Privacy Preserving Biometrics-Based and User Centric Authentication Protocol

Hasini Gunasinghe and Elisa Bertino

Purdue University,
West Lafayette, IN, United States
{huralali,bertino}@purdue.edu

Abstract. We propose a privacy preserving biometrics-based authentication protocol by which users can authenticate to different service providers from their own devices without involving identity providers in the transactions. Authentication is performed through a zero-knowledge proof of knowledge protocol which is based on a cryptographic identity token created using the unique, repeatable and revocable biometric identifier of the user and a secret provided by the user which enables two-factor authentication as well. Our approach for generating biometric identifiers from the user's biometric image is based on the support vector machine classification technique in conjunction with a mechanism for feature extraction from the biometric image. The paper includes experimental results on a dataset of iris images and a security and privacy analysis of the protocol.

Keywords: Privacy, Security, Biometrics, Authentication.

1 Introduction

The safe and secure use of web-based services requires strong authentication mechanisms able to provide assurance about the identity of the service users while, at the same time, protecting the privacy of these users. Biometrics represents a relevant technology for authentication, especially today that mobile devices include biometric sensors allowing users to conveniently provide biometrics for authentication. However, whereas biometrics has many advantages such as uniqueness, its use raises privacy concerns. Upon user enrollment, a biometrics-based authentication system typically stores information extracted from biometrics, referred to as biometric template, into some database. This template is then matched with the template generated when the user needs to authenticate. Security of the biometric template database is thus critical to assure the privacy of biometrics. If the biometric template database is compromised, users may lose their biometric identity permanently due to the lack of revocability. To address such problem, approaches have been proposed to create revocable biometric identifiers (BIDs) [3] and biometric keys (BKs)[1]. With respect to these approaches, it is crucial that BIDs and BKs do not leak sensitive

[1] A BID is a repeatable binary string derived from biometrics whereas a BK is a cryptographic key generated from biometrics.

M.H. Au et al. (Eds.): NSS 2014, LNCS 8792, pp. 389–408, 2014.

information about the original biometric image. At the same time, they should preserve the uniqueness nature of the biometrics, that is, no two individuals should be assigned the same BID or BK. Also, as two readings of the same biometric such as face, iris and fingerprints of the same individual are usually not identical, it is crucial that the generated BID or BK is repeatable, in the sense that each time the BID or BK is generated from a user's biometric template, we should get a BID or BK which is equal to the one generated at enrollment time.

In a typical biometrics-based identity management architecture, a user initially enrolls his/her biometrics at a trusted authority usually referred to as identity provider (IDP). When the user needs to authenticate to a third party service provider, the service provider contacts the IDP for the biometrics-based authentication of the user. The service provider thus relies on the IDP for authenticating the user so that the user does not to have to register his/her biometrics at the third party service provider, thereby better protecting his/her biometrics-based identity. However, this type of architecture, referred to as IDP-centric identity management, raises other types of privacy concerns. Because the IDP is involved in each transaction, it can infer sensitive information such as users' transaction behavior with different service providers.

User-centric identity management architectures address such issues as they do not require the involvement of the IDP in the transactions the users carry out with the service providers. Under such architectures, after the initial enrollment with the IDP, the user can authenticate to the service provider in a secure manner, without involving the IDP. In the VeryIDX system, for example, upon enrollment at the IDP, the user is given some cryptographic authentication token using which the user can authenticate directly to the service provider without having to disclose passwords or other authentication information to the service provider [9]. The design of this kind of identity management solution is however challenging when dealing with biometrics-based authentication through users' mobile phones. The use of mobile phones requires digital identity management solutions able to prevent identity theft in cases in which the phone is stolen, lost or compromised. Today there are commercial products [1] which support biometrics-based user authentication through mobile devices. However, they do not address the above privacy concerns.

The goal of this paper is to propose a privacy preserving and secure protocol for authenticating users from mobile devices to service providers based on their biometrics that addresses the above challenges. The main contributions of this paper can be summarized as follows:

1. We introduce the usage of perceptual hash of biometric images to improve the accuracy of BID generation from a support vector machine (SVM) classifier model.
2. We propose a privacy preserving protocol for user authentication to service providers through mobile devices using BIDs and the techniques to ensure the security of the proposed protocol.
3. An experimental evaluation of the BID generation technique on a dataset of iris biometric images [12], [11].

There is both ongoing and past research that investigates reliable biometrics-based authentication mechanisms. Out of previous research work, the one which is closest to the biometrics-based authentication mechanism that we propose, is by Bhargav-Spantzel et al. [3]. Such previous approach has some major drawbacks which make it not suitable for authentication in mobile devices. It assumes a centralized IDP which is involved in both the enrollment and verification of the biometric identity of the users. In particular, such previous protocol requires that the user connects at the IDP when having to authenticate, as the IDP stores the SVM classifier which is required to generate the BID[2]. Such previous approach is thus an IDP-centric identity management approach, and therefore it makes it possible for the IDP to learn information about transactions executed by the users. Even if such previous protocol [3] were extended for use in mobile devices by storing the SVM classifier on the mobile device to allow the user to authenticate without connecting to the IDP, leakage of the internals of the SVM classifier by a malicious user of the system or an attacker who steals the user's mobile phone, could compromise the BIDs of all the other users of the system, as the SVM classifier is the same for all the users of the system. By contrast, in our approach, each user has a different customized SVM classifier. The image hashing mechanism used in [3] also does not perform with good accuracy when used to classify biometric images that are not present when training the SVM classifier, which is the main reason why we use the perceptual hashing mechanism which improves the accuracy in classifying newly captured biometric images. Furthermore, based on our experiments, the approach for generating BIDs proposed by Bharghava-Spantzel et al. [3] does not assure repeatability in practice (although it seems theoretically achievable), because it depends on the repeatability of the probability estimates provided as classification output by the SVM classifier with respect to multiple classes. Therefore, we make use of a single class prediction output provided by the SVM classifier when generating the BID. We also define an extension to the core protocol which supports the application of error correction codes on the feature vector extracted from the biometric image, in order to improve the repeatability of the biometric identity without compromising the uniqueness.

The rest of the paper is organized as follows. Section 2 introduces the main concepts used in our solutions. Section 3 explains our approach. Details and results of the experiments are presented in Section 4. We analyze the security of our approach in Section 5. We discuss related work in Section 6 and outline conclusions and future work in Section 7.

2 Background

In what follows we introduce the main concepts and techniques which are used as building blocks in our solution.

[2] We describe the SVM classifier in details in sections 2.2 and 3.1.

2.1 Perceptual Hash

A perceptual hash (P-Hash) is a signature of an underlying media source file's perceptual content [7]. While perceptual hashing applies to all multimedia types such as audio, video and images, we will only consider images as the media type of interest in our discussion. Perceptual hashes are intended to establish the perceptual equality in different images that look similar. In order to serve this purpose, perceptual image hashing functions extract features from the image and calculate a hash value based on these features. In general, there are four properties that need to be satisfied by a P-Hash function [14] which can be summarized as: equal distribution (unpredictability) of hash values, pairwise independence for perceptually different images, invariance for perceptually similar images, and distinction of perceptually different images. We leverage those properties to identify the similarity in the biometric images of the same user, captured at different times.

Many different perceptual image hashing functions have been proposed in the literature and one should select the relevant P-Hash function based on the application scenario. The Discrete Cosine Transformation (DCT) based hash is the mostly used one with several implementations available such as the publicly available pHash library by Zauner [14], [7]. The DCT is a linear and invertible function. The most common variation of the DCT is the type-II DCT which is the one used in our solution, which we simply refer to as DCT.

Algorithm 1 lists the P-Hash algorithm based on the DCT. The conversion to greyscale using luminance (line 2) is common to all P-Hash functions since the essential information resides in the luminance component of the image [14]. Resizing (line 4) is done to simplify computing the DCT of the image. 64 low frequency DCT coefficients are extracted for computing the hash, which constitutes a square matrix of size 8×8 (line 8). The low frequency coefficients are considered to be perceptually most significant because most of the image information tends to be concentrated in a few low frequency components of the DCT. The elements of one dimensional array created from the DCT coefficient matrix are normalized based on the median, to compute the final hash as listed in lines 17-19. Accordingly, the final hash output of the P-Hash function does not reveal the actual low frequencies; it just represents a very rough relative scale of the frequencies to the median.

2.2 Support Vector Machine

Given a set of training examples composed of pairs of the form $\{x_i, y_j\}$, the SVM classification technique finds a function $f(x)$ that maps each attribute vector x_i to its associated class y_j, $j = 1, 2, 3 \ldots n$ where n is the total number of classes represented by training data. The SVM is a discriminative classifier defined by separating hyper planes, that is, given the labeled training data, the algorithm outputs optimal hyper planes (i.e. maximum separating hyper planes) which categorize new samples which are also known as testing data. The SVM algorithm includes a kernel function which maps training data to improve its

Algorithm 1. Biometric Image Hashing Algorithm

1: $I \leftarrow$ input image.
// Preprocessing: (lines 2-4)
2: $I_1 \leftarrow$ convert I to greyscale using its luminance.
// apply mean filter on I_1:
3: $I_2 \leftarrow$ convolution of I_1 with a 7×7 kernel.
4: $I_3 \leftarrow$ resize I_2 to 32×32.
// generate DCT matrix: C of size 32×32
5: $C[n,m]=\sqrt{\frac{2}{N}}.\cos(\frac{(2m+1).n\pi}{2N})$, where $m,n = 0,\ldots,N-1$ & $N = 32$
6: $C' \leftarrow$ transpose of C.
// generate DCT coefficient matrix of I_3: (lines 7-8)
7: $I_4 \leftarrow C.\ I_3.\ C'$
8: $V \leftarrow$ extract 8×8 DCT coefficient matrix from I_4:
$v(i,j) = I_4(i,j)$ where $i = 1,\ldots,8$; $j = 1,\ldots,8$
// Rows of matrix V are concatenated to form one dimensional array Z
9: **for** each i where $0 < i < 9$ **do**
10: **for** each j where $0 < j < 9$ **do**
11: $z[8 * (i-1) + j] = v(i,j)$
12: $j++$
13: **end for**
14: $i++$
15: **end for**
16: $m \leftarrow$ median of all $z[k]$ where $0 < k < 65$.
// calculate P-Hash H
17: **for** each k where $0 < k < 64$ **do**
18:

$$H(k) = \begin{cases} 0 & \text{if } z[k] < m \\ 1 & \text{if } z[k] \geq m \end{cases}$$

19: **end for**
20: **return** hash H.

resemblance to a linearly separable set of data. This increases the dimensionality of data. We incorporate the Radial Basis Function (RBF) kernel with optimal values for C and γ parameters [3] selected based on k-fold cross validation accuracy in grid search, as we discuss in Section 4. We use the prediction output of the trained SVM classifier to generate the BID of the user, during enrollment as well as during authentication.

2.3 Pedersen Commitment

The Pedersen commitment [10] is a secure commitment scheme whose security is based on the hardness of solving discrete logarithms. The operation of this

[3] C trades off misclassification of training samples against simplicity of the decision surface in the SVM. A low value of C makes the decision surface smooth, while a high value of C aims at classifying all training examples correctly. γ is a kernel specific parameter which determines the RBF width.

commitment scheme, which involves a committer and a verifier, can be described by following three steps.

Setup: Let p and q be large primes, such that q divides $p-1$. Typically p is of 1024 bits and q is of 160 bits. G_q is a unique, order-q sub group of Z_p^* -which is the multiplicative group of order p. A trusted party chooses g -a generator of G_q and h ($= g^a \text{mod p}$ where 'a' is secret) -an element of G_q such that it is computationally hard to find $log_g h$, and publishes (p, q, g, h).

Commit: The committer creates the commitment of $x \in Z_q$ by choosing $r \in Z_q$ at random and computing: $C(x, r) = g^x h^r \text{ mod p} \in G_q$.

Open: To open the commitment, the committer reveals x and r and the verifier checks if $C = g^x h^r$ to verify the authenticity of the commitment.

The Pedersen commitment has two properties: it is unconditionally hiding - every possible value of x is equally likely to be committed in C, and it is computationally binding - one cannot open the commitment with any $x' \neq x$, unless he can compute $log_g h$. We leverage these properties to hide the BID of the user in an identity token.

2.4 Zero Knowledge Proof of Knowledge Protocol

A zero knowledge proof of knowledge (ZKPK) protocol is a protocol by which the owner of a secret can prove to a verifier his/her knowledge about the secret without making it any easier for the verifier to obtain the actual secret. In our work, we use the protocol listed in Protocol 1 to prove the knowledge of the two secret values x and r hidden in the Pedersen commitment, without revealing the actual values of x and r to the verifier. This protocol has three properties: completeness - if the committer and verifier are honest, the protocol succeeds with overwhelming probability; soundness - the protocol does not allow the committer to prove a false statement; and zero knowledge - the proof does not leak any information about the secrets. We leverage these properties in our solution for the user to prove that he/she is the actual owner of the BID and the secret hidden in the identity token.
Let U denote the committer and V denote the verifier.

Protocol 1. Zero Knowledge Proof of Knowledge

1: $U \rightarrow V$: U randomly picks $y, s \in Z_q$ and sends $d = g^y h^s \in G_q$ to V.
2: $V \rightarrow U$: V sends random challenge $e \in Z_q$ to U.
3: $U \rightarrow V$: U sends $u = y + ex$ and $v = s + er$ to V.
4: V: accepts if $g^u h^v = dC^e$.

2.5 Hadamard Error Correction Code

There are two main types of error correction code (ECC): convolutional codes - which are processed on a bit-by-bit basis, and block codes - which are processed on a block-by-block basis. The Hadamard ECC (HECC) [4] falls into the second category, in which a large stream of data is broken into fixed size blocks called 'message' and each message is encoded to a codeword in the HECC, before being sent

over a noisy channel. At the receiver end, the received n-bit long block is decoded in order to recover the original message. Codewords in the HECC are derived from a Hadamard matrix which is a square orthogonal matrix consisting of elements 1, -1. The Hadamard codewords matrix is obtained from the Hadamard matrix and the Hadamard negative matrix by replacing -1 with 0:

$HC_{nx2n} = \begin{bmatrix} H_{nxn} \\ -H_{nxn} \end{bmatrix}$

In general, an instance of the HECC, represented as $\{n, k, d\}_q$, encodes messages of size k with codewords of size n which are generated from the Hadamard matrix of size n where n = 2^{k-1}. It applies on an alphabet of size q and can correct errors up to $\lfloor (d-1)/2 \rfloor$ number of bit errors where d (= 2^{k-2}) represents the distance or the number of bit positions in which any two distinct codewords differ. Therefore, the size of the codeword matrix, the error correction capability and also the overhead due to encoding (which is denoted by $1 - (k/n)$) differ based on the parameters of the particular instance of the block code. We make use of this technique to correct errors occurring in the features extracted from different biometric images of the same user. To enhance security, we incorporate a pair of encoding and decoding algorithms that involve a secret key, as introduced by Kande et al. [6], instead of directly encoding the feature vector obtained at the enrollment time, as described in Section 3.

2.6 Key Derivation from a Password

In our proposed scheme, three secrets ($S_i : i \in \{1, 2, 3\}$) are used in different steps of the protocol: S_1 - is combined with the class label output by the SVM to generate the BID of the user (size: 128 bits); S_2 - is used as the secret r in the Pedersen Commitment (size: 160 bits); S_3 - is the key used in error correction encoding/decoding algorithm (size: 104 bits). In order to address usability concerns, such as the user having to enter three passwords during the execution of the protocol, and security concerns, such as having to store the secrets somewhere and the secrets not being uniformly randomly distributed in the key space, we make use of the password based key derivation function 2 (PBKD2) for deriving the three secrets from a single password provided by the user, which involves PKCS#5 as the pseudo random function (PRF) and a salt value to make dictionary attacks harder. The key derivation algorithm is thus as follows. We first generate a secret S as:
S = PBKDF2 (PKCS#5, Password, Salt, derived key length (=392 bits)).
We then partition S into three parts of which the first is S_1 with 128 bits, the second is S_2 with 160 bits, and the third is S_3 is 104 bits.

3 Our Approach

In what follows we first present an overview of our methodology for training and customizing the SVM classifier, generating unique and repeatable BIDs from biometric images, and using the BIDs for generating cryptographic identity tokens in order to perform multi-factor authentication with ZKPK protocol. Then

we present our proposed protocol for privacy preserving biometrics-based authentication of users in a user-centric identity management system.

3.1 Methodology

As biometric images of the same individual captured at different times are not identical, we rely on the output of SVM classification to generate a repeatable BID. We selected the P-Hash as the feature extraction mechanism after comparing it with the singular vector decomposition (SVD) based image hashing mechanism used in [3] in order to improve the accuracy of the SVM classification (see Section 4 for experimental results).

3.1.1 Training and Customizing the SVM Classifier

The IDP trains a SVM classifier using the P-hash vectors computed from biometric images of a population of different individuals including several biometric images from each individual. We refer to such classifier as the base SVM model. How the set of biometric images is selected to train the base SVM model is based on the organizational context of the IDP. For example, in the case of an organization under one administrative domain, the IDP can collect biometric images of the existing employees and train the base SVM model. In contrast, in the case of a public authority which is going to manage users' biometrics-based identity enrollment, the IDP can train the initial base SVM model using the biometric images obtained from a sample population of citizens or even from publicly available biometric datasets. A customized SVM model is then created for each user in the system, from the trained base SVM model during the enrollment of the user in the system. The customization is done by replacing each class label in the original model with a new class label, which is a randomly generated integer. The customized SVM model generated for a particular user is different from the customized SVM models of all other users in the system since each customized model uses different sets of random integers as class labels. When a user enrolls his/her biometric at the IDP, the user is given the customized SVM model to be stored in the user's device along with the authentication client application, as described in Section 3.2. The SVM model is customized for each user in order to prevent an attacker, who by some means gains access to the trained SVM model in a user's device, from learning all possible class labels that are used to construct the BIDs of all the users in the system. The trained and customized SVM model which is saved as a structured file can later be loaded in order to obtain the classification output when the user performs biometrics-based authentication. The authentication process uses the customized SVM model and it is thus independent from the base SVM model (see Figure 2).

It is important to note that both types of IDP mentioned above can periodically build a new base SVM model with the biometric images of the new users added to the system, thus introducing more classes to the SVM model. Because the users who already enrolled have a customized SVM model obtained at the enrollment time, they can continue using such customized model independent

from the updates made to the base SVM model at the IDP. Experimental details concerning the training of the SVM model using the biometric hash vectors obtained from the biometric images are discussed in Section 4.

3.1.2 Generating the BID

The BID generation takes place in two main phases of the protocol that we propose, namely enrollment and authentication. As shown in Figure 1, there are two main phases in the BID generation process. During phase 1, the biometric image is captured, preprocessed, and the P-Hash of the image is computed by extracting the features from the image. The output of this phase is the biometric hash vector which is a binary array of 64 bits. This hash vector is then given as input to the trained and customized SVM model in order to obtain the classification output which is the class label that represents the class that the hash vector belongs to. This class label concatenated with the secret S_1, which is derived from user's password, is the BID of the user. The class label is an integer represented by 32-bits and S_1 is 128-bits as mentioned in Section 2.6. Therefore, the generated BID (which is 160-bits) is an element of Z_q defined in Section 2.3. This BID is then used to create the cryptographic identity token as described in the following section.

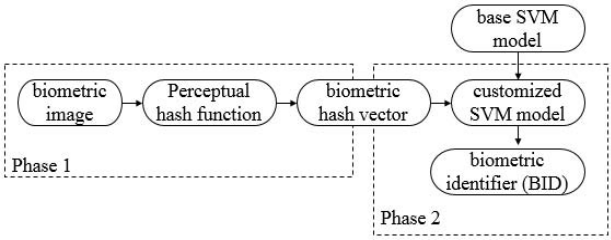

Fig. 1. BID generation during enrollment

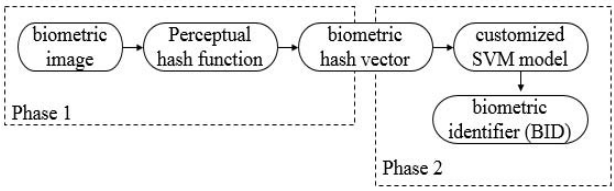

Fig. 2. BID generation during authentication

3.1.3 Creating Cryptographic Identity Token from the BID

The cryptographic identity commitment to be included in the identity token (IDT) is computed using the Pedersen commitment scheme with the BID and a random secret as input as described in Section 2.3. This random secret is the S_2 derived from user's password as described in Section 2.6, which is 160-bits

Algorithm 2. Creating Biometrics-based Identity Token

1: $x \leftarrow$ BID | S_1 (| represents concatenation)
2: $r \leftarrow S_2$
3: $c \leftarrow g^x h^r$
4: $IDT \leftarrow$ Create identity token with following fields:
Commitment (c):
Expiration Timestamp:
From:
To:
5: **return** IDT digitally signed by the IDP.

long and hence is an element of Z_q. The public parameters (p, q, g, h) required for the commitment scheme are initialized and published by the IDP. As listed in Algorithm 2, in addition to the 'Commitment' and 'Expiration timestamp' fields, we also explicitly add 'From' and 'To' fields in the IDT in order to avoid certain types of attacks on the ZKPK identity verification protocols referred to as Mafia fraud attacks which is further discussed in Section 5. In the 'From' field, the user can request the IDP to include a pseudonym of the user in order to prevent identity linkability. In the 'To' field, user can request the IDP to include a commitment on the name of the service provider to whom the IDT will be provided, in order to prevent a malicious service provider from impersonating the user to another service provider.

3.2 Biometrics-Identity Management Protocol

There are three main parties involved in our protocol: users, the IDP, and the service provider (also called relying party). The protocol consists of two main phases: Enrollment - by which the user obtains his/her biometrics-based cryptographic identity token digitally signed by the IDP; Authentication - by which the user proves his/her biometrics-based identity at the third party service providers. While the enrollment phase involves all three parties mentioned above, the authentication phase involves only the user and the service provider.

Let *IDP* denote the identity provider, *U* denote the user, *SP* denote the service provider and *IDT* denote the cryptographic identity token based on the user's biometric. Please note that when we refer each of these entities, both human and software aspects related to them are implied. For example, when we refer to *U* in the protocol, we refer to the actions taken by both the human user and the software installed in user's device such as the software provided by the *IDP* as well as by the *SP*. Protocol 2a lists the steps executed during the enrollment phase. The secure channel mentioned in the step 1 of protocol 2a refers to a channel with message level security. In order to prevent a malicious user from providing a fake biometric image instead of the actual biometric image in step 1, the enrollment of the user's biometric should be executed at the IDP following the necessary legal processes such as requiring the user to visit the authority in person and proving his/her identity using a legal identifier that he/she possesses, such as SSN or passport, which is outside the technical scope of our

Protocol 2a. Enrollment Phase

1: $U \rightarrow IDP$:

- Provides the biometric image and password over a secure channel.
- Sends details for 'From' and 'To' fields of the IDT (optional).

2: IDP:

- Customizes the base SVM for the user.
- Computes the P-Hash of the user's biometric image.
- Obtains SVM prediction output for the computed P-Hash vector.
- Derives three secrets from user's password by choosing a salt value (t). (see section 2.6)
- Creates the BID. (see section 3.1.2)
- Creates the cryptographic identity commitment. (see sections 2.3 and 3.1.3)
- Creates the IDT (see section 3.1.3).

3: $IDP \rightarrow U$: provides the authentication client application along with the trained and customized SVM model, the IDT, and the salt value (t).

current work. In order to enable a user to perform biometrics-based authentication from his/her mobile device(s) without involvement of the IDP, the BID generation software which is a part of the authentication client application listed in the step 3 of protocol 2a, the trained and customized SVM model, and the salt value (t) used in deriving the secrets based on the user's password are provided to the user by the IDP at the end of the enrollment phase. Those artifacts can be saved and installed securely on the user's mobile device(s) through the provisioning mechanisms of the Trusted Execution Environment (TEE) enabled in the modern mobile devices. Today there are commercial mobile devices that facilitate application developers in making use of hardware based TEE such as the onboard credential (ObC) architecture [8]. We assume the use of such techniques to securely store these sensitive meta-data related to the proposed protocol in order to prevent identity theft in cases in which the user's device is stolen by a computationally powerful attacker as discussed under different adversary models in Section 5. We utilize the standard PKI based digital signature for digitally signing the IDT in our protocols. The user can provide the IDT obtained at the end of the execution of protocol 2a, when signing up with the service provider, by executing the Protocol 2b.

Protocol 2b. User signs up at SP

1: $U \rightarrow SP$: U signs up at SP and provides the IDT to SP as a strong identifier.
2: SP: verifies the signature and the expiration date on the IDT.
3: SP: stores the IDT linked to user's identity.

As shown in Protocol 2b, the service provider verifies the digital signature of the IDP and the expiration time stamp on the IDT and stores the IDT linked

to the user's identity to be used in multi-factor authentication during subsequent authentication attempts. Note however that it also possible for the service provider not to store the IDT as the authentication client application can send the token to the service provider whenever the user needs to authenticate. Web services APIs exposed by the service providers are usually accessed from the user's device(s) through web based or native client applications provided by the service provider. However, making each service provider's client able to handle biometrics-based authentication of the user is neither secure nor efficient. Therefore, service providers' clients can delegate the authentication step to the authentication client application installed in the user's device which is provided by the IDP during the enrollment phase.

During authentication, as shown in Protocol 2c, the cryptographic identity commitment is created following the same steps as in Protocol 2a after which the authentication client application sends it to the corresponding service provider along with the IDT that was obtained during enrollment. Upon receiving the authentication request, the service provider verifies the signature of the IDP on the IDT. If the signature verification is successful, the service provider and the user execute the ZKPK protocol, described in Section 2.4, in order for the user to prove his/her ownership of the biometrics and knowledge of the secrets involved in the IDT. If the ZKPK protocol succeeds, the service provider creates a session for the user and provides the session details to the authentication client application. After the authentication application hands over the control back to the service provider's client along with the session information, the service provider's client can perform the transactions requested by the user with the corresponding service provider.

Protocol 2c. Authentication Phase

1: U:

- Captures the biometric image and inputs the password.
- Computes the P-Hash of the biometric image.
- Obtains the SVM prediction output for the computed P-Hash.
- Derives three secrets from the password and the stored salt value (t). (see section 2.6)
- Creates the BID. (see section 3.1.2)
- Creates the cryptographic identity commitment (C'). (see sections 2.3 and 3.1.3)

2: $U \rightarrow SP$: Sends the IDT and the commitment (C') along with the authentication request.
3: SP: verifies signature of the IDP on the IDT.
4: **if** signature verification is successful **then**
5: $U \leftrightarrow SP$: executes ZKPK protocol for two-factor authentication.
6: SP: accepts U as authenticated if ZKPK protocol succeeds.
7: **end if**

3.3 Extended Protocol with Support for Error Correction Code

We extend the core protocol presented in the previous section to improve the repeatability of the biometric identity of the user by performing error correction

on the biometric feature vector obtained from the feature extraction mechanism, which is the P-Hash. The reason why the commitment created during authentication may not match the commitment included in the IDT is that the SVM classification output obtained during authentication is not the same as the output obtained during enrollment, assuming that the user has provided the correct password. In such cases, the authentication client application installed in the user's device performs error correction on the biometric feature vector, using some meta-data provided by the IDP during enrollment. We use HECC [4] for error correction encoding and decoding. However, for security, we do not directly encode the biometric hash vector obtained during the enrollment and store it in the user's device. The reason is that if an attacker gets hold of it, the attacker can easily decode it to obtain the actual hash vector which is highly sensitive data. Therefore we adopt the secure error correction encoding and decoding mechanism introduced by Kande et al. [6] in which a secret key is encoded using HECC and then XORED with the biometric hash vector in order to create the error correction meta-data to be used in the decoding phase during authentication.

3.3.1 Error Correction Encoding Algorithm

The equation (1) given below is used to encode the error correction meta-data during enrollment time, which is to be used during authentication time. X is the biometric feature vector obtained during enrollment time, 'key' is the secret, Z is the error correction meta data provided to the user and HE stands for Hadamard Encoding.

$$Z = HE(key) \oplus X \tag{1}$$

In the use of this algorithm in [6], X is a biometric feature vector of size 1188 bits (obtained using a feature extraction mechanism called Iris Code), and they have used HECC instance of $(32, 6, 16)_2$ which results in a key length of 222 bits ($= 6 \times 37$: message size $\times$ number of blocks), which is large enough to resist brute force attacks. In our case, since the feature vector, which is the P-Hash, is of only 64 bits length, and since we utilize the HECC instance of $(16, 5, 8)_2$ based on our experimental evaluation (see Section 4), the required key size happened to be only 20 bits (5×4) which is not long enough to be resistant against brute force attacks. In order to make the secret key large enough, we repeatedly concatenate the biometric hash vector to itself 5 times. Therefore, in the use of equation (1) in our extended protocol, $X = X_1 \mid X_2 \mid X_3 \mid X_4 \mid X_5$ where X_i is x_{i1},x_{i2},...,x_{i64} which is the 64 bits long biometric hash vector computed using the P-Hash and 'key' is S_3 (the first 100 bits of S_3 derived from the user's password). Z is stored in the TEE of user's mobile device which is used to store the SVM classifier as well. X and S_3 protect each other as one time pads and are secure as long as both Z and the password are not stolen.

3.3.2 Error Correction Decoding Algorithm

During authentication, the error correction decoding algorithm performs block-wise decoding (block size is 16 bits) as listed in Algorithm 3. Like in the encoding

algorithm, Y is the biometric feature vector extracted from the biometric image of the user, repeatedly concatenated with itself for 5 times, and similarly Y' is the error corrected feature vector concatenated to itself 5 times and HD stands for Hadamard Decoding. After we obtain Y', we extract the first 64 bits of it

Algorithm 3. Error Correction Decoding Algorithm

```
In blocks of 16 bits:
  Do: S = Y ⊕ Z = Y ⊕ [HE(S3) ⊕ X]
if S3 = HD(S): then
    Y' = HE(S3)⊕Z (=X)
else
    Y' = Y
end if
```

which is the error corrected feature vector of the user's biometric image captured during authentication. This error corrected feature vector is then given as input to the SVM classifier to obtain the classification output which is used to build the BID of the user. This error correcting decoding process needs to be performed only if the commitment created during the authentication does not match the commitment created during enrollment which is included in the IDT.

4 Experiments

We have conducted experiments to evaluate the proposed protocol for biometrics-based authentication, using both the SVD based hash [3] and the P-Hash as the feature extraction mechanisms and using iris images as biometrics. The SVM model was trained by finding the optimal parameters for the SVM algorithm as described in Section 4.1.1. We measured the accuracy in classifying the biometric images during authentication (which were not present during training of the SVM classifier), along with other measurements. We also present the results of the experiments carried out with the extended protocol using error correction code.

4.1 Data Set and Experimental Setup

The experiments were conducted on a laptop machine with the Ubuntu 13.4 OS, Intel Core i7-3537U CPU, and 5 GB memory. The experiments were conducted in two rounds with iris images from the two UBIRIS V.1 [12] and UBIRIS V.2 [11] databases.

Iris is considered as the most accurate biometric trait in the context of biometrics, and is being used in different domains such as airport check-in and refugee control [2]. Since the accuracy of current systems depends on the accuracy of the iris image capturing process which requires the cooperation from the user such as requiring the user to stand close to the camera and look for a period of

about three seconds, the UBIRIS databases have been built with the purpose of analyzing the methodologies to recognize users with minimum cooperation. The iris images in UBIRIS V.1 were captured in environments with some minimal constraints, whereas the UBIRIS V.2 images were captured in non-constrained environments such as at-a-distance and on-the-move, with more realistic noise factors [11]. However, the quality of the images in UBIRIS V.2 seems better than those in UBIRIS V.1.

4.1.1 Experiments on the Core Protocol

500 iris images were selected from each of the above databases including 5 images from each of the 100 individuals. The training data set was constructed with the first four images from each individual and the testing data set was constructed with the fifth image from each individual. Feature extraction and computation of the P-Hash of the iris images were executed using the DCT based P-Hash function implemented in the publicly available pHash library [14]. The SVM model was built using the LIBSVM library [5]. The SVM type used was C-SVC which is intended for multi-class classification and the type of kernel function used was Radial Basis Function. The P-Hash output was formatted and prepared as input for the SVM classifier such that each bit is marked as a feature in an input vector of 64 elements. Given the training data, a range of values for the optimal pair of C and γ parameters of the SVM model was selected by evaluating 10-fold cross validation accuracy (CV accuracy) of each combination of values within the range of: $C - \{2^{-10}, 2^{15}\}$ and $\gamma - \{2^{-10}, 2^{15}\}$ using grid search. Finally the SVM classifier was trained using the C and γ values selected by evaluating the aforementioned range of C and γ values for the best CV accuracy in grid search, as listed in Table 1.

4.1.2 Experiments on the Extended Protocol

We tried to further improve the classification accuracy by correcting errors occurring in the P-Hash vector of the biometric image captured during authentication time when compared to the P-Hash vector of the biometric image of the same user captured at enrollment time, using the encoding and decoding algorithms described in Section 3.3, which are based on Hadamard Error Correction Code. We applied this to the data set on which SVM showed better classification accuracy which is UBIRIS V.2. Experiments were carried out with Hadamard Codes of increasing length to find the optimal length in terms of improved accuracy and low overhead.

4.2 Results

The results summarized in the Table 1 demonstrate the accuracy of classifying a biometric image captured at authentication time which was thus not known to the classifier at the training time, false rejection rate (FRR), and false acceptance rate (FAR). With P-Hash as the feature extraction mechanism, the accuracy of classifying previously unknown biometric samples is 88% for the UBIRIS.v2

dataset and 79% for the UBIRIS.v1 dataset. In contrast, the SVD based hashing mechanism results in accuracy below 62% for both data sets. It is important to emphasize that the proposed protocol is flexible enough to adapt to any other feature extraction mechanism as well, if it contributes to even better accuracy in SVM classification of the biometric images of the users.

Another criterion to evaluate the classification performance is on the basis of falsely classified images. The FAR and the FRR are two commonly used metrics to quantify the probability of falsely classified images. FRR is the probability that perceptually similar images are identified as different while FAR is the probability that perceptually different images are identified as similar. The latter is the crucial factor in a biometrics-based authentication system. According to Table 1, both FRR and FAR for the P-Hash are comparatively low for both datasets. Table 2 summarizes the SVM classification performance when the error correction was applied to P-Hash vectors of the biometric images of UBIRIS V.2 dataset, by varying the length of the Hadamard Codes. An Hadamard Code length of 0 means that no error correction was applied. As we can observe, with the increasing length of Hadamard Codes, the accuracy of the SVM classification increases due to the increase of error correction capability. However, the overhead resulting from the error correction also increases because the ratio of message size(k) to the Hadamard Code size(n) decreases. Therefore, considering the trade off between improvement of accuracy and the overhead, using 16-bit Hadamard codes seems the best trade-off in order to correct errors occurring in the P-Hash vector of the user's biometric image at authentication time, if the commitment created during authentication time does not match the commitment in the IDT

Table 1. Summary of the Experimental Results w/o Error Correction

Data Set	Measurement	P-Hash based SVM	SVD-Hash based SVM
(a)	Best CV accuracy in grid search	85.75%	52.5%
	Optimal values for C & γ	C=16, γ=0.0078125	C=8, γ=1024
	Accuracy of classification during authentication	88%	61%
	FRR	0.12	0.39
	FAR	0.0012	0.0039
(b)	Best CV accuracy in grid search	75.75%	43.5%
	Optimal values for C & γ	C=128, γ=0.00097656	C=32, γ=512
	Accuracy of classification during authentication	79%	49%
	FRR	0.21	0.51
	FAR	0.0021	0.0051

(a) UBIRIS V.2 (b) UBIRIS V.1

Table 2. Summary of the Experimental Results with Error Correction

Hadamard Code Length	0	8	16	32	64
Accuracy of classification during authentication	88	89%	90%	90%	91%
Overhead (1-(k/n))	0	0.5	0.6875	0.8125	0.8906
FRR	0.12	0.11	0.1	0.1	0.09
FAR	0.0012	0.0011	0.001	0.001	0.0009

created at enrollment. It is also noteworthy that the FAR does not increase with the error correction applied on the P-Hash vectors.

Our experimental results show that the P-Hash from a given iris image is computed in 0.105 seconds on average. The SVM model with 400 training instances was built in 8 seconds on average (once the suitable values for C and γ had been identified) and a given testing instance was classified in 0.013 seconds on average. Please note that the results related to the accuracy of classification mostly depend on the particular feature extraction algorithm used in the feature extraction phase of the proposed protocol. In our future work, we will explore other feature extraction mechanisms to further increase the accuracy of classifying biometric images.

5 Security Analysis

In analyzing the security of our approach, we identify the security and privacy related properties of the proposed protocol and related techniques. We also identify potential attacks against the proposed identity management protocol and show that the proposed protocol resists such attacks by preserving the desirable properties that the attackers try to compromise.

All three privacy sensitive elements related to the user's biometric identity (i.e: the biometric image, the P-Hash, and the generated BID) which are used in the intermediate steps of the enrollment and authentication phases of the protocol are discarded after the IDT is generated, and are not stored anywhere or transmitted to other parties. This ensures confidentiality of the user's biometric image, the P-Hash, and the BID which possess the uniqueness feature of the user's biometric identity. Since the three secrets used for creating the BID, the Pedersen commitment, and the meta-data used for error correction of the P-Hash are derived from a password provided by the user, and thus do not need to be stored anywhere as explained in Section 2.6, confidentiality of those secrets is assured as long as the password is kept secret by the user. The token verification process also assures the confidentiality of the secrets because of the use of the ZKPK protocol. The proposed protocol is also secure against replay attacks by external parties because each time a new challenge and new random values y and s are chosen during the proof of knowledge. The reason why we do not use distance matching as the biometric authentication mechanism coupled with P-Hash as mentioned in Section 2.1, is to protect the P-Hash of the user's

biometric from being exposed to the service provider at authentication. However, as mentioned in Section 3.1.3, an external attacker or a malicious service provider can impersonate the user through a man-in-the middle (MITM) type of attack on the zero knowledge proof of identity commitments which is known as Mafia Fraud attack or Chess Grand Masters' Problem [13]. While a MITM attack carried out by an external attacker can be avoided with the proper use of secure communication channels such as SSL, preventing such attacks by only using such secure communication channels is not possible when the attacks are carried out by malicious service providers. In order to prevent such an attack by a malicious service provider, the unique identity (e.g., a registered name) of the service provider with which the user actually intends to interact is explicitly bound to the IDT itself. Users can request the IDP to do this binding in step 1 of Protocol 2a (section 3.2). In order to prevent the IDP from learning which service providers the users interact with, the users can create a commitment of the name of the service provider and request the IDP to bind that commitment to the IDT, instead of the actual name of the service provider. An example of a simple and efficient commitment scheme which can be used for this purpose is PRNG(name of SP | time stamp) where PRNG denotes a secure pseudo random generator. The time stamp is used to make the multiple commitments created with the name of the same service provider different. At authentication, when the user opens this commitment and proves the committed value, the service provider can confirm that it is not a MITM impersonation attempt carried out by another malicious service provider and that the genuine user actually intends to communicate with it.

The SVM is also a sensitive source of information since it encodes the labels of all the classes in the trained SVM which becomes part of the BID of a user. We have taken two steps to prevent attacks based on the SVM model. The first is to store the SVM in encrypted form in the hardware based TEE which can only be accessed by the authentication client application. The second is to randomly change the class labels in the base SVM model to obtain a customized SVM model for each user. Such step guarantees that an attacker, who gets hold of one SVM model, would not be able to figure out the possible class labels related to the BIDs of all the other users registered in the identity management system. Our approach for customizing the base SVM model for each user also enhances the revocability of the IDT. If a compromise happens, the user is able to cancel the already registered IDT for his/her biometrics and register a new IDT created with a new BID obtained from a new customized SVM model, or with a new secret or with new values for both. Any party which gets hold of the IDT, which is the only information exposed related to the user's biometric identity in this protocol, cannot successfully authenticate without being able to provide both the biometric and the password used to derive the secrets. The IDT thus provides ownership assurance of the biometric identity of the user which prevents identity theft and impersonation. The service providers also have assurance about the validity of the IDT by verifying the digital signature of the IDP on the IDT.

Therefore, we can observe that our approach and the proposed protocol preserves the confidentiality of sensitive information related with users' biometrics and the secrets used to generate the IDT and assures ownership of the biometric identity, and revocability and validity of the identity tokens. It also protects users' privacy against honest but curious (semi-honest) IDPs by not involving the IDP in the transactions between the user and the service provider, while being secure against adversaries which steal users' devices, eavesdroppers in the network and even the malicious service providers.

6 Related Work

A privacy preserving user centric identity management system has been previously proposed by Paci et al. in [9] to enable multi-factor identity verification, which is closely related to our work. The main difference is that our work supports biometrics-based authentication, whereas such previous work only supports non-biometric identifiers. Our approach of using SVM classification is closely related with the work by Bhargav-Spantzel et al. [3]. Our approach uses the P-Hash of users' biometrics as input to the SVM classifier and improves the accuracy in classifying biometric samples with respect to such previous work. The high accuracy classification results shown in [3] are due to fact that the same set of biometric images were used in both training and testing phases.

Another improvement in our approach is that we customize the trained SVM model for each user, which has several advantages compared to the use of one global SVM model proposed in [3]. In addition the BID generation approach proposed in [3] makes use of multiple class labels and expects the probability estimates of prediction confidences across multiple class labels to be repeatable in the output of the SVM classification, which does not happen practically based on the observation of our experiments. Instead, we use the single class output of the SVM classification and concatenate it with a sufficiently large key (S_1) to make the BID resistant against brute force attacks.

The error correction mechanism that we have used in the extended protocol to improve the repeatability of the created biometric identity is based on the work by Kande et al. [6]. We make use of only the first phase of their approach for reducing errors between two biometric samples of the same user (i.e: genuine errors). In contrast to their approach that uses Hamming distance based comparison to identify impostor errors between two biometric samples, which has certain drawbacks such as having to reveal the biometric feature vector to the verifier at authentication time, our approach uses SVM based classification which avoids such drawbacks. Another difference is that we do not store the key used in the error correction algorithms whereas the approach by Kande et al. requires to store the hashed version of the key which is generated during the encoding process to be used in the decoding process.

7 Conclusions and Future Work

In this paper, we have presented a novel and secure approach for user-centric biometrics-based authentication which preserves user's privacy. Since a real world authentication system needs higher accuracy, in our future work we will explore other feature extraction mechanisms which would help in further improving the repeatability of the BID while preserving its uniqueness. We plan to carry out further experiments to measure the performance of the proposed protocol in terms of computational time, resource consumption, and communication overhead in other types of user devices such as mobile phones. We also plan to extend this work by generalizing the proposed improvements.

References

1. IdentityX | World-Class Mobile Biometric Authentication, http://www.identityx.com
2. UBIRIS, http://iris.di.ubi.pt/
3. Bhargav-Spantzel, A., Squicciarini, A.C., Bertino, E., Kong, X., Zhang, W.: Biometrics-based identifiers for digital identity management. In: IDtrust 2010 Conference Proceedings. ACM (April 2010)
4. California State University, East Bay: Coding theory - hadamard codes, http://www.mcs.csueastbay.edu/~malek/TeX/Hadamard.pdf
5. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
6. Kande, S., Dorizzi, B.: Cancelable iris biometrics and using error correcting codes to reduce variability in biometric data. In: Computer Vision and Pattern Recognition. IEEE (April 2009)
7. Klinger, E., Starkweather, D.: phash.org: Home of pHash, the open source perceptual hash library (2008-2010), http://www.phash.org/
8. Kostiainen, K., Ekberg, J., Asokan, N., Rantala, A.: On-board credentials with open provisioning. In: Proceedings of ASIACCS 2009 (2009)
9. Paci, F., Bertino, E., Kerr, S., Lint, A., Squicciarini, A.C., Woo, J.: VeryIDX - A digital identity management system for pervasive computing environments. In: Brinkschulte, U., Givargis, T., Russo, S. (eds.) SEUS 2008. LNCS, vol. 5287, pp. 268–279. Springer, Heidelberg (2008)
10. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)
11. Proença, H.: The UBIRIS.v2: A database of visible wavelength images captured on-the-move and at-a-distance. IEEE Trans. PAMI 32(8), 1529–1535 (2010)
12. Proença, H., Alexandre, L.A.: UBIRIS: A noisy iris image database. In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 970–977. Springer, Heidelberg (2005)
13. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd edn. Wiley (1996)
14. Zauner, C.: Implementation and Benchmarking of Perceptual Image Hash Functions. Master's thesis, Upper Austria University of Applied Sciences, Hagenberg Campus (2010)