CERIAS Tech Report 2017-4

PrivBioMTAuth: Privacy Preserving Biometrics-Based and User Centric Protocol for User Authentication from Mobile Phones

by Hasini Gunasinghe, Elisa Bertino
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

PrivBioMTAuth: Privacy Preserving Biometrics-Based and User Centric Protocol for User Authentication from Mobile Phones

Hasini Gunasinghe*, and Elisa Bertino†
Department of Computer Science, Purdue University, West Lafayette, Indiana, USA.
Email: *huralali@purdue.edu, †bertino@purdue.edu

***Abstract*—We introduce a privacy preserving biometrics-based authentication solution by which users can authenticate to different service providers from mobile phones without involving identity providers in the transactions. Authentication is performed via zero-knowledge proof of knowledge, based on a cryptographic identity token that encodes the biometric identifier of the user and a secret provided by the user, making it three-factor authentication. Our approach for generating a unique, repeatable and revocable biometric identifier from the user's biometric image is based on a machine learning based classification technique which involves the features extracted from the user's biometric image. We have implemented a prototype of the proposed authentication solution and evaluated our solution with respect to its performance, security and privacy. The evaluation has been performed on a public dataset of face images.**

I. Introduction

Assurance on the verification of the user's identity and on privacy and security of the user's identity are key requirements for remote authentication to online services through the users' mobile phones. Biometrics is a stronger authentication factor compared to traditional password and hardware tokens based authentication factors. However, biometrics as an authentication factor has both advantages, such as uniqueness and permenance, and disadvantages, such as non-repeatability (the same biometric traits of a user captured at two different times are not identical) and non-revocability (biometrics are permenantly coupled with the user). Therefore, an authentication mechanism built using biometrics as the authentication factor needs to address the disadvantages while preserving the advantages. The key requirement is thus to develop techniques supporting the derivation from the user's biometric template of a biometric identifier (BID) which is unique, repeatable and revocable.

As today mobile phones include various biometric sensors, major service providers, such as banking institutions [1], credit card companies [2] and e-commerce organizations [3], are adopting biometrics based authentication. However, existing biometrics based remote authentication systems have security and privacy issues. Service providers implement their proprietary biometrics-based authentication mechanism based on different biometrics traits such as fingerprints and face. Such mechanisms typically require users to enroll their biometrics with the service provider (SP). Upon user enrollment, a biometrics-based authentication system then records 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 and the authentication channel is thus critical for the privacy of biometrics. If the biometric template database and/or the biometric template provided at authentication are compromised, users may lose their biometric identity permanently unless the SP adopts techniques for cancellable biometric identity [4].

Security and privacy risks increase when users have to provide their biometric information to multiple SPs. Such risks could be addressed by using an identity provider centric authentication solution. In such identity management architecture, the user initially enrolls her/his biometrics at a trusted authority usually referred to as identity provider (IDP). When the user needs to authenticate to a third party SP, the SP contacts the IDP for the biometrics-based authentication of the user. The SP thus relies on the IDP for authenticating the user so that the user does not to have to register and reveal her/his biometrics at the SP, thereby better protecting her/his biometric identity. However, such an authentication solution raises other types of privacy concerns. Because the IDP is involved in each transaction, it can infer sensitive information, such as users' transaction patterns with different SPs. Today there are commercial products [5] which support biometrics-based user authentication based on the IDP centric architecture. However, they do not address such privacy concerns.

User-centric identity management architectures, on the other hand, address such concerns as they do not require the involvement of the IDP in the authentication of the users when executing transactions on the SPs. Under such architectures, after the initial enrollment with the IDP, the user can authenticate to the SP in a secure manner, without involving the IDP. In the VeryIDX system [6], for example, upon enrollment of the user's static identities, such as email address, credit card number, social security number etc., at the IDP, the user is given some cryptographic authentication artifact using which the user can authenticate directly to the SP without having to disclose passwords or other authentication information to the SP. This type of secure, privacy preserving and user-centric authentication is achieved with the use of Zero Knowledge Proof of Knowledge (ZKPK) [7] and cryptographic commit-

ments [8].

The design of such identity management architecture is however challenging when dealing with biometrics-based authentication through users' mobile phones, due to several reasons. Firstly, unlike static identities, because of the non-repeatable nature of biometrics, the genuine owner of the biometric identity might fail the ZKPK based authentication. The reason is that one needs to be able to re-generate the exact same secrets at both enrollment time and authentication time in order for the ZKPK based authentication to succeed. Second, 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. Therefore, the authentication artifacts given to the user at the end of the enrollment phase should not make the authentication system vulnerable to attacks. Third, ZKPK based authentication protocols are inherently vulnerable to special type of impersonation attacks called Mafia Fraud [9] which needs to be prevented for strong assurance on security and privacy of the user's biometric identity.

The goal of this paper is to propose a secure, privacy preserving and user centric protocol for authenticating users from their mobile phones to online SPs based on the users' biometric identity. Our proposed protocol, does not have the drawbacks that we have discussed, namely: (i) it does not require storing and transmitting users' biometrics at multiple SPs as it involves a trusted IDP to enroll users' biometric identity (ii) it does not even require storing the user's biometrics at the IDP and (iii) it does not require involving the IDP in each authentication attempt as it adopts a user centric identity management architecture. It also addresses the challenges of biometrics-based and user centric authentication carried out from mobile phones by: (i) deriving a unique, repeatable and revocable BID from the users' biometrics (ii) employing secure authentication artifacts to be stored in the mobile device and (iii) including a key agreement mechansim tied to the authentication protocol to mitigate Mafia attacks.

The main contributions of our work can be summarized as:

(i) A secure, privacy preserving and user centric authentication protocol based on biometrics for authenticating the users to online services from the users' mobile phones.
(ii) A prototype implementation of the proposed authentication system.
(iii) An experimental evaluation of the solution with respect to different metrics. Note that in Section III, we present the generic solution which does not depend on any particular biometric trait, feature extraction mechansim or learning algorithm. In our experiments, we use face as the biometric trait, eigen faces as the feature extraction mechanism and SVM as the learning algorithm, which we describe in detail in Section IV.
(iv) A security and privacy analysis of the authentication protocol.

The rest of the paper is organized as follows. Section II introduces the main concepts used in our approach. Section III explains our approach in detail. We present the details of the prototype and the experimental evaluation in Section IV. We analyze security and privacy in section V. We discuss related work in Section VI and outline conclusions and future work in Section VII.

II. Background

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

A. Eigen Faces based feature extraction

There are two main categories of face recognition techniques namely: (i) appearance based and (ii) geometric facial feature based [10]. The Eigen faces based face recognition technique [11] belongs to the first category. The Eigen faces based feature extraction mechanism extracts features by projecting face images on to a feature subspace called "eigen faces". This subspace is computed by applying Principal Component Analysis (PCA) on a set of training face images. PCA is a dimension reduction method which projects n dimensional data onto K dimensional subspace where $K < n$. This K dimensional subspace identifies the dimensions with the maximum variance.

In order to compute the eigen faces based features of a given image I, we need to first compute the "eigen faces" subspace denoted by W, and the mean image X_{mean} from a set of training face images X, and then project the given image onto W, after substracting X_{mean} from I. In what follows, we discuss how the computation of these two stages is performed.

1) Computing the eigen faces subspace [10]: Each face image in the training data set, which is represented as a pxq matrix of pixel values, is converted into a vector of $p*q$ rows. Let $X = \{x_1, x_2, x_3, .., x_n\}$ be a matrix containing the vector representations of n such face images.

(i) Compute the mean: $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$
(ii) Compute the covariance matrix:
$$S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T$$
(iii) Compute the eigen vectors v_i and eigen values λ_i represented by the following equation:
$$Sv_i = \lambda_i v_i$$
(iv) Normalize the eigen vectors.
(v) Order the eigen vectors in descending order by their eigen values and select the K eigen vectors corresponding to the K largest eigen values. These K eigen vectors form the eigen faces subspace which is referred to as W.

2) Projecting an image onto eigen faces subspace: Given a face image I, its eigen faces based features is extracted via the following steps, using the eigen faces subspace W and the mean image $X_{mean}(= \mu)$ computed in the previous stage.

1) Normalize the image : $I_N = \frac{I}{\|\mathbf{I}\|}$
2) Substract the mean image of the training set:
$$I_S = I_N - X_{mean}$$
3) Project I_S onto W : $F_I = W^T I_S$

F_I is the set of features extracted from the image I via the eigen faces based feature extraction mechanism which can then be used for face recognition tasks.

B. 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 label 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 the 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 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 [1] selected based on k-fold cross validation accuracy in grid search, as we discuss in Section IV.

C. Pedersen Commitment

The Pedersen commitment [8] is a secure commitment scheme whose security is based on the hardness of solving discrete logarithms. The operation of this 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 integer group of order p. A trusted party chooses g - a generator of G_q and h (= g^amod 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: $C(x, r) = g^x h^r$ 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 one can compute $log_g h$.

D. 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. Let U denote the committer and V denote the verifier.

[1] 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.

Protocol 0 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 $g^u h^v = dC^e$.

E. Key Derivation from a Password

Our approach uses three secrets ($S_i : i \in \{1, 2, 3\}$) in different steps of the protocol: S_1 of size 128 bits, S_2 of size 160 bits, and S_3 of size 256 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 two 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 (=544 bits)).
We then partition S into three parts of aforementioned sizes in order to form the three secrets.

III. Approach

Our authentication approach involves three entities, namely: (i) user - the entity which is authenticating using the biometric identity, (ii) service provider (SP) - the entity which authenticates the user before allowing the user to perform any transactions, and (iii) identity provider (IDP) - which vouches for the user's biometric identity. Our approach consists of two main phases: (i) enrollment phase - by which the user obtains her/his biometrics-based cryptographic identity token digitally signed by the IDP; (ii) authentication phase - by which the user proves her/his biometrics-based identity at the SPs.

A. Enrollment Phase

During the enrollment phase, a user is given: i) an identity token (IDT) digitally signed by the IDP, which encodes in a cryptographic commitment a secret derived from the user's biometrics and a secret derived from the user's password and ii) some secure artifacts. These secure artifacts enable the user to regenerate the secrets during the authentication phase.

In what follows we discuss the key challenges in designing the enrollment phase followed by a detailed description of the enrollment protocol. In the discussion we refer to any enrolled

user in the authentication system that is different to a particular enrolled user of interest, as an *imposter*.

1) Deriving the biometric identifier (BID): As we discussed in Section I, due to the dynamic nature of biometrics, deriving a repeatable secret from a user's biometric template is a challenge. This secret should also be unique (so that it is hard to be guessed and bruteforced) in order to preserve the inherited uniqueness of raw biometrics and revocable in order to cancel it in case of a compromise. The technique that we use to obtain a repeatable BID from a user's biometrics is to train at enrollment a multi-class classification based machine learning model which predicts the class label that best represents the enrolling user's biometric features. Depending on its robustness, the trained classifier is expected to predict the same class label at all the authentication attempts by a specific enrolling user and to predict a different class label at the authentication attempts by an *imposter*. We decided to use multi-class classification approach as opposed to binary classification approach because of the requirement that the class label associated with an enrolling user must be unique and hard to guess. Such requirements would not be met by a trained binary classifier which outputs either 0 or 1. A random set of binary strings that are 128 bits long, is selected as labels of the training biometric features used to train the multi-class classifier. The BID is created by concatenating the class label (l), which is 128 bits long, predicted by the trained classifier (on an input of biometric features of a specific enrolling user), with the secret $S1$, which is also 128 bits long, derived from the user's password through function PBKDF2. Therefore, the BID takes the format in equation 1 and it is 256 bits long (BID mod $q \in Z_q$).

$$BID = l|S1 \tag{1}$$

The user can revoke an existing biometric identity derived using the aforementioned approach and request the IDP to issue a new IDT that encodes a new BID generated using a new password and a new trained classifier which associates a different class label with the user's training biometric features. Therefore, the BID generated according to our approach is repeatable, unique and revocable.

2) Selecting training data to train the classifier: The method for selecting the training data to train the classifier needs to take into account both security and usability. On one hand, since popular multi-class classification techniques, such as kernel based SVMs, encode the training data in the trained model and since the trained model is stored in the user's mobile phone in order to support user centric authentication as we discussed in Section I, preserving the security and privacy of the training data is important. Although the classifier will be stored securely in the user's device, it is critical to encode the minimum amount of sensitive data in the trained classification model to minimize the impact on the authentication system in case in which the user's mobile device is stolen and the classifier is compromised. On the other hand, the training data used to train the classification model should be desciminative in order to make the trained model robust enough to map at the authentication the correct class label with the genuine user's biometric features and to not map the genuine user's class label with the *imporsters'* biometric features.

We experimented with three potential methods for selecting the training data to train the classifier for a specific user enrolling in the authentication system. We have carried out experiments to empirically evaluate the performance of the classifiers trained with the data obtained from those three methods in terms of False Rejection Rate (FRR) and False Acceptance Rate (FAR).

Note that in the context of the multi-class classification model that we use, FRR is the rate at which the trained classification model predicts a class label different than the one associated with the genuine user's biometric features at enrollment, when the genuine user's biometric features is given as input to the model at authentication. Usability decreases as FRR increases because the genuine user finds difficulties in authenticating. FAR is the rate at which the trained classification model outputs the class label that is associated with the genuine user's biometric features at enrollment, when an *imposter's* biometric features is given as input to the model at authentication. An authentication application becomes less secure as FAR increases because the probability that an *imposter* authenticates as the genuine user increases.

TABLE I
FALSE REJECTION RATES OVER FOUR TRIALS

Method	Trial 1		Trial 2		Trial 3		Trial 4	
	FRR	STDV	FRR	STDV	FRR	STDV	FRR	STDV
Method 1	0.556	0.233	0.644	0.257	0.622	0.239	0.622	0.239
Method 2	0.022	0.083	0.044	0.113	0.022	0.083	0.044	0.113
Method 3	0.05	0.1	0.033	0.084	0.033	0.084	0.033	0.084

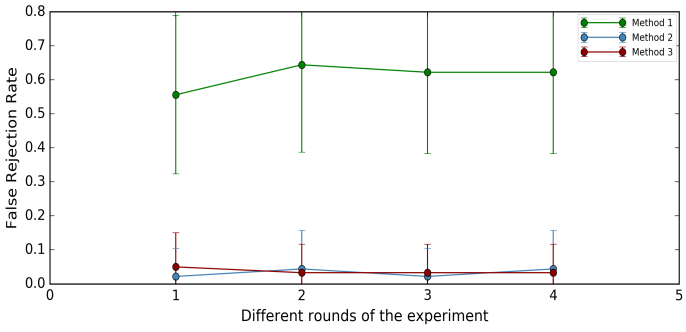

Fig. 1. False Rejection Rates over four trials

TABLE II
VARIATION OF FALSE ACCEPTANCE RATES WITH NUMBER OF IMPOSTERS

Method	3 Imposters		6 Imposters		9 Imposters		12 Imposters	
	FAR	STDV	FAR	STDV	FAR	STDV	FAR	STDV
Method 1	0.322	0.171	0.341	0.102	0.348	0.057	0.352	0.053
Method 2	0.0	0.0	0.022	0.036	0.053	0.065	0.056	0.044
Method 3	0.005	0.020	0.008	0.022	0.007	0.015	0.002	0.010

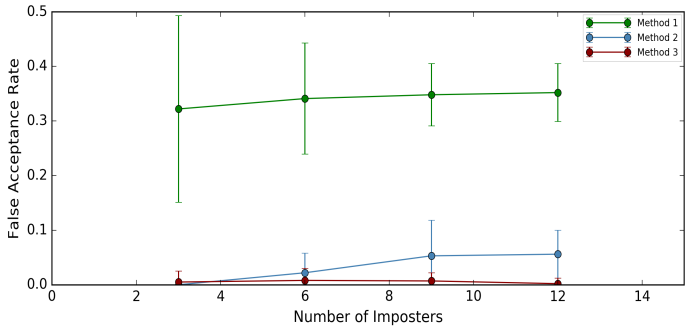

Fig. 2. Variation of False Acceptance Rates with number of imposters

Based on the experimental results on the three methods summarized in the Tables: IX and X and illustrated in the Figures: 6 and 7 (please refer the Appendix for details on the three methods (Appendix-A) and their experiment results (Appendix-B) summarized here), we selected method 2 which achieves a balanced trade-off between security and usability, in order to select the data to train the machine learning model. This trained model is then used to derive the BID from the user's biometrics at authentication. In method 2, the training data is selected as the biometric features of a set of random users, who are outside the authentication system (so that they do not overlap with the space of potential *imposters*), and the biometric features of the current enrolling user.

With this method, we assume that the class label associated with a particular enrolling user's biometric features in the training data will be output by the trained model at the authentication attempts of the same enrolling user and that an *imposter's* biometric features will be matched with any of the random users' biometric features used to train the model. The only sensitive biometric features included in the classifier is the biometric features of the currently enrolling user. Therefore, in case of a compromise of the trained classifier, the biometric features of the other users enrolled in the authentication system will not be exposed. This method also allows us to achieve the desired FAR and FRR by varying the number of random users whose data is used to train the model (see Section B for experimental details).

Remarks: There are several approaches which can be used by the IDP to construct the dataset with random set of biometric features for training: (i) selecting one out of the many publicly available datasets; (ii) creating a set of synthetic biometric images; (ii) collecting a set of biometric images by the IDP itself for training purposes from a population which does not overlap with the population of enrolled users. In second and third approach, the IDP keeps the random biometric data set secret as such data sets become IDP's proprietary data. As we have already mentioned, the only requirement of such random biometric images is that, they are retrieved from subjects outside the authentication system, that is, subjects who are not the enrolled users of the authentication system. Therefore, the probability that the adversary knows which random images are involved in the training data of a particular users' classifier is negligible.

3) Enrollment Protocol: In what follows we discribe the complete enrollment protocol designed based on the aforementioned design decisions. The enrollment protocol is executed between the user and the IDP and the steps are listed in Protocol 1. Note that when we refer to these entities, both human and software aspects related to them are involved. For example, when we refer to the user in the protocol, we refer to the actions taken by both the human user and the software installed in user's device.

When a request for enrolling a biometric identity is received at the IDP, along with the user inputs mentioned in Protocol 1, the IDP first selects a set of biometric images from a random set of users, following the method described in A, in order to train the classification model for the current enrolling user. The number of such random users is decided based on the required assurance on FRR and FAR of the authentication application, as discussed in Section B. Then the IDP extracts biometric features from the biometric images of both the enrolling user and the selected random users, using an appropriate feature extraction mechanism. The training data is constructed by assigning a random integer class label to the biometric features of each user and the classifier is trained using an appropriate learning algorithm.

In step 4 of protocol 1, the IDP generates a random salt value which is given as input to the PBKDF2 function in step 5, along with the user's password and the required length of the secret to be derived. Then the BID is constructed in step 6 as described in Section III-A1. In step 7.i, the cryptographic commitment (C) is created by committing the two secrets, that is, the BID and the second secret derived from the user's password ($S2$), in the Pedersen commitment scheme. We leverage the properties of Pedersen commitment scheme described in Section II-C to hide the BID and $S2$ in the IDT. Finally, the IDT is created as the concatenation of the commitment, meta-data included in the IDT, and the digital signature of the IDP on the content of the IDT. Meta-data may include: serial number of the IDT, the public parameters of the Pedersen commitment (to be used in the authentication protocol), the expiration timestamp and any meta-identity information provided in the user input. The meta-identity information helps the SP to identify the user via some other identity attributes such as name, email, social security number etc., at authentication. We utilize the standard PKI based digital signature for digitally signing the IDT. The secure channel mentioned at the information exchange steps of Protocol 1 refers to a channel with message level security.

In order to allow a user to perform user-centric biometrics based authentication from her/his mobile phone without involvement of the IDP, the IDT, the trained model, the salt value (T) used in deriving the secrets based on the user's password, and the trusted software that executes the user's part of the authentication protocol are provided to the user by the IDP at the end of enrollment. The artifacts that contain sensitive information, such as the trained model are encrypted using the private key of a key pair generated for each enrolling user by the IDP. This key pair is stored in a keystore which is

Protocol 1 Enrollment Protocol

Input from user: biometric images, password, meta-identity information (optional).

Input from IDP: biometric images of random set of users, private key of the IDP, public parameters of the Pedersen commitment.

Output from IDP: digitally signed IDT, trained model, salt-value for PBKDF2, key store.

Protocol execution:

User		IDP
	1. input of the user $\xrightarrow{\text{(over a secure channel)}}$	
		2. $F \leftarrow$ Training data prepared by extracting features from biometric images and assigning labels. ($l \leftarrow$ class label assigned to the training features of the enrolling user)
		3. $M \leftarrow$ Classifier trained with with F.
		4. $T \leftarrow$ Generated random salt value.
		5. Derive three secrets from the user's password: S1, S2, S3 $\leftarrow$ PBKDF2(PKCS#5, password, T, derived key length)
		6. BID $\leftarrow l \vert S1$
		7. Create IDT: i. Commitment (C) $\leftarrow g^x h^r modp$; where x = BID and r = S2. ii. IDT $\leftarrow$ C $\mid$ meta-data $\mid$ Signature of IDP on {C $\mid$ meta-data}
		Generate a key pair, encrypt the trained model, protect the key store with S3.
	8. output of the IDP $\xleftarrow{\text{(over a secure channel)}}$	

password-protected by S3. These secure artifacts are stored in the sandboxed internal storage of the IDP software application or in the Trusted Execution Environment (TEE) enabled in the modern mobile devices [12].

Remarks:

- The main motivation for using the user's password to generate the three secrets (S1, S2 and S3) used in the protocol is to avoid having to store the secret keys on the device and instead having them generated based on input by the actual user, each time the authentication is performed. This mitigates the risk of an attacker being able to steal the secrets kept in a storage.
- In order to prevent a malicious user from providing fake biometric images instead of her/his own biometric images in user input, for high assurance authentiation, the enrollment protocol should be executed at the IDP following the necessary legal processes such as requiring the user to visit the authority in person and proving her/his identity using a legal identifier that she/he possesses, such as a passport, which is outside the technical scope of our current work.

B. Revocation of the biometric identity

Revocation of an issued IDT can be initiated by either the IDP or the user. An IDT which is expired based on the expiration timestamp issued by the IDP at enrollment is also considered as revoked. Once an existing IDT is revoked, the user may request a new IDT by initiating the enrollment protocol and the IDP may issue a new IDT as mentioned in Section III-A1, by creating a new commitment using a new password provided by the user and a new trained classifier.

It is critical that SP be able to check for IDT revocation without undermining user's privacy. There are three main requirements: 1) user privacy - the identity of IDT owner should not be revealed to the IDP unless the IDT is in the revocation list; 2) revocation list hiding - the IDTs in the revocation list should not be revealed to the SP; 3) SP authorization - the SP cannot execute an arbitrary number of revocation checks without detection. The most common revocation checking mechanism is the Online Certificate Status Protocol (OCSP), in which the verifier sends the serial number of the certificate to the revocation authority and obtains the decision whether or not the given certificate is revoked. This violates the first requirement mentioned above, as it exposes a legitimate user's transaction patterns to the IDP. On the other hand, the IDP can let the SP obtain the entire revocation list to perform revocation check locally. While this addresses the first requirement, it violates the second and third requirements. The second and third requirements are important because someone who steals an arbitrary number of IDTs should not be able perform revocation checks on them in order to learn which of them are valid, without the risk of being detected. Therefore, a privacy preserving revocation checking mechanism that addresses all three requirements should be coupled with our authentication scheme. There are several

approaches that address privacy-preserving revocation [13], [14], [15]. Both the approaches proposed in [13], [14] require the SP to query a range of K elements instead of the single token being verified. The degree of privacy depends on the size of this interval [15].

On the other hand, the approach proposed in [15] is based on an efficient private set intersection scheme [16]. The protocol consists of two phases: $Init$ and $Query$. During the $Init$ phase, the IDP creates a revocation list that hides the serial numbers of the revoked IDTs and sends it to the SP. During the $Query$ phase, the SP sends the blinded serial number of the IDT being verified, to the IDP, who then includes a RSA blind signature on it. The SP then unblinds the signature and checks it against the list sent by the IDP during the $Init$ phase. If there is a match, the IDT being verified is revoked. We refer the reader to [16] for further details. Communication complexity and computation complexity for the issuer of the $Init$ protocol is in the order of the size of the revocation list. No computation is required from the verifier in the $Init$ protocol. Both communication and computation complexity is constant for the $Query$ protocol. The cost of the $Init$ protocol is amortized if the same version of the revocation list is used over multiple runs of the $Query$ protocol. Such a scheme can be directly integrated into PrivBioMTAuth, as it addresses all the three requirements mentioned above.

C. Authentication Phase

At authentication, the user proves her/his biometric identity to the SP by proving the ownership of the IDT signed by the trusted IDP. This is accomplished by proving in zero knowledge, the knowledge of the two secrets, that is, the BID and the secret derived from the user's password ($S2$), that are encoded in the cryptographic commitment of the IDT. Accordingly, this is a three-factor authentication protocol which involves the user's IDT, her/his biometric image and the password. In what follows we present the basic authentication protocol in which the client authenticates to the SP, and then we extend it into a stronger authentication protocol which includes a key agreement phase that enables both parties to derive a secret key to verify authenticity of each other and encrypt the subsequent communications.

1) Basic Authentication Protocol: As shown in Protocol 2a, the user initiates the authentication protocol by sending the authentication request along with the IDT, a helper commitment d and optionally, some meta-identity information. d encodes two random secrets y and s in a Pedersen commitment, which helps later in the protocol in proving the actual secrets encoded in the IDT. The SP first checks the validity of the IDT by verifying the signature of the IDP, the expiration timestamp, the revocation list and any meta-identity information sent by the user against the information included in the signed IDT. The meta-identity information may help the SP to associate the authenticating user with the user account held at the SP. Next the SP creates a challenge and sends it to the user. To prove the knowledge of the secrets against the challenge, the user first re-generates the BID using a newly captured biometric image, the password, the trained model, and the salt value, as shown in the steps 5-10 in Protocol 2a. Then the user computes the proofs u and v as shown in step 11 and sends them to the SP. Upon verifying the zero knowledge proof as per step 13, the SP accepts or rejects the user's biometric identity based authentication. If the authentication succeeds, the user and the SP establish a session to carry out the transaction.

Based on the properties of standard ZKPK protocol described in Section II-D, any information transmitted from the user to the SP in Protocol 2a does not help the SP to learn any additional information about the secrets encoded in the IDT. However, it helps the SP to verify with confidence whether or not the prover is the actual owner of the biometric identity encoded in the IDT.

2) Limitations of the Basic Authentication Protocol: Although none of the information exchanged in Protocol 2a is sensitive, we have to rely on Transport Layer Security (TLS) protocol, such as HTTPS, in order to verify the identity of the SP whom the user is interacting with and to establish a secure communication channel for exchanging any sensitive information of subsequent transactions. This is similar to the use of TLS in traditional user name and password based authentication protocols. However, even if Protocol 2a is followed by TLS, security issues still arise.

(i) Identity theft attack by a malicious SP.
This is a known man-in-the-middle type attack on ZKPK based identity verification protocols [9], also known as Mafia attack. This attack can be carried out by a malicious SP with whom the user performs a ZKPK based identity verification. When the user sends an authentication request to the malicious SP, this SP simultaneously initiates an authentication request to some other genuine SP, claiming the user's identity. When the genuine SP sends the challenge, the malicious SP simply forwards it to the user. When the user submits the identity proof, the malicious SP uses this proof to authenticate to the genuine SP by impersonating the user. (Protocol A in Appendix-C lists the steps of this attack based on the steps of the standard ZKPK based identity verification). This type of identity theft attack is possible because the basic authentication protocol does not include a mechanism for the two parties to verify that a man-in-the-middle attack has not taken place during authentication, before carrying out the transaction.

(ii) Exposure of sensitive information about the transactions at the intermediaries of the communication path.
Because TLS only protects communication at the transport layer, confidentiality and integrity is not guaranteed at the intermediaries of the communicatin path. This could lead to certain attacks, such as session hijacking attacks, in which a malicious party can steal the session id of the user (which is provided by the SP at the end of the successful execution of Protocol 2a) and perform transactions on behalf of the user, using the stolen session id. Therefore, it is preferable to have message level security in order to secure sensitive information.

Protocol 2a Basic Authentication Protocol

Input from user: IDT, helper commitment ($d = g^y h^s mod p \in G_q$; where $y, s \in Z_q$ are random secrets), biometric image, password, meta-identity information(optional)

Output from SP: Authentication result: success/failure

Protocol execution:

User		SP
	1. authentication request with IDT, d and meta-identity $\longrightarrow$	
		2. Verify the validity of the IDT.
		3. Create a random challenge: $e \in Z_q$
	$\longleftarrow$ 4. challenge: e	
5. I $\leftarrow$ Newly captured biometric image.		
6. $S1'$, $S2'$, $S3' \leftarrow$ PBKDF2(PKCS#5, password, salt-value(T), derived key length)		
7. Decrypt the trained model (by using $S3'$ to open the keystore).		
8. f $\leftarrow$ Features extracted from I.		
9. $l' \leftarrow$ Predicted class label for f.		
10. $BID' \leftarrow l'\|S1'$		
11. Computes: $u = y + ex$ and $v = s + er$; where x = BID' and r = $S2'$		
	12. u and v $\longrightarrow$	
		13. Verifies if: $g^u h^v = dC^e$; where C is the commitment in the IDT.
	$\longleftarrow$ 14. Authentication result	

3) Extended Authetication Protocol: The solution to those issues associated with the basic authentication protocol is to integrate a key agreement mechanism with the identity verification phase which serves two purposes, namely: i) helps mitigating the man-in-the-middle impersonation attack by allowing the user and the SP to verify the authenticity of each other ii) establishes a session based key for secure communication. We extend our basic authentication protocol to a strong authentication protocol which is listed in Protocol 2b. The steps that are additional with respect to the basic authentication protocol are shown in bold font. Accordingly, in the step 4 of the identity verification phase, the SP sends two parameters a and b to the user, in addition to the challenge e. In the key agreement phase, the user derives the secret key using a, b and the secrets known to the user, while the SP derives the same key using the random secret w (which was used to create a and b), the commitment C in the IDT and the helper commitment d. The user and the SP then uses the derived key to: i) perform a handshake to verify that the impersonation attack has not taken place during the identity verification phase ii) secure the subsequent communication along with a symmetric encryption scheme (for confidentiality) and a keyed cryptographic hash function (for intergrity). Secure communication enabled by Protocol 2b achieves forward secrecy as the communication is encrypted using per-session keys, any future compromise of the user's BID or the password will not compromise the past session keys, thereby preserving the secrecy of the past communication. A malicious SP could still perform the steps of the impersonation attack until the end of the identity verification phase of Protocol 2b. However, it cannot continue the communication with the genuine SP beyond that point as it can not derive the secret key and hence cannot succeed in the handshake. It is important to note that tying the key agreement phase to the identity verification phase is critical. Otherwise, if the key agreement is performed independently after the identity verification is has been completed, malicious SP can derive two different secrets with the user and the genuine SP separately and carry out the impersonation attack.

Our authentication approach thus achieves the main goals set forth in Section I: i) it avoids storing biometrics either at the IDP or at the SP ii) it avoids the revealing of biometrics at the SP at authentication iii) it avoids the involvement of the IDP at authentication iv) it provides a mechansim to derive a unique, repeatable and revocable BID from the user's biometrics to be used for user-centric authentication and iv) it protects against known attacks on ZKPKP identity verification protocol.

IV. Implementation and Experiments

In this section, we present the architecture of the prototype of our approach and the experimental evaluation.

Protocol 2b Extended Authentication Protocol

Input from user: IDT, helper commitment ($d = g^y h^s$ mod p $\in G_q$; where $y, s \in Z_q$ are random secrets), biometric image, password, meta-identity information(optional)

Output from SP: Authentication result: success/failure

Protocol execution:

User		SP
Identity Verification Phase:		
	1. authentication request with IDT and d $\longrightarrow$	
		2. Verify the validity of the IDT.
		3. i. Create a random challenge $e \in Z_q$ **ii. Create a random** $w \in Z_q$ **iii. Compute** $a = g^w$ mod p and $b = h^w$ mod p
	$\longleftarrow$ 4. e, **a, b**	
5. I $\leftarrow$ Newly captured a biometric image. 6. $S1'$, $S2'$, $S3' \leftarrow$ PBKDF2(password, salt-value(T), derived key length) 7. Decrypt the trained model (by using $S3'$ to open the keystore). 8. f $\leftarrow$ Features extracted from I. 9. $l' \leftarrow$ Predicted class label for f. 10. $BID' \leftarrow l'\|S1'$ 11. Computes: $u = y + ex$ and $v = s + er$; where x = BID' and r = $S2'$		
	12. u and v $\longrightarrow$	
		13. Verifies if: $g^u h^v = dC^e$
	$\longleftarrow$ 14. Authentication result	
Key Agreement Phase:		
15. K_{user} = $a^{(x+y)}.b^{(r+s)}$ mod p = $g^{w(x+y)}.h^{w(r+s)}$ mod p		K_{sp} = $C^w.d^w$ mod p = $(g^x h^r)^w.(g^y h^s)^w$ mod p = $g^{w(x+y)}.h^{w(r+s)}$ mod p
	$\longleftrightarrow$ 16. Secure handshake and communication	

A. Architecture

Figure 8 shows the main components of the authentication system and the flow of the execution of the protocols in high level. Details of the components and interactions among them are given in what follows.

1) Components: There are three main components that represent the three main entities described in Section III. They are: i) the software component in the user's mobile phone ii) the IDP software component and iii) SP software component. The software in the user's mobile phone consists of two main sub components: IDP-client and SP-client. The IDP-client is provided by the trusted IDP and performs enrollment and facilitates authentication. The SP-client is the client application provided by the SP such as the client application provided by banking and e-commerce providers. Several SP-clients can be installed in a user's mobile phone. The software in the user's mobile device is devided into two applications for two reasons: i) *component re-use*: the IDP-client encapsulates the key steps of the user's part of the authentication protocol such as capturing and processing of biometrics, key derivation from the user's password and creation of the helper commitment and the zero-knowledge proofs. All the SP-clients installed in the user's mobile phone can consume this functionality when performing user authentication with their corresponding SPs. This ensures that the authentication related critical functionality is not duplicated and is transparent to the developers of SP-clients ii) *securing user credentials and authentication artifacts*: because the user enters the credentials only at the IDP-client and the authentication artifacts provided by the IDP is only accessed by the IDP-client during the authentication, they are not exposed to third party SP-clients.

We have developped three self-contained modules, namely: Crypto Lib, ZKPK-ID Lib and Biometrics module, which encapsulate different building blocks of the protocols and which are re-used accross multiple components of the solution as illustrated in Figure 8. Further details about these modules

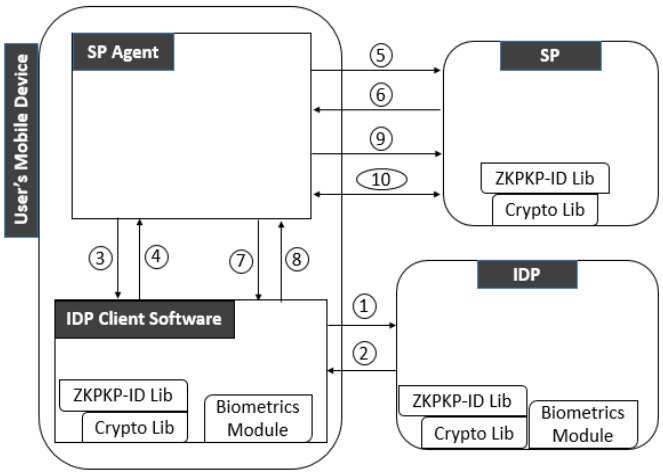

Fig. 3. Overall Architecture of the Authentication Solution

and the implementation can be found in Appendix-D and Appendix-E respectively.

2) Flow: Numbered arrows in Figure 8 illustrate the order of the interactions between the entities during the enrollment and the authentication protocols. Steps 1 and 2 represent the enrollment protocol executed between the user and the IDP, at the end of which the user obtains the authentication artifacts mentioned in Protocol 1. When the user initiates authentication with a particular SP via the corresponding SP-client, the SP-client first sends a request to the IDP-client in step 3, in order to obtain the initial authentication elements, such as the IDT and the helper commitment. These are given to the SP-client by the IDP-client in Step 4. The SP-client then sends an authentication request to the SP in step 5 along with the obtained authentication elements. The SP returns the challenge (and the two parameters used for key derivation if the extended authentication protocol - listed in Protocol 2b - is used) in step 6. The SP-client forwards the challenge (and the two parameters) to the IDP-client in step 7. The IDP-client then prompts the user to enter her/his biometrics and password in order to create the zero-knowledge identity proofs (and the agreed key) which are handed back to the SP-client in step 8. The SP-client then forwards the proofs to the SP in step 9. If the zero-knowledge proof is successful, the SP-client and the SP establish a secure session (or carry out the secure handshake using the derived key, if the extended authentication protocol was used) in step 10 for further communication.

B. Experiments

The goals of our experiments are two folds: i) evaluating the performance of the classifiers trained using the training data obtained using the method described in Section A and observe how the number of random users involved in a classifier affects FRR and FAR ii) measuring the end-to-end execution times our authentication scheme and resource consumption of the mobile applications in order to identify any potential bottlenecks.

1) Experimental Setup: We hosted the IDP and the SP software in an Apache Tomcat web server running in a laptop machine with Ubuntu 14.04 OS, Intel Core i7-3537U CPU, and 5 GB memory. The two mobile applications were deployed in a mobile phone with the model number 'Moto G' [17] and Android version 5.1. The mobile applications communicated with the IDP and the SP web services over a wireless network with a speed of 140 Mbps. We used 'faces' as the biometric trait and utilized the publicly available 'AT&T' face dataset provided by AT&T Laboratories Cambridge [18] which contains face images of 40 individuals with 10 images from each. 'Eigen Faces' was used as the feature extraction mechanism [10] and 60 Eigen components were extracted from each image since most of the variation in the image data was captured in the first 60 eigen components [10]. Therefore, each biometric feature vector consisted of 60 elements. SVM was used as the classification technique and the Java version of LibSVM [19] library was used to implement the classification model. We used the 'C-SVC' SVM type which is intended for multi-class classification and the 'Radial Basis' function as the kernel function. After selecting the training data for each experiment, the optimal pair of values for the C and γ parameters were selected to train the SVM model by evaluating the 5-fold cross validation accuracy (CV accuracy) of each combination of values within the ranges of {-6, 6, 1} for C and {-10, 0, 1 } for γ, using grid search. Finally the SVM classifier was trained using the C and γ values selected with the best CV accuracy.

2) Evaluation of the trained classifiers: In what follows we discuss the experimental evaluation of the classifiers trained using the training data obtained from the method described in Section A, in terms of FRR and FAR.

We performed the experiment by varying the number of random users involved in the trained classifier, for a given number of enrolling users in the authentication system. In this experiment, we assume three authentication systems with varying number of enrolled users (n) as 9, 12 and 15. In each such authentication system, we train six different classifiers for each enrolled user with varying number of random training users (x) as 9, 12, 15, 18, 21 and 24. For each of the six experiments related with each of the three authentication systems, the dataset was divided into two sets: i) set of random users with 40 - n number of users and ii) set of enrolling users with n number of users. The six classifiers of each enrolling user are trained using 6 images of randomly selected x number of users from the set (i) and 6 images of the enrolling user her/him self. Mean FRR was computed over the false rejections made on 3 images out of the 4 testing images of each of the n number of enrolling users by each of their six classifiers. Mean FAR was computed over the false acceptances made on the 4 testing images of n - 1 number of *imposters* by the six classifiers of each of the n enrolling users. FRR and FAR results are reported in the tables III and IV and in the graphs 4 and 5.

Accordingly, FRR stays in a constant range, i.e: below 0.075, for the three authentication systems except for one case

(i.e: the one with 12 random users in the trained classifier) in the authentication system with 9 enrolled users. High standard deviation associated with this case indicates that it is an outlier. There is no specific pattern in the variation of FRR based on the number of random users involved in the trained classifier, in an authentication system with a given number of enrolling users. In contrast, FAR decreases with the increasing number of random users in the classifier for all three authentication systems. FAR achieves 0.01 (commonly accepted FAR [20], [21]) when the ratio between number of enrolling users and number of random users is 1:2. This is proved for the authentication systems with 9 and 12 enrolling users. It could have been proved for the authentication system with 15 enrolling users as well, if there were 30 random training users, as the trend indicates. However, with 15 enrolling users, we could go only up to 25 random training users, because our data set size is 40.

The interesting result that we can observe from this experiment is that 'number of random users in the trained model' acts as a discriminative threshold for FAR, analogous to 'distance threshold' in a distance based biometric matching system (i.e: in a distance based biometric matching system, FAR increases and FRR decreases with the increased distance threshold). Therefore, based on the security and usability trade-off requirement of a particular application, one can select the appropriate ratio between the number of random users involved in the trained model and the number of enrolling users (or the number of imposters) in the authentication system.

3) End-to-end performance: As mentioned before, the second goal of the experiments is to measure the execution times of the main steps of both enrollment and authentication protocols as well as to observe the resource consumption by the IDP-client in the mobile phone during the execution of the authentication protocol. Since all the critical biometric processing steps of the enrollment protocol are executed by the IDP, we measured the execution times of such steps in the IDP software by taking the average execution times over 100 runs of the corresponding functions, as listed in Table V. On the other hand, since all the critical biometric processing steps of the authentication protocol are executed in the mobile phone, we implemented a performance test suite by utilizing Android's intrumented test framework which automates the inter-application interactions, so that we could automatically run the end-to-end authentication protocol (which involves the aforementioned two mobile applications) in the mobile phone for multiple times and take the average execution times. The breakdown of the execution time between the key steps of the authentication protocol is listed in Table V.

Accordingly, the feature extraction step takes the most time as the eigen faces based feature extraction algorithm involves matrix multiplication which is a function known to be slow in Java, as discussed in [22]. Performance benckmarks such as [23] indicate that the running time could be improved by using certain perfromance enhanced libraries for matrix multiplication, which we did not experiment with, in the scope of this work, as our goal is to investigate whether it is feasible to carry out our proposed privacy preserving, user centric and biometrics based authentication approach from a mobile device, and not to achieve the best performance possible.

We compared our ZKPK based biometrics identity verification approach, with the ZKPK based static identity (such as: email, credit card number, social security number etc.) verification scheme in terms of their end-to-end execution times (see Table VI) and resource consumption in the mobile phone during the authentication phase (see Table VII).

TABLE III
VARIATION OF FALSE REJECTION RATES WITH NUMBER OF ENROLLING USERS AND NUMBER OF RANDOM USERS IN THE MODEL

No. of random users	9 enrolled users		12 enrolled users		15 enrolled users	
	FRR	STDV	FRR	STDV	FRR	STDV
9	0.0	0.0	0.027	0.092	0.0	0.0
12	0.111	0.179	0.0	0.0	0.022	0.083
15	0.037	0.104	0.027	0.092	0.022	0.083
18	0.037	0.104	0.027	0.092	0.022	0.083
21	0.074	0.138	0.0	0.0	0.044	0.113
24	0.037	0.104	0.027	0.092	0.044	0.113

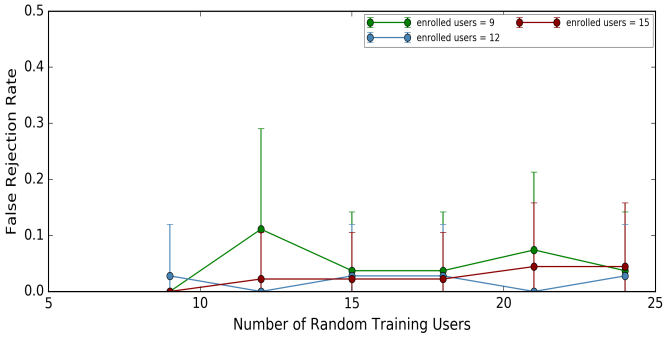

Fig. 4. Variation of False Rejection Rates with number of enrolling users and number of random users in the model

TABLE IV
VARIATION OF FALSE ACCEPTANCE RATES WITH NUMBER OF ENROLLING USERS AND NUMBER OF RANDOM USERS IN THE MODEL

No. of random users	9 enrolled users		12 enrolled users		15 enrolled users	
	FAR	STDV	FAR	STDV	FAR	STDV
9	0.050	0.054	0.053	0.081	0.076	0.087
12	0.027	0.043	0.030	0.066	0.074	0.062
15	0.018	0.028	0.037	0.071	0.063	0.080
18	0.009	0.017	0.027	0.045	0.058	0.069
21	0.013	0.039	0.022	0.051	0.023	0.044
24	0.009	0.017	0.010	0.025	0.023	0.037

TABLE V
BREAKDOWN OF THE EXECUTION TIMES OF BIOMETRICS PROCESSING STEPS OF THE ENROLLMENT AND AUTHENTICATION PHASES

Enrollment Phase	
Steps	Execution Time (ms)
Image reading and feature extraction (of 21*4 images)	4833.0
Parameter selection	7039
Training the classifier	23
Authentication Phase	
Steps	Execution Time (ms)
Reading the image	1.0
Feature extraction	13825.0
Prediction	120.0

TABLE VI
COMPARISON OF THE END-TO-END EXECUTION TIMES OF THE AUTHENTICATION PROTOCOLS

Protocol	Type of the identity verification	
	Static Identity	Biometric Identity
Enrollment	199.0(ms)	12194(ms)
Basic Authentication	2666.0(ms)	15423(ms)

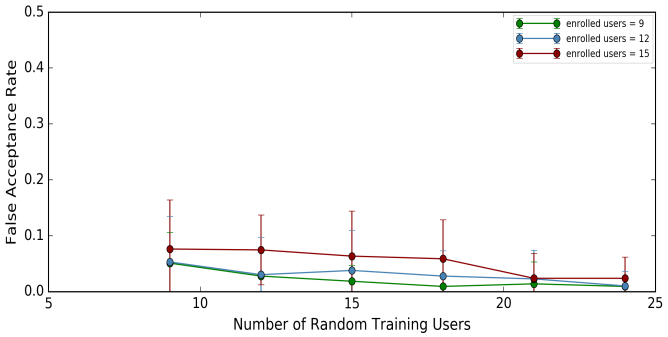

Fig. 5. Variation of False Acceptance Rates with number of enrolling users and number of random users in the model

ZKPK based static identity verification is a well known scheme which was first proposed by Feige et al [24]. It allows users to prove ownership of their static identities without revealing them to the verifiers. We built this functionality also into the IDP-client mobile application for comparison purposes. In the static identity based ZKPK authentication protocol, a user enters a static identity instead of the biometric trait when she/he is prompted to enter credentials by the IDP-client (in step 7 of Figure 8). Then such static identity is cryptographically hashed and used as secret x while the password is cryptographically hashed and used as secret r in commitment C (see Protocol 1 and/or 2a). Therefore, ZKPK based static identity verification does not involve any feature extraction step, classification based prediction step or derivation of multiple secrets from the password. It also does not involve any authentication artifacts stored in the user's mobile phone except the identity token given by the IDP at the end of the enrollment protocol.

According to the experimental data in Table V and Table VI, the additional time taken in the end-to-end execution of the ZKPK based biometric identity verification scheme when compared with the ZKPK based static identity verification scheme, in both enrollment and authentication phases, is due to the time taken by the biometric processing steps - specially the time taken by the feature extraction step, as previously discussed. According to the comparison of the resource consumption in the mobile phone during the authentication protocol, the IDP-client application shows higher performance requirements in the ZKPK based biometric identity verification scheme in terms of CPU usage, memory and storage, while the communication cost is the same for both the schemes. Both schemes exchange the same information between the prover and the verfier during the authentication protocol. The communication cost shown in Table VII furhter breaks down as: 3.589KB of transferring cost and 1.187KB of receiving cost. Transfer cost is higher than the receiving cost because according to Protocol 2b, the prover sends more elements than what verifier does. Storage requirement is higher because in our implementation, we have used eigen faces as the feature extraction mechansim and SVM as the classification technique, which contribute eigen sub space W, the mean image X_{mean} (see Section II-A) and the trained SVM classifier to the artifacts stored in the mobile phone. Note that in the prototype of our system which is used to measure end-to-end performance, we involved 20 random users in the trained classifier. CPU and memory usage is higher due to the expensive operations take place during the biometric processing phase, such as loading the artifacts into the memory and performing multiplication of large matrices.

TABLE VII
COMPARISON OF THE RESOURCE CONSUMPTION IN THE MOBILE PHONE DURING THE BASIC AUTHENTICATION PROTOCOL

Resource	Type of the identity verification	
	Static Identity	Biometric Identity
CPU	2%	47%
Memory	48598.0KB	112448.0
Communication Cost	4.776KB	4.776KB
Storage	1.6MB	7.5MB

In summarizig the information presented in this section, we first experimented with the proposed method for selecting training data to train the classification model which achieves the accepted performance of a biometrics authentication system, based on the AT&T face dataset. According to the paper: 'Guidelines for Best Practices in Biometrics Research' by A. Jain et al [20], accepted FAR is about 1.0% (i.e: 0.01) and according to the talk: 'Biometrics Shor Course' by Andrew S. et al [21], FRR reported at 1% FAR on the FVRT dataset is 25%. Figure 5 shows that the applications for which achieving a low FAR (at the cost of relatively high resource consumption) is a high priority, can achieve the aforementioned FAR by employing a classifier trained with the data obtained from the method proposed in A and an appropriate ratio between the number of random users involved in the trained model and the number of enrolled users in the authentication system. Second, we analyzed the performance of the end-to-end authentication protocol and observed that the only bottleneck is in the feature extraction

algorithm that we employed. Overall the performance proves the feasibility of the proposed user centric and biometrics based authentication approach for mobile phones given that the experiments were carried out in a very low end mobile phone like 'Moto G', and that no optimization technique was applied in the implementation of the matrix multiplication step. It is important to note that the privacy preserving biometrics based authentication solution that we have presented in Section III is very generic so that one can plug-in any other better and optimized algorithms for commitment, feature extraction and classification steps by using the extension points provided in our prototype implementation.

V. Security and Privacy Analysis

In this section we formally define PrivBioMTAuth scheme and analyze its properties.

A. Modelling PrivBioMTAuth

There are three principal entities: User U, IDP and SP. We will also differentiate the human user and the user agent running in the user's device. Let the user's device be denoted by D. Before authentication can take place, user enrolls with the IDP. After the enrollment is completed, user can authenticate with SP using biometrics.

Definition I: We define two procedures of PrivBioMTAuth scheme: Enroll and Auth.

Enroll: is a protocol between U and IDP. If the enrollment is successful, U obtains the IDT and other artifacts, which are stored in D.

Auth: is a protocol between U and SP, where U uses biometrics, password, IDT and D to generate the proofs of authentication for SP to verify

Definition II: PrivBioMTAuth (Enroll, Auth) is a scheme with following properties:

1) *Complete*: A genuine user can authenticate successfully.
2) *Secure*: A computationally bounded adversary has a negligible success probability in breaking the Auth procedure and impersonating a genuine user.
3) *Privacy-preserving*: No sensitive information about the biometrics or the password is leaked from the IDT, D or from the transcripts of *Enroll* and *Auth* protocols. IDP will not learn any information about the user's interactions with SPs, as long as the IDP follows the protocol and does not collude with the SPs. (i.e: IDP is modelled as an honest but curious adversary against preserving user's transaction privacy).

Aforementioned properties cover the goals that we mentioned in Section I. Now we define the building blocks of Enroll and Auth procedures and their properties.

Definition III: BID-Gen is a function f:{biometric, password} $\rightarrow$ BID ($\{0,1\}^{160}$); bound to a user U, with following properties:

1) *Unique*: BID is unique for each U. i.e: Pr[eq(x,y)=1 | x$\leftarrow$ BID-Gen($BIOM_u$, PW_u), y$\leftarrow$BID-Gen($BIOM_{u1}$, PW_{u1})] $\leq \epsilon_1$, for negligibly small ϵ_1, where eq stands for equality, $BIOM$ stands for biometrics and PW stands for password.
2) *Repeatable*: Output of BID-Gen for the same user U, at two different times is the same. i.e: Pr[eq(x,y)=0 | x$\leftarrow$BID-Gen($BIOM_u$, PW_u), y$\leftarrow$BID-Gen($BIOM'_u$, PW_u)] $\leq \epsilon_2$, for negligibly small ϵ_2.
3) *Revocable*: User U can safely replace BID1$\leftarrow$BID-Gen($BIOM_u$, $PW1_u$) with BID2$\leftarrow$BID-Gen($BIOM_u$, $PW2_u$). i.e: BID1=BID2.

We can build such a BID-Gen function using two main building blocks: a classification function (Cl-F) which is defined below and PBKDF2 which was defined in Section II-E.

Definition IV: A classification based learning algorithm is given by two procedures: Train and Predict.

Train: Takes set of training data (Tr-Data), prepared as pairs of {class-label (cl), feature-vector (f)}, as inputs and outputs a classification function (Cl-F): {f} $\rightarrow$ {cl}. i.e: Train: {Tr-Data} $\rightarrow$ {Cl-F}.

Predict: Takes a feature-vector (f) and Cl-F as input and outputs the mapping class-label (cl) of f. i.e: Predict: {f, Cl-F} $\rightarrow$ {cl}. Predict must have two properties: low-FAR and low-FRR.

Low-FAR: Pr[eq(cl1, cl2)=1 | cl1$\leftarrow$ Cl-F(f1), cl2$\leftarrow$Cl-F(f2), f1$\not\approx$f2] $\leq \epsilon_1$, for negligibly small ϵ_1.

Low-FRR: Pr[eq(cl1, cl2)=0 | cl1$\leftarrow$ Cl-F(f), cl2$\leftarrow$Cl-F(f'), f$\approx$f'] $\leq \epsilon_2$, for negligibly small ϵ_2.

We call such a classification function, a (ϵ_1,ϵ_2) Cl-F, where ϵ_1 and ϵ_2 are false acceptance rate (FAR) and false rejection rate (FRR) respectively.

We define two functions: $PBKDF_{s1}$ and $PBKDF_{s2}$ which derives two secrets S1 and S2 used in PrivBioMTAuth scheme. $PBKDF_{s1}$(PW) = PBKDF2(PW)$|_{0-127}$ and $PBKDF_{s2}$(PW) = PBKDF2(PW)$|_{128-288}$. i.e: $PBKDF_{s1}$ and $PBKDF_{s2}$ extracts the first 128 bits and second 160 bits respectively, from the output of PBKDF2 (see Section II-E) on the input of the password and the salt value. We have ignored the salt value for brevity.

Definition V: Therefore, we can model BID-Gen as follows, using three sub functions: Cl-F, $PBKDF_{s1}$ and Concat:$\{(0,1)^{32}, (0,1)^{128}\} \rightarrow \{(0,1)^{160}\}$.

BID-Gen(BIOM, PW):
1. cl $\leftarrow$ Cl-F(BIOM).
2. S1 $\leftarrow$ $PBKDF_{s1}$(PW).
3. BID $\leftarrow$ Concat(cl, S1).
4. return BID.

Accordingly, Enroll procedure (defined in Protocol 1 of Section III) is built using BID-Gen, $PBKDF_{s2}$ and Pedersen Commitment (defined in Section II-C), which cryptgraphically encodes the two secrets: BID and S2. Auth procedure (defined in Protocol 2a of section III) is built using BID-Gen, $PBKDF_{s2}$ and and Zero-Knowledge-Proof-of-Kowledge (defined in Protocol 0 of Section II-D).

B. Analyzing PrivBioMTAuth

In this section, we analyze the properties of PrivBioMTAuth and its building blocks defined in Section V-A. First of all, low-FAR and low-FRR properties of Cl-F (see Definition IV) is empirically proven in Section IV-B, which implies uniqueness and repeatability properties of BID-Gen (see Definition III) respectively. Completeness property of PrivBioMTAuth (see Definition II) scheme is implied by three things: i) repeatability of BID-Gen ii) deterministic property of PBKDF2 iii) completeness property of ZKPK (as mentioned in II-D). We do not list the details of proof of completeness of ZKPK here and refer the reader to e.g. [25].

Proof of security of PrivBioMTAuth (see Definition II) involves showing that breaking PrivBioMTAuth scheme implies that the discrete log problem can be solved. Thus this implies (assuming that discrete log problem is hard) that a computationally bounded adversary has a negligible success probability in breaking the scheme.

Theorem: If $PBKDF_{s1}$ is a random oracle and there exists an adversary A that successfully authenticates with Auth procedure with non-negligible probability, then A contains a knowledge extractor that can solve the discrete log problem with non-negligible probability.

Proof: Assume that such an adversary A exists, and that an adversary B is given a discrete log problem instance: g, p, q, $g^x h^r$ (i.e: pedersen commitment of x and r). B is also given oracle access to Concat function, $PBKDF_{s2}$ and Cl-F trained for a random enrolling user. B sends a randomly chosen challenge e and $g^x h^r$ to A. A is expected to eventually output ZKPK of x and r against challenge e.

To simulate $PBKDF_{s1}$, B creates a set of tuples PSET. B initializes PSET by choosing a random password PW_B and adding (PW_B, $S1_B$) to PSET for randomly chosen $S1_B$ of length 128 bits. B initializes another set of tuples BSET which is initialized by choosing a random biometric feature vector $BIOM_B$, obtaining the output: $cl_B \leftarrow$ Cl-F($BIOM_B$) and adding ($BIOM_B$, cl_B) to BSET. When A queries each of Cl-F and $PBKDF_{s1}$ on a value m, B does the following. If there is a tuple (m, n) already in the corresponding tuple-set, it responds with n. Otherwise; 1) if it is PSET, it chooses a random r', adds (m, r') to PSET and responds to A with r', and 2) if it is BSET, it queries $cl' \leftarrow$Cl-F(m), adds (m, cl') to BSET and responds to A with cl'. When A queries Concat and $PBKDF_{s2}$ with (cl_A, $S1_A$) and PW_A respectively, B does the following. If $cl_A = cl_B$ and $S1_A = S1_B$, B outputs FAIL. Otherwise, B queries Concat and $PBKDF_{s2}$ with (cl_A, $S1_A$) and PW_A respectively, receives BID_A and $S2_A$ and sends them to A.

It is straight forward to see that if B does not output FAIL, then the above view is the same as the view when B is engaging in the protocol. In the following we show that: (i) B outputs FAIL with negligible probability and (ii) if B does not output FAIL and A succeeds with non-negligible probability, then B can use A to obtain x and r.

(i) B outputs FAIL only when A asks for Concat and $PBKDF_{s2}$ responses for inputs: (cl_A, $S1_A$) and PW_A and $cl_A = cl_B$ and $S1_A = S1_B$. In this FAIL scenario, two things can happen: (a) A queries Cl-F on $BIOM_B$ and $PBKDF_{s1}$ on PW_B (b) A does not query Cl-F on $BIOM_B$ and $PBKDF_{s1}$ on PW_B. Case (a) implies that A knows $BIOM_B$ and PW_B, which is an event with negligible probability, as A does not know B's biometrics and password. Case (b) implies that A randomly guesses cl_B and $S1_B$ which is negligible because cl is drawn from a space of size: 2^{32} and $S1$ is drawn from a space of size 2^{128}.

(ii) If B does not output FAIL and A can create a ZKPK of discrete logarithm of $g^x h^r$, then by properties of ZKPK, there must be a knowledge extractor for A that produces x and r. B uses this knowledge extractor to solve discrete log problem. If A succeeds, then so does B. However, assuming discrete log problem is hard, an adversary A does not exist.

Proof of privacy of PrivBioMTAuth (see Definition II) involves showing that no sensitive information about the biometrics or the password is leaked from: i) IDT ii) D iii) transcripts of *Enroll* and *Auth* protocols, and iv) no information about the interaction between U and SP is learned by the IDP. Case (i) directly follows from the perfectly hiding property of Pedersen Commitment scheme [8]. Case (ii) is ideally achieved by the use of Trusted Execution Environment (TEE) [26] of modern mobile devices for storing the artifacts given by the IDP to the user at the end of Enroll procedure. However, since the mobile device that we used for experiments did not allow developper access to TEE, we followed a defense-in-depth approach to secure the artifacts stored in D. First, the artifacts are stored in the IDP-client application's internal storage, which cannot be accessed by any other application, based on the application sandboxing mechanism provided by the Android kernel. Second, the artifacts are encrypted using a key stored in a keystore which also resides in the application's internal storage. Third, the keystore is secured with a secret ($S3$) which is derived from the user's password. Therefore, an attacker needs to break all the defense mechanisms in order to get hold of the artifacts and try to infer any sensitive information. We would like to emphasize that even if an attacker is able to break this defense-in-depth protection mechansims and get hold of the artifacts stored in D, the attacker can not break the security of PrivBioMTAuth scheme as biometrics and the password of the user is not stored in the device. Case (iii) directly follows from case i) that IDT in the protocol transcripts relveals nothing, from the zero-knowledge property of ZKPK scheme used in the Auth protocol such that u, v in the *Auth* protocol transcirpt reveals nothing and from the fact that no other sensitive information is exchanged in plaintext during the *Enroll* and *Auth* protocols. Case (iv) follows from the fact that we have adopted a user-centric architecture in the authentication protocol which avoids any user-authentiation attempt from going through the IDP and the assumption of a honest but curious IDP which does not collude with SPs.

Accordingly, we can conclude that PrivBioMTAuth(Enroll, Auth) scheme satisfies the three properties mentioned in Definition II, and hence it is complete, secure and privacy

preserving.

VI. Related Work

Reliably extracting a reproducible random string from noisy biometric data has been widely investigated in the area of biometric cryptosystems [27]. Fuzzy extractors [28], constructed using a secure sketch and a strong extractor, are a well known primitive used in such biometric cryptosystems. A fuzzy extractor extracts a random value R and helper data P from an original biometric feature set W and enables to reconstruct R in an error-tolerant way, given P and a close enough biometric feature set W'. P could be made public without much entropy loss in R. Secure sketch (i.e: P) is based on error-correcting codes and it is used to reconstruct the original biometric image from a noisy biometric reading. A strong extractor is a universal hash function used to extract the secret (i.e: R) from the biometric feature set. Accordingly, R derived from fuzzy extractors (which is analogous to BID in our scheme) is unique and repeatable, which are two of the three main properties of BID, as discussed in Section I. However, there are issues when using fuzzy extractors to address the third critical property of BID, namely revocability followed by renewability (see Section I). The reason is that in fuzzy extractors the release of multiple sketches associated with the same biometric features poses security and privacy issues due to the unavoidable information leakage, and hence, as already proven in [29], [30], fuzzy extractors cannot be securely reused even in the presence of very weak adversaries. Addressing such security and privacy issues requires additional measures such as using a second authentication factor.

Many proposed privacy preserving biometrics systems have focused on identification [31], [32], [33] in which a stored user profile is retrieved for a person whose biometric input matches any of the stored biometric templates in the server. The schemes which aim to address the authentication problem [34], [35] have formulated the problem as returning a single bit indicating whether or not there is a match. In contrast, we formulate the authentication problem as verifying if a user is who she/he claims to be. Previous work that has focused on biometrics based authentication for mobile devices has primarily focused on continuous authentication [36]. Continuous authentication aims at authenticating the user to the device, which is different from the biometrics based remote authentication problem that we aim to address, in which the user is directly authenticated to the remote SP using biometrics. Further, such previous work on continuous biometrics based authentication in mobile phones has focused on performance of the matching techniques rather than on privacy and security.

Preliminary approaches have been developed to derive a secret from the user's biometrics to be used in ZKPK based biometric identity verification [37] and [38]. However, they have some major drawbacks. For example, the approach presented in [37] is not robust in terms of FRR and FAR, as it is disussed in detail in [38], since it uses Singular Vector Dicomposition (SVD) based feature extraction mechanism and multi-label based prediction output. On the other hand, the approach proposed in [38] uses perceptual hashing which is a general purpose feature extraction mechanism and does not perform very well with different biometric traits, which caused it to use error correcting code to improve accuracy of prediction. In contrast, in this work we recommend to use a feature extraction algorithm specific to a particular biometric trait. Such previous work also do not clearly define how the training data is selected to train the classifier, which is a critical aspect that not only affects the robustneess of the final classifer, but also usability and security of the entire authentication system.

Accordingly, existing approaches are not suitable for our setting because of one or more of the following reasons: (i) they address identification and not authentication [31], [32], [33]; (ii) they consider a setting in which a user stores biometric template at each SP that the user wants to authenticate to, using biometrics and does not consider involving an IDP for avoiding the replication of user's sensitive biometric information at multiple SPs and using a user-centric identity management architecture; (iii) they only support authentication local to the phone and not authentication to a remote server [36], [39]. We did not find any previous approach that presents a complete end-to-end, privacy preserving, user centric and biometrics based remote authentication solution for mobile phones and which also addresses theoretical aspects such as security and privacy preserving techniques and practical aspects such as implementation challenges and end-to-end performance evaluation.

The main concept behind the extended authentication protocol proposed in this paper to address man-in-the-middle impersonation attacks on ZKPK identity verification is inspired by M-PIN authentication protocol [40]. M-PIN proposes a static identity verification scheme based on elliptic curve and pairing based cryptography. Although M-PIN claims to be a ZKPK protocol, it is actually not zero knowledge, because the trusted party shares two secrets with the prover and the verifier so that the verifier can estimate how much the difference between a wrong PIN and the actual secret PIN is.

VII. Conclusion and Future Work

We have presented an authentication scheme in which users can authenticate to remote services using their biometrics from their mobile phones. Our scheme mainly focuses on privacy preserving and user-centric authentication in order to overcome the drawbacks of the existing biometrics based authentication solutions. At the same time our scheme also provides the verifier with a strong assurance about the ownership of the biometric identity of the prover. Two significant contributions in our solutions are: i) a methodology to securely derive a unique, repeatable and revocable biometric identifier in the user's mobile phone, which is used as the identity secret in the commitment scheme and the ZKPK authentication ii) an extended authentication protocol which mitigates the threat of man-in-the-middle impersonation attack affecting the traditional ZKPK based identity verification and which also helps to lay a foundation for securing any communication that

takes place between the prover and the verifier following the authentication.

We have performed experimental evaluations of our scheme by implementing a prototype of the scheme, which proves the feasibility of the scheme in terms of robustness in authentication-decision making and end-to-end execution performance. While there is room for improvement in terms of performance optimization, we believe that this work will lay a foundation for the researchers to investigate privacy preserving and user-centric biometrics authentication solutions for mobile phones. In our future work, we plan to investigate authentication functions that do not involve classification techniques requiring the training features to be encoded in the classifier. In this way, the authentications scheme can be made independent of the platform security mechanisms that we have utilized in order to secure the classifier stored in the mobile phone.

REFERENCES

[1] N. Cappella, "HSBC announces biometric banking with voice and fingerprints," Feb. 2016, Accessed: 10-Jan-2017. [Online]. Available: https://thestack.com/world/2016/02/19/hsbc-voice-biometric-online-banking/

[2] A. MacGregor, "Security in rich internet applications," Feb. 2016, Accessed: 12-Jan-2017. [Online]. Available: https://thestack.com/iot/2016/02/22/mastercard-rolls-out-selfie-verification-for-mobile-payments/

[3] ——, "Amazon wants to replace passwords with selfies and videos," 2016, Accessed: 15-Jan-2017. [Online]. Available: https://thestack.com/security/2016/03/15/amazon-wants-to-replace-passwords-with-selfies-and-videos/

[4] V. M. Patel, N. K. Ratha, and R. Chellappa, "Cancelable biometrics: A review," in *IEEE Signal Processing Magazine*, vol. 5, August 2015.

[5] daon.com, "IdentityX Platform," Accessed: 14-Nov-2017. [Online]. Available: https://www.daon.com/products/identityx-platform

[6] F. Paci, "Veryidx - a digital identity management system for pervasive computing environments," in *Proceedings of 6th IFIP*, 2008.

[7] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *CRYPTO 1986*, May 2015.

[8] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *CRYPTO'91*.

[9] Y. Desmedt, C. Goutier, and S. Bengio, "Special uses and abuses of the fiat-shamir passport protocol," in *CRYPTO '87*, August 1987.

[10] wihoho, "Implement face recognition using PCA, LDA and LPP," Oct. 2016, Accessed: 10-Nov-2016. [Online]. Available: https://github.com/wihoho/FaceRecognition

[11] M. Turk and A. Pentland, "Eigen faces for recognition," in *Journal of Cognitive Neuroscience*, vol. 3, August 1991.

[12] K. Kostiainen, J. Ekberg, and et al., "On-board credentials with open provisioning," in *Proceedings of ASIACCS'09*, 2009.

[13] J. Solis and G. Tsudik., "Simple and flexible revocation checking with privacy." in *In Proceedings of the 6th international conference on Privacy Enhancing Technologies, PET'06.*, 2006.

[14] M. Narasimha, J. Solis, and G. Tsudik., "Privacy-preserving revocation checking." in *Int. J. Inf. Secur.*, 2009.

[15] R. Peeters and A. Pashalidis., "Privacy-friendly checking of remote token blacklists." in *3rd Policies and Research in Identity Management (IDMAN)*, 2013.

[16] E. D. Cristofaro and G. Tsudik., "Practical private set intersection protocols with linear complexity." in *In 14th Financial Cryptography and Data Security, FC 2010)*, 2010.

[17] GSMARENA, "Motorola moto g - full phone specifications," Accessed: 14-Nov-2017. [Online]. Available: http://www.gsmarena.com/motorola_moto_g-5831.php

[18] AT&T, "The database of faces," 2002, Accessed: 14-Nov-2017. [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[19] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[20] A. Jain, B. Klare, and et al., "Guidelines for best practices in biometric research," in *International Conference on Biometrics*, May 2015.

[21] A. Senior and N. Ratha, "Biometrics short course." [Online]. Available: http://www.research.ibm.com/people/a/aws/documents/CVPR-BiometricsShortCourse-Part2.pdf

[22] Stack Overflow, "Performance of java matrix math libraries?" 2015, Accessed: 14-Nov-2017. [Online]. Available: http://stackoverflow.com/questions/529457/performance-of-java-matrix-math-libraries

[23] Java-Matrix-Benchmark, "Runtime : Intel core i7-2600 processor," Accessed: 14-Nov-2017. [Online]. Available: http://lessthanoptimal.github.io/Java-Matrix-Benchmark/runtime/2013_10_Corei7v2600/

[24] U. Feige and A. S. A. Fiat, "Zero-knowledge proofs of identity," *Journal of Cryptology*, 1988.

[25] D. Chaum, J. Evertse, and J. Graaf, "An improved protocol for demonstrating possession of discrete logarithms and some generalizations." in *EUROCRYPT*, 1987.

[26] N. Asokan, J. Ekberg, and K. Kostiainen, "The untapped potential of trusted execution environments on mobile devices," in *Proceedings of Financial Cryptography and Data Security*, April 2013, pp. 293–294.

[27] C. Rathgeb and A. Uhi, "A survey on biometric cryptosystems and cacelable biometrics," *EURASIP Journal on Information Security*, 2011.

[28] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *EUROCRYPT*, 2004.

[29] K.Simoens, P. Tuyls, and B. Preneel, "Privacy weaknesses in biometric sketches," in *IEEE Symp. Security and Privacy*, 2009.

[30] M. Blanton and M. Aliasgari, "Analysis of reusability of secure sketches and fuzzy extractors," in *IEEE Trans. Inf. Forensics and Security*, 2013.

[31] Z. Erkin, M. Franz, and et al., "Privacy preserving face recognition," in *Privacy Enhancing Technologies*, 2009.

[32] Y. Huang, L. Malka, and et al., "Efficient privacy-preserving biometric identification," in *NDSS'2011*. IEEE, February 2011.

[33] M. Blanton and P. Gasti, "Secure and efficient protocols for iris and fingerprint identification," in *Computer Security - ESORICS*, 2011.

[34] H. Chun, Y. Elmehdwi, and et al., "Outsourceable two-party privacy-preserving biometric authentication," in *ASIA CCS*. ACM, June 2014.

[35] M. Barni, T. Bianchi, and D. C. et al., "Privacy-preserving fingercode authentication," in *12th ACM workshop on Multimedia and security*, 2010.

[36] D. Crouse, H. Han, and et al., "Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data." in *International Conference on Biometrics*, May 2015.

[37] A. Bhargav-Spantzel, A. C. Squicciarini, and et al., "Biometrics-based identifiers for digital identity management," in *IDtrust '10*. ACM, April 2010.

[38] H. Gunasinghe and E. Bertino, "Privacy Preserving Biometrics-Based and User Centric Authentication Protocol," in *Network and System Security - 8th International Conference, NSS 2014*, 2014, pp. 15–17.

[39] M. Gofman and S. Mitra, "Multimodal biometrics for enhanced mobile device security," in *International Conference on Biometrics*, April 2016.

[40] M. Scott, "M-pin: A multi-factor zero knowledge authentication protocol," Accessed: 14-Nov-2017. [Online]. Available: http://cdn2.hubspot.net/hubfs/230906/miracl/white_papers/MIRACL_M-Pin_ZeroFactor.pdf?t=1467068304878

[41] MIT, "Face Databases," Accessed: 14-Nov-2017. [Online]. Available: http://web.mit.edu/emeyers/www/face_databases.html

[42] A. Escalante and L. Wiskott, "Gender and age estimation from synthetic face images," in *13th IPMU*, July 2010.

[43] Jersy, "Jersy - restful web services in java." Accessed: 14-Nov-2017. [Online]. Available: https://jersey.java.net/

[44] E. Chin, A. P. Felt, K. Greenwood, and D. Wagner, "Analyzing inter-application communication in android," in *MobiSys*, June 2011.

APPENDIX

A. Selecting training data to train the classifier

We experimented with the following three methods for selecting the training data to train the classifier for a specific user enrolling in the authentication system:

(i) Training data consisting of the biometric features of a set of random users.

With this method, the classification model is trained

with the biometric features of some random set of users who are outside the authentication system (eg. publicly avaialable biometric dataset [41] or a synthetic biometric images [42]), so that they do not overlap with the space of potential *imposters*. We assume that a given enrolling user's biometric features will be associated with the class label that belongs to the biometric features in the training data that has the closest match with the enrolling user's biometric features and that the same class label will be predicted by the trained model at both enrollment and authentication. We also assume that an *imposter's* biometric features will be matched to a different class label than that of the given enrolling user.

(ii) Training data consisting of the biometric features of a set of random users and the ones of the current enrolling user.
With this method, we assume that the class label associated with a particular enrolling user's biometric features in the training data will be output by the trained model at the authentication attempts of the same enrolling user and that an *imposter's* biometric features will be matched with any of the random users' biometric features used to train the model.

(iii) Training data consisting of the biometric features of all the enrolling users of the authentication system.
With this method, we assume that the class label associated with a given enrolling user's training features is output at the authentication attempts of the same enrolling user and an *imporster's* biometric features is matched with the class label associated with her/his training features.

We have carried out experiments to emphirically evaluate the performance of the classifiers trained with the data obtained from the above three methods in terms of False Rejection Rate (FRR) and False Acceptance Rate (FAR). The detailed experiment results are presented in section B. Table VIII reports high level comparison of those three methods in terms of security and usability.

Note that in the context of the multi-class classification model that we use, FRR is the rate at which the trained classification model predicts a class label different than the one associated with the genuine user's biometric features at enrollment, when the genuine user's biometric features is given as input to the model at authentication. Usability decreases as FRR increases because the genuine user finds difficulties in authenticating. FAR is the rate at which the trained classification model outputs the class label that is associated with the genuine user's biometric features at enrollment, when an *imposter's* biometric features is given as input to the model at authentication. An authentication application becomes less secure as FAR increases because the probability that an *imposter* authenticates as the genuine user increases.

Method 1 incurs the minimum privacy threat for the authentication system if the classifier in a user's mobile phone is compromised as no sensitive biometric features are included in the classifier. However, it is not very secure as FAR is high and is not usable as FRR is high (see section B for experimental results). In contrast, method 3 incurs the highest privacy threat on the authentication system if the classifier in a user's mobile device is compromised as the biometric features of all the enrolled users are included in the trained classifier that is given to a particular enrolled user. However, it has a low FAR and is more usable as it has a low FRR than other two methods (see section B for experimental results). One of the goals of our approach is to avoid storing biometric features of enrolled users even at the IDP, after the enrollment phase is over. However, with method 3, the IDP has to store the biometric features of the users who have enrolled in the system so far, in order to train the classifier for the future enrolling users, which is not desirable in terms of security and privacy of biometric features.

TABLE VIII
SECURITY AND USABILITY ASPECTS OF THE POTENTIAL METHODS FOR SELECTING THE TRAINING DATA TO TRAIN THE CLASSIFIER.

Method	Security	Usability
1	-Biometric features of any enrolled user in the authentication system is not included in the classifier. -FAR is high.	- FRR is high.
2	- A given enrolling user's biometric features is the only sensitive informatio stored in the classifier. - FAR varies with the number of random users in the training data.	- FRR varies with the number of random users in the training data.
3	- Biometric features of all the enrolled users in the authentication system is included in the classifier. - FAR is minimum and stays almost the same. - Biometric features of all enrolling users need to be stored at the IDP	- FRR is minimum and stays almost the same.

On the other hand, method 2 achieves a balanced trade-off between security and usability. The only sensitive biometric features included in the classifier is the biometric features of the currently enrolling user. Therefore, in case of a compromise of the trained classifier, the biometric features of the other users enrolled in the authentication system will not be exposed. This method also allows us to achieve the desired FAR and FRR by varying the number of random users whose data is used to train the model (see section B for experimental details). Therefore, out of the three methods, we used method 2 in order to select the training data to train the machine learning model which is then used to derive the BID from the user's biometrics at the authentication times.

B. Experimental evaluation of the trained classifiers

In what follows we discuss the evaluation of the classifiers trained using the training data obtained from the three potential methods described in section A, in terms of FRR and FAR.

I) *Experimental details of method 1*: The face dataset was divided into two sets: i) training set which includes the images of 25 users is considered as the set of random users ii) testing set which includes images of 15 users is considered as the set

of enrolling users in the authentication system. A classification model was trained using the images of the 25 random users and it was given to each of the 15 enrolling users. Out of the 10 images of each enrolling user, 6 images were used to assign a class label to a particular user via majority votes (i.e: the class label that was predicted the highest number of times out of the predictions for the 6 images, is selected as the class label representing that particular user's biometric features) at enrollment. The remaining 4 images of each enrolling user were used for measuring the FRR and FAR of predictions at authentication.

II) *Experimental details of method 2*: The partition of the dataset into training and testing data is the same as in method 1. For each of the 15 enrolling users, 20 random users were selected from the training set and a classification model was trained using 6 images of each of those 20 random users and 6 images of the current enrolling user. The class label assigned to each enrolling user's feature vectors in the training data was expected to be predicted at authentication. The predictions on the remaining 4 images of each enrolling user were evaluated for FRR and FAR.

III) *Experimental details of method 3*: In this method, all the users in the dataset were considered as enrolling users and 6 images of each user were used to train a classification model which was given to each user. The class label assigned to each user's feature vectors in the training data was expected to be predicted at authentication, which was tested using the remaining 4 images of each user.

In summary, in method 1, the training users and enrolling users (hence the *imposters*) of the authentication system are totally disjoint. In method 2, it is the same except that the biometric features of an enrolling user is also included in training her/his classifier. In method 3, training users and enrolling users (hence the *imposters*) overlap.

Evaluation of FRR: Table IX and Figure 6 report mean FRR and standard deviation (STDV) for the three methods over four random rounds of the experiment. In measuring FRR for one round of the experiment for method 1 and method 2, false rejections made by the trained classifier on 3 images out of the 4 testing images of each enrolling user were counted and the mean FRR was computed over all 15 enrolling users, along with the corresponding standard deviation. In reporting FRR for method 3, 15 users were selected at random (because all 40 users were considered as enrolling users in that case) as the enrolling users and the computation was done for each round in the same way mentioned above. As discussed in section A, method 1 shows the worst performance and methods 2 and 3 show better performance in terms of FRR which is related with the usability of the end application.

TABLE X
VARIATION OF FALSE ACCEPTANCE RATES WITH NUMBER OF IMPOSTERS

Method	3 Imposters		6 Imposters		9 Imposters		12 Imposters	
	FAR	STDV	FAR	STDV	FAR	STDV	FAR	STDV
Method 1	0.322	0.171	0.341	0.102	0.348	0.057	0.352	0.053
Method 2	0.0	0.0	0.022	0.036	0.053	0.065	0.056	0.044
Method 3	0.005	0.020	0.008	0.022	0.007	0.015	0.002	0.010

TABLE IX
FALSE REJECTION RATES OVER FOUR TRIALS

Method	Trial 1		Trial 2		Trial 3		Trial 4	
	FRR	STDV	FRR	STDV	FRR	STDV	FRR	STDV
Method 1	0.556	0.233	0.644	0.257	0.622	0.239	0.622	0.239
Method 2	0.022	0.083	0.044	0.113	0.022	0.083	0.044	0.113
Method 3	0.05	0.1	0.033	0.084	0.033	0.084	0.033	0.084

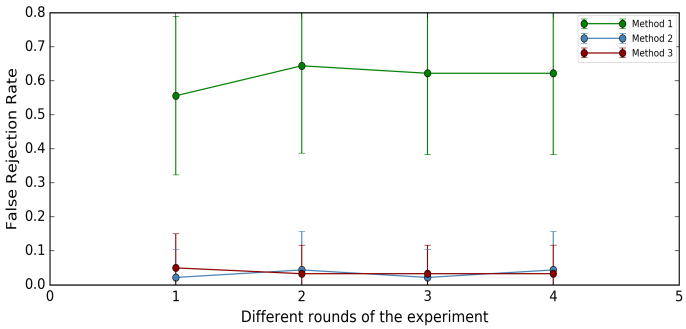

Fig. 6. False Rejection Rates over four trials

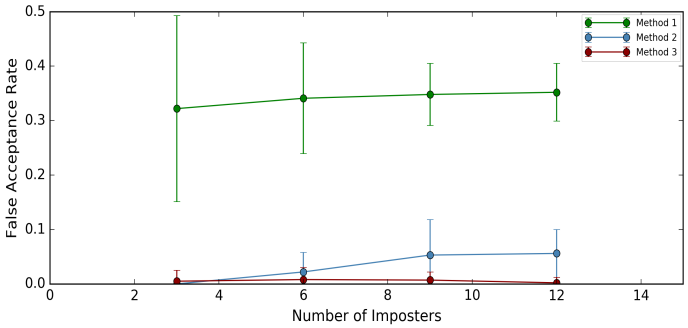

Fig. 7. Variation of False Acceptance Rates with number of imposters

Evaluation of FAR: Table X and Figure 7 report mean FAR and STDV for the three methods over four rounds of the experiment, which was carried out by varying the number of imposters as 3, 6, 9 and 12. In measuring FAR for one round of the experiment, for each of the 15 enrolling users, false acceptances made by the trained classifier on the 4 test images of each of the *imposters* were counted and mean FAR was computed over all 15 enrolling users, along with the corresponding STDV. As discussed in section A, method 1 shows the worst performance and method 3 shows the best performance in terms of FAR which is related with the security of the end application. Method 2 reports increasing FAR for increasing values in the number of imposters and with a constant number of random users (20) involved in training the classifier. As discussed in section A, we decided to use method 2 for selecting the training data to train the classifier, as it gives the best trade-off between security, privacy and usability of the overall authentication system.

C. Impersonation attack

Protocol A lists the steps of the man-in-the middle type impersonation attack (also known as Mafia attack [9]), based

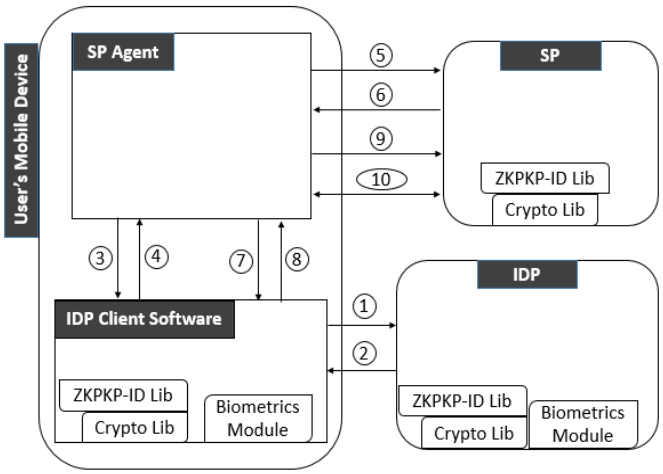

Fig. 8. Overall Architecture of the Authentication Solution

on the steps of the standard ZKPK based identity verification (details of the biometric identity is removed for brevity).

D. Details of the modules

We have developped three self-contained modules which encapsulate different building blocks of the protocols, which are re-used accross multiple components of the solution as illustrated in Figure 8. They are: i) Crypto Lib: this implements the Pedersen Commitment algorithm and the ZKPK scheme based on the Pedersen Commitment ii) ZKPK-ID Lib: this implements the layer that encapsulates the identity verification scheme on top of ZKPK iii) Biometrics module: this implements all the steps related to biometrics processing in both enrollment and authentication protocols. These components make it easy for the developpers to adopt the proposed scheme in their systems and also serve as the extension points in our solution where one could plug-in other (better) algorithms for commitment, feature extraction and classification steps.

E. Implementation Details

The SP and the IDP software were implemented as RESTful web services using the Jersy RESTful framework [43]. The three modules mentioned in section D were implemented in Java 1.7. The IDP-client and the SP-client were implemented as two separate applications using the Android framework. Android's security model handles the two applications as mutually distrusting entities [44]. Data stored in one application's internal storage is isolated and cannot be accessed by other applications. Therefore, we utilized the IDP-client's internal storage to store the artifacts handed over by the IDP at the end of the enrollment protocol. We utilized the Android's inter-application communication mechansim to implement the steps 3, 4 and 7, 8 of the communication flow, shown in Figure 8.

Protocol A Mafia Attack on the Basic ZKPK Authentication Protocol

User	malicious SP	genuine SP
1. authentication request with IDT_{user} and d $\longrightarrow$		
	2. authentication request with IDT_{user} and d $\longrightarrow$	
	$\longleftarrow$ 3. challenge: e	
$\longleftarrow$ 4. challenge: e		
5. u and v $\longrightarrow$		
$\longleftarrow$ 6. signal as authentication failure		
	7. u and v $\longrightarrow$	
	$\longleftarrow$ 8. authentication success	
	$\longleftrightarrow$ 9. transactions carried out via impersonation	