identification system based on iris patterns. Technical report, University of Western Australia.

Nagar, A. (2012). *Biometric Template Security*. Dissertation, Michigan State University.

Nandakumar, K., Nagar, A., and Jain, A. K. (2007). Hardening fingerprint fuzzy vault using password. In *International Conference on Advances in Biometrics*.

Rane, S., Nagar, A., and Vetro, A. (2010). Method and system for binarization of biometric data. US Patent App. 12/688,089.

Ratha, N. K., Connell, J. H., and Bolle, R. M. (2001). Enhancing security and privacy in biometrics-based authentication systems. *IBM Syst. J.*

Rathgeb, C. and Uhl, A. (2011). A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*.

Smart (2011). Smart cards and biometrics. A Smart Card Alliance Physical Access Council White Paper. Publication Number: PAC-11002.

Syta, E., Fischer, M. J., , Wolinsky, D., Silberschatz, A., García, G. G., and Ford, B. (2015). Private Eyes: Secure Remote Biometric Authentication (Extended Version). Technical Report TR1510, Yale University.

VeriSign (2012). Symantec Corporation web site. http://www.verisign.com/.

Waraksa, T. J., Michaels, P. A., Slaughter, S. A., Poirier, J. A., and Rea, I. B. (1995). Rolling code for a keyless entry system. US Patent 5,412,379.