

Biometrics-Based Identifiers for Digital Identity Management

Abhilasha
Bhargav-Spantzel
Intel Corporation
2191 Laurelwood Avenue
Santa Clara, CA 95054
abhilasha.bhargav-
spantzel@intel.com

Anna Squicciarini
College of Information
Sciences and Technology
Pennsylvania State University
University Park, PA
16802-6823
asquicciarini@ist.psu.edu

Elisa Bertino
Department of Computer
Science
CERIAS
Purdue University
West Lafayette, IN 47906
bertino@cs.purdue.edu

Xiangwei Kong
Information Security Research
Center
Dalian University of
Technology
Liaoning Province 116023
kongxw@dlut.edu.cn

Weike Zhang
Patent Examination
Collaboration Center 33
No. 18 South Fourth
Street, Zhongguan Cun
Haidian District, Beijing
100190
zhangweike@sipo.gov.cn

ABSTRACT

We present algorithms to reliably generate biometric identifiers from a user's biometric image which in turn is used for identity verification possibly in conjunction with cryptographic keys. The biometric identifier generation algorithms employ image hashing functions using singular value decomposition and support vector classification techniques. Our algorithms capture generic biometric features that ensure unique and repeatable biometric identifiers. We provide an empirical evaluation of our techniques using 2569 images of 488 different individuals for three types of biometric images; namely fingerprint, iris and face. Based on the biometric type and the classification models, as a result of the empirical evaluation we can generate biometric identifiers ranging from 64 bits up to 214 bits. We provide an example use of the biometric identifiers in privacy preserving multi-factor identity verification based on zero knowledge proofs. Therefore several identity verification factors, including various traditional identity attributes, can be used in conjunction with one or more biometrics of the individual to provide strong identity verification. We also ensure security and privacy of the biometric data. More specifically, we analyze several attack scenarios. We assure privacy of the biometric using the one-way hashing property, in that no information about the original biometric image is revealed from the biometric identifier.

Categories and Subject Descriptors

K.6.5 [Management of Computing and Information Systems]: Security and protection; E.3 [Data Encryption]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDtrust '10, April 13-15, 2010, Gaithersburg, MD
Copyright ©2010 ACM ISBN 978-1-60558-895-7/10/04... \$10.00

General Terms

Algorithms, Security, Experimentation, Human Factors

Keywords

Security, Privacy, Biometrics, Multi-factor Authentication, Identity, Cryptography

1. INTRODUCTION

To support online activities, such as commerce, healthcare, entertainment and scientific collaboration, it is crucial to be able to verify and protect the digital identity of the individuals involved. Misuse of identity information can result in identity theft, that is, the act of impersonating another's identity by presenting stolen identifiers or proofs of identities. Identity theft has been receiving increasing attention because of its high financial and social costs. An approach that can help in protecting from identity theft is the privacy-preserving multi-factor verification of identity¹. Such a verification requires an individual to prove his/her identity by proving the knowledge of several *identity attributes* (also called *identifiers*). When talking about identifiers, we distinguish between *weak* and *strong identifiers*. A strong identifier uniquely identifies an individual in a population, whereas a weak identifier can be applied to many individuals in a population. The number and types of strong identifiers used in verification should not be fixed a-priori and each party interested in verifying the identity of an individual should be able to require any combination of such identifiers [3]. Biometric data represent an important class of identity attributes. To fully realize their potential, identity verification protocols should be able to support the use of biometric data in combination with other digital identifiers, such as a social security number (SSN) or a credit card number (CCN). The privacy of the biometric data and other sensitive identifiers should, however, be protected to mitigate attacks

¹Effective solutions to protect from identity theft require a combination of technical and non-technical measures. Our approach represents one such measure which if used alone, however, may not be sufficient to address all possible threats to the security and privacy of identity information.

such as identity theft. By privacy of the biometric data we mean that minimal information about the biometric is revealed during the biometric verification process, and that this information cannot be reused in contexts outside a given biometric verification.

The use of biometric data in the context of identity attribute verification poses several non trivial challenges because of the inherent features of the biometric data. In general, two subsequent readings of a given biometrics do not result in exactly the same biometric template². Therefore the matching against the stored template is probabilistic. Storing biometric templates in repositories along with other personally identifiable information introduces security and privacy risks [16]. Those databases can be vulnerable to attacks by insiders or external adversaries and may be searched or used for purposes other than the intended one. If the stored biometric templates of an individual are compromised, there could be severe consequences for the individual because of the lack of revocation mechanisms for biometric templates. To overcome the shortcomings of server-based storage and matching, several efforts have been devoted to the development of techniques based on client side matching [26, 27]. Such an approach is convenient as it is relatively simple and cheap to build biometric verification systems supporting biometric storage at the client end able to support local matching. Nevertheless, systems of this type are not secure if the client device is compromised; therefore additional security mechanisms are needed.

Client side verification systems has lead to research on key generation mechanisms that use biometrics [50, 48, 15, 26, 27, 58, 38]. A biometric key (BK for brevity) is never stored at any location and the key generation mechanisms should not allow the re-generation of the BK without the individuals’ real biometrics. Note that under those approaches the biometric template is stored; therefore the verification does not involve biometric matching and instead uses the BK. Current techniques, however, are not sufficient because of several unresolved challenges concerning BK generation [35]. In particular, most BK generation approaches [24] do not differentiate between the cryptographic keys, used in the BK generation process, and the specific information retrieved from the actual biometrics. For example in [24] the BK is a repeatable string derived from a user biometrics. The final BK is essentially a pre-defined cryptographic key which can only be derived from information stored by the user and the users biometric information. As such the BK is never stored and cannot be derived without the users biometric information. Other approaches map biometric data into a unique and repeatable binary string [50, 48, 15, 26, 27, 58, 38]. Subsequently, the binary string would be mapped to an encryption key known as the BK by referring to a look-up table. In this work we focus on the repeatable binary string, referred to as the biometric identifier (BID), that is derived from the biometrics.

The goal of this paper is to identify the biometric information necessary and sufficient to generate a BID, which can in turn be used to generate a BK or simply as conventional strong identifiers such as SSN or CCN. To be used as strong identifiers, BIDs need to satisfy two key properties, namely uniqueness and repeatability. Uniqueness of BID ensures that two different individuals do not generate the same BID. If each individual is considered as a class in a given classifier model [22], then for uniqueness property to hold, the BIDs should have large inter-class variation. Repeatability of BID refers to the ability by an individual to re-generate his own

²The digital representation of a biometric is referred to as *biometric template*.

BID (small intra-class variation). Another main challenge is to ensure the security and privacy of the biometric data. In particular, it should not be possible to re-create the BID without the original biometrics and the final BID should not leak information about the original biometrics. There are additional challenges with respect to the protection of the BID from brute force attacks conducted by exploiting meta-data stored at the client. As such several well-known solutions to the problem of BK generation have shown to be vulnerable to this threat [35].

We develop an approach that does not need to use specific features of the biometrics. We in fact use generic properties of biometric images that are shown to be suitable for multimodal biometric systems [45]. Multimodal biometric systems utilize more than one physiological or behavioral characteristic for enrollment and verification. This is an original contribution of our work as most of today’s approaches are designed for a specific biometrics and cannot be trivially generalized to other biometrics. Additionally in the current approach, we depend on cryptographic keys in combination with the biometric data to preserve the privacy of the biometric during biometric verification.

Our Approach. The method for generating BIDs from biometric measurements is characterized by two phases [38]. During the first phase the biometric features are analyzed and used to compute a bit string representing these features. Such bit string should have uniqueness and repeatability properties. The bit string is then used in the second phase to generate a unique BID with the help of some meta-data. If two instances of the bit strings are sufficiently similar, then the BID generated is the same.

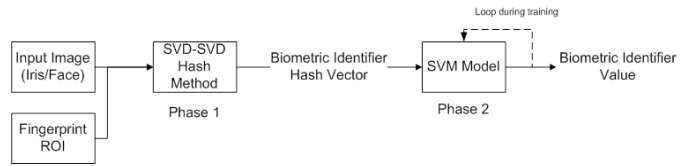


Figure 1: Two main phases of the biometric key generation.

In our approach, in Phase 1, a biometric hash vector is generated. Such biometric hash vector is a bit string which represents the biometrics and is obtained from the biometrics through an image hashing algorithm based on Singular Value Decomposition (SVD) (see Figure 1). In Phase 2, a classifier model based on Support Vector Machines (SVM) is used to classify and rank the resulting biometric hash vector. More specifically, the resulting biometric hash vector is classified to obtain a combination of classes which represent the user’s unique and repeatable BID. The meta-data needed to execute Phase 1 and 2 consists of the classifier model and the pseudorandom secrets involved in the hashing algorithm.

The final BID generated at the end of Phase 2 is used for multi-factor identity verification. Identity verification based on the use of BIDs can be executed according to different strategies. For example the BID can be used as a password or as an attribute embedded in a digital certificate. In our approach we focus on the use of BIDs in the context of a privacy-preserving multi-factor cryptographic protocols for identity verification [3]. More specifically such protocol is based on the notion of *proof of identity* which consists of a cryptographic token bound to an individual, versus the actual value of the individuals’ identity attribute. A proof is created so that only the individual to whom the proof is bound can properly

use it. Proofs of identity attributes are built using zero knowledge proof of knowledge (ZKPK for brevity) techniques [6, 18]. Efficient mechanisms have been developed to prove the knowledge of multiple strong identifiers stored as cryptographic commitments using aggregated ZKPK protocols [3].

In our approach the BID is used for identity verification based on ZKPK. The BID is used together with a random secret r to generate a Pedersen commitment [9]. This commitment is used to construct a ZKPK proof. This proof is sufficient for verification purposes as it corresponds to the biometrics enrolled in the system. The commitment is enrolled with a party and can be used by any verifying party. The use of ZKPK proof enables us to support *two-factor* (i.e. the BID and the secret random r) verification. At the time of verification the individual needs both to provide r and to reconstruct the BID, to prove knowledge of the value committed at enrollment. To revoke a BID, the commitment corresponding to enrolled biometrics is added to a revocation list which is similar to certificate revocation lists [25] in a public key infrastructure. In our approach, we consider the case where a revocation list consists of the biometric commitments which have been revoked. After a commitment has been published in the revocation list, the individual cannot do a proof of knowledge with that BID because it relies on a revoked commitment.

Contributions. The key contributions of the paper are as follows. First we present algorithms for reliable and secure generation of BIDs from different types of biometrics. We focus on techniques that are suitable for fingerprints, irises and faces. Second, we propose an approach for encoding BIDs into cryptographic biometric commitments that are used in ZKPK at the time of verification. It follows from the zero-knowledge proof protocols that the cryptographic proofs do not leak information except for the fact that the verifier learns that the prover verifies the proof. As such the verifying party obtains no information about the characteristics of the real biometrics from the cryptographic proof. Therefore, multi-factor verification techniques can use one or more biometrics interoperably with one or more non-biometric features to achieve strong identity verification. Our protocols ensure that the privacy of the biometrics is preserved as the final BID does not reveal any information about the original biometric image. We also present a detailed security analysis of the resulting biometric verification system. We provide an empirical analysis of the biometric key generation for different types of biometrics in order to provide evidence of the correctness of the proposed algorithms. Finally, we briefly discuss several use scenarios for our techniques to identify relevant infrastructural and organizational requirements for the use of our technique.

The rest of the paper is organized as follows. In Section 2 we introduce the main algorithms for the BID generation. In Section 3 we present the experimental results. In Section 4 we develop a comprehensive analysis of the proposed solution. In Section 5 we discuss related work. Finally in Section 6 we make some concluding remarks and additional considerations concerning the use of our approach.

2. BIOMETRIC KEY GENERATION ALGORITHMS

In this section we first introduce some preliminary concepts related to the techniques underlying our proposed solution. Then, we discuss the two core algorithms for the BID generation, that is, the

SVD based image hashing algorithm and the SVM classification algorithm.

2.1 Preliminary Concepts

Singular Value Decomposition (SVD). SVD is a well known technique for factorizing a $m \times n$ matrix into a diagonal form. As proven by Golub and Loan [23], if A is a real m -by- n matrix, two orthogonal matrices exist:

$$U = [u_1, \dots, u_m] \in \mathbb{R}^{m \times m} \quad V = [v_1, \dots, v_n] \in \mathbb{R}^{n \times n}$$

such that

$$UAV^T = \text{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m \times n} \quad p = \min\{m, n\}$$

where V^T is the transpose of matrix V and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$. σ_i 's, $i = [1 \dots p]$, are the *singular values* of A , and the vectors u_j , $j = [1 \dots m]$, and v_k , $k = [1 \dots n]$, are the j th *left singular vector* and the k th *right singular vector* respectively. $\sigma_i(A)$ denotes the i th largest singular value of A .

The singular values of a matrix A are unique. The singular values σ_i 's reflect the variations along the corresponding i singular vectors. It can be shown that computation of the right singular vectors and the singular values can be obtained by computing the eigenvectors and eigenvalues of the symmetric matrix $M = A^T A$ where A^T is the transpose matrix of A .

Support Vector Machines (SVM). SVM [22] is a classifier based on statistical learning technique developed by Vapnik *et al.* [13]. It aims at finding optimal hyperplanes to determine the boundaries with the maximal margin separation between every two classes while training the classifier model. Then additional data, which is not used during the training, is used as test data and can be classified using the separate hyperplanes.

Let $\{x_i, y_i\}$, $i = [1, \dots, L]$, be a training data vector, where x_i is the data item and y_i , $y_i \in \{-1, +1\}$ is a class label. Given an input vector x , SVM constructs a classifier of the form

$$f(x) = \text{Sign}(\sum_{i=1}^L \alpha_i y_i K(x_i, x) + b)$$

where: α_i , $i = [1, \dots, L]$, is a non-negative Lagrange multiplier; each multiplier corresponds to an example from the training data; b is a bias constant; and $K(\cdot, \cdot)$ is a kernel function satisfying the conditions of Mercer's theorem [53]. Some frequently used kernel functions are the polynomial kernel $K(x_i, x_j) = (x_i \cdot x_j + 1)^d$ and the Gaussian Radial Basis Function (RBF) $K(x_i, x_j) = e^{-|x_i - x_j|^2 / 2\gamma^2}$. Note that there are several approaches adopting SVM for classification problems with three or more classes as well.

SVM applies to classification of vectors, or uni-attribute time series. To classify multi-attribute data, which are matrices rather than vectors, the multi-attribute data must be transformed into uni-attribute data or vectors. We use the combination of the SVD technique with SVM which has been explored by previous work [31, 37, 55]. SVD is used to reduce multi-attribute biometric data to feature vectors.

2.2 SVD Image Hashing

In this section we describe the hashing mechanism used in Phase 1 of BID generation. The techniques presented build on the basic image hashing process described in [30]. The main steps of the algorithm (summarized in Figure 2) are as follows.

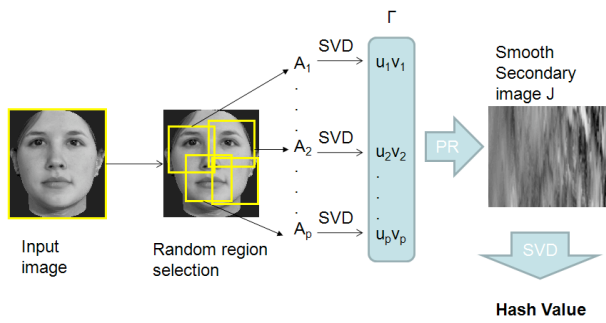


Figure 2: Key steps of the biometric image hashing algorithm.

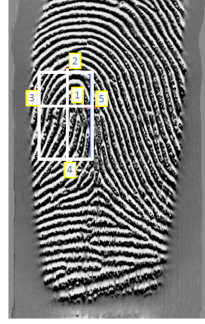


Figure 3: Fingerprint region of interest.

Pre-processing. As a first step the biometric image may be pre-processed so as to obtain a clear well focused biometric image I . Pre-processing provides an effective region in a selected biometric image for subsequent feature extraction. We support three types of biometric data: face, iris and fingerprint.

For the specific case of fingerprint image, as a part of pre-processing, the *region of interest* (ROI) is identified (See step 2 of Algorithm 1). The unique characteristics of the fingerprint are known to be around the core point or delta point [54]. The outside portion of a fingerprint is generally prone to small translations and is typically cropped out. Also, a larger area of the central portion of fingertip skin is in contact with the scanner surface as compared to the peripheries, giving a better image. The center is also better for liveness analysis. Since data such as the rate of perspiration can be measured, the center region is also more robust to pressure dispersion as compared to the other regions. Importantly, as the experimental results show, it preserves enough information to identify individuals. The procedure to determine the ROI corresponds to steps 6-15 of Algorithm 1 (see Figure 3). This ROI is then used as an image input for the rest of the algorithm (step 15 of Algorithm 1).

Feature Extraction. Once the image I of size $n \times n$ is finalized, the features are extracted based on a random region selection. The selection is executed by choosing p semi-global regions based on a pseudorandom (PR) generator that uses a secret key r . The obtained matrices corresponding to the selected sub-images (denoted by ρ_i) are then transformed under matrix invariant functions such as SVD.

The random partitioning of the image introduces unpredictability in the hash values and hence increases the security of the overall system. As long as these sub-images are sufficiently unpredictable, the

resulting intermediate hashes are also different with high probability [36]. The squares ρ_i 's determined in steps 18–23 and used in the partitioning (see Figure 2) are deliberately chosen to be overlapping to further reduce the vulnerability of the algorithm to malicious tampering. Note that an increased number of squares increases the pseudorandomness in the resulting hash value, and therefore helps in increasing security as explained in Section 4, assuming a secure pseudorandom number generator. As a further advantage, the random partitioning decreases the probability of collision and increases the robustness against noise that may be present in the biometric image. As reported in line 22 of Algorithm 1, the A_i 's, $1 \leq i \leq p$, are matrices corresponding to the selected sub-image blocks. Here each element of the matrix A_i corresponds to the 256 grey level value of the pixel of the selected sub-image. The encoding of the actual matrix used in the transformation is done based on the fact that every element in the matrix has a grey value g , $0 \leq g \leq 255$, a position v and a direction d . A single pixel may not have a direction, but for a group of pixels, the grey value may change hence defining a concrete direction. Grouping pixels is important as isolated components may not be robust.

Transformation. Each sub-image A_i , $1 \leq i \leq p$, is used to perform the SVD transformation. As a result for each A_i a unitary reduction to the diagonal form is performed to obtain $U_i S_i V_i^T$, $1 \leq i \leq p$, such that $A_i = U_i S_i V_i^T$. As such the SVD selects the optimal basis vectors in the L_2 norm³ sense such that, for any $m \times m$ real matrix A_i , we have

$$(\sigma_k, \vec{u}_k, \vec{v}_k) = \arg \min_{a, \vec{x}, \vec{y}} |A - \sum_{i=1}^{k-1} \sigma_i \vec{u}_i \vec{v}_i^T - a \vec{x} \vec{y}^T|_F^2$$

where: $1 \leq k \leq m$; $a \in \mathbb{R}$; $\vec{x}, \vec{y} \in \mathbb{R}^m$; $\sigma_1 \geq \sigma_2 \dots \geq \sigma_m$ are singular values, $\{\vec{u}_i\}$ and $\{\vec{v}_i\}$, $1 \leq i \leq p$, are the corresponding singular vectors; and $(\cdot)^T$ is the transpose operator [30]. By using the SVD we preserve both the magnitude of the important features in singular values and also their location geometry in the singular vectors. The combination of the left most and right most singular vectors which correspond to the largest singular values, in turn, captures the important geometric features in an image in the L_2 norm sense. Therefore as a next step for each A_i , \vec{u}_i^T , that is, the first left singular vector and \vec{v}_i^T , that is, the first right singular vector are retrieved. Those vectors are then combined in $\Gamma = \{\vec{u}_1^T, \dots, \vec{u}_p^T, \vec{v}_1^T, \dots, \vec{v}_p^T\}$.

The next step is to form a pseudorandom (based on pseudorandom numbers) smooth secondary image J from Γ . J is formed according to an iterative process, at each step of which an element from Γ is selected and added to J . As a first step an element is pseudorandomly selected from Γ and set at the first column of J . Then for the i^{th} column of J , an element from Γ is selected such that it is closest to the $(i-1)^{\text{th}}$ column of J in the L_2 norm sense as denoted in step 39 in Algorithm 1. An element can only be chosen once from Γ , therefore an element chosen at the i^{th} step cannot have been chosen at any of the previous $(i-1)^{\text{th}}$ steps. Hence after $2p$ steps all the elements of Γ are pseudo-randomly reordered to form the secondary image J of size $m \times 2p$. Note that the secondary image is required to ensure the one-way property of the SVD image hashing algorithm (See the analysis in Section 4).

Once J is formed, SVD is re-applied to it, to finally obtain the image hash vector (steps 49 – 52 of Algorithm 1). The left and right singular vectors are obtained by $J = U_J S_J V_J^T$. Then the

³ L_2 norm, defined for a vector $\vec{x} = \{x_1, \dots, x_n\}$ is denoted by $|\vec{x}| = \sqrt{\sum_{k=1}^n |x_k|^2}$.

singular vectors corresponding to the largest singular values, that is, the first left (\vec{u}_j) and the first right (\vec{v}_j) are chosen. These vectors are simply combined to obtain the final hash value $\vec{H} = \{\vec{u}_j, \vec{v}_j\}$.

2.3 SVM Classification

As discussed in the previous section, from one input biometric sample, a hash vector $\vec{H} = \{\vec{u}_j, \vec{v}_j\}$ of length $m + 2p$ is obtained. Since the hash vectors obtained from different biometric samples of the same user may be the same or may differ from sample to sample, we train a classifier to determine which hash values correspond to a given user (or class), so that at the time of verification, the classifier can identify the correct class of the user. To achieve this goal several biometric samples of different users are taken. Algorithm 1 is run on each sample to get the corresponding hash vector.

These samples are then divided into training and test data to perform the classification. We use K-fold cross-validation to divide the training and testing data. All sample hash vectors are partitioned into K subsamples. Of the K subsamples, a single subsample is retained as the validation data for testing the model, and the remaining K - 1 subsamples are used as training data. The cross-validation process is then repeated K times (the folds), with each of the K subsamples used exactly once as the validation data. The K results from the folds are then averaged to produce a single estimation [2].

The obtained hash vectors do not greatly differ with respect to the Euclidean distance, as inferred through experimental analysis; therefore we use SVM techniques to map the input hash vectors onto a higher dimensional space where a maximal separating hyperplane can be constructed.

As explained in Section 2.1 the hyperplane constructed using SVM is such that it has the maximum distance to the closest points of the training set. These closest points in the training set are called *support vectors*. Here we use the Gaussian radial basis kernel function (RBF for brevity) $K(\vec{H}_i, \vec{H}_j) = e^{-|\vec{H}_i - \vec{H}_j|^2 / 2\gamma^2}$ where \vec{H}_i and \vec{H}_j are two of the training samples and $\gamma > 0$.

During training, two specific parameters have to be assessed, namely γ used in the RBF kernel function and the penalty parameter C used in the evaluation of an optimal hyperplane balancing the tradeoff between error and margin. To select the pair with the best CV accuracy, all combinations of C and γ are tried using a grid search method [8]. After training, the SVM model encodes all the classes that this SVM classifier has been trained with.

Note that an increased number of classes increases the number of choices for an attacker executing guessing attacks on the SVM model, to guess the right BID. Additional classes can be added to the original SVM classifier model by training additional samples of the given biometrics. These samples have to be carefully added as the added classes, which do not resemble the original biometric classes, would most likely be easily ruled out by an attacker. We therefore employ a strategy to make the additional classes similar to the original set of classes. For each class in the SVM model we define a *protector class* which is similar to the original class so that the cluster formed by the *protector class* is close to the original SVM class, and yet is different enough to be distinguished as a different class. There could be different ways of obtaining the protector classes. The first is to find biometric images of different individuals which look perceptually similar. The second possibil-

Algorithm 1 Generic Biometric Image Hashing Algorithm

Require: Biometric image I
Ensure: The quality of the image is suitable based on biometric.

- 1: Input biometric image I
- 2: **{Pre-process fingerprint images to calculate ROI}**
- 3: **if** (type(I) == 'fingerprint') **then**
- 4: $point_1 = \text{Algorithm_R92}(I)$ {Compute core or delta point}
- 5: size = 4 {Set fingerprint ROI threshold size}
- 6: count = 0
- 7: **for** each line i in orthogonal directions (N,S,E,W) **do**
- 8: **repeat**
- 9: increment length of line;
- 10: **if** line encounters a ridge **then**
- 11: $point_i = \text{coordinate of intersection of line and ridge}$
- 12: count++
- 13: **end if**
- 14: **until** (count \neq size)
- 15: **end for**
- 16: $I = \text{crop}(point_2, point_3, point_4, point_5)$
- 17: **end if**
- 18: Let resultant image $I \in \mathbb{R}^{n \times n}$ be of size $n \times n$
- 19: **{Random Selection}**
- 20: Let p be the number of rectangles
- 21: Let ρ_i be the i^{th} rectangle and m be the height/width of ρ_i .
- 22: **for** each i where $1 < i < p$ **do**
- 23: Randomly position rectangle ρ_i at (x_i, y_i) such that $x_i + m < n$ and $y_i + m < n$
- 24: Let A_i be the "sub-image" that is formed by taking the portion of image that is in ρ_i : $A_i \in \mathbb{R}^{m \times m}$, $1 \leq i \leq p$.
- 25: **end for**
- 26: **{First SVD Transformation}**
- 27: **for** each A_i where $1 \leq i \leq p$ **do**
- 28: $A_i = U_i S_i V_i^T$ {Collect singular vectors corresponding to the largest singular value}
- 29: $\vec{u}_i^L = \text{first left singular vector}$
- 30: $\vec{v}_i^R = \text{first right singular vector}$
- 31: **end for**
- 32: $\Gamma = \{\vec{u}_1^L, \dots, \vec{u}_p^L, \vec{v}_1^R, \dots, \vec{v}_p^R\}$
- 33: Initialize secondary image $J[m, 2p]$ {Constructing secondary image from singular vectors}
- 34: **for** all c where $1 \leq c \leq 2p$ **do**
- 35: Initialize variable e_c corresponding to element in Γ
- 36: **if** $c = 1$ **then**
- 37: $e_c = PR_Select(\Gamma)$
- 38: **else**
- 39: var_loop = true
- 40: **while** var_loop **do**
- 41: $e_c = \min_{k=1}^{2p} (\sqrt{\sum_{l=1}^{c-1} (J(l) - \Gamma(k))^2})$
- 42: **if** not(e_c already chosen for J) **then**
- 43: var_loop=false
- 44: **end if**
- 45: **end while**
- 46: **end if**
- 47: **for** all r where $1 \leq r \leq m$ **do**
- 48: $J[r][c] = e_c[r]$
- 49: **end for**
- 50: **end for**
- 51: **{Second SVD Transform}**
- 52: $J = U_J S_J V_J^T$ {Collect singular vectors corresponding to the largest singular value}
- 53: $\vec{u}_j^L = \text{first left singular vector}$
- 54: $\vec{v}_j^R = \text{first right singular vector}$
- 55: $\vec{H} = \{\vec{u}_j^L, \vec{v}_j^R\}$
- 56: **return** Hash Value \vec{H}

ity is to add noise to the original biometric image. For example, the face images could be modified to render naturally asymmetric features to symmetric or changing other specific aspects as the size of the face characteristic such as the eyes, nose and so on. If there are n original classes, then we add a protector class for each, thus resulting in $2n$ classes. We also add other spurious classes which are not similar to the original biometric samples (as the protector classes) but are of the same biometric type.

As a final step, a combination of the classes is chosen based on SVM ranking which provides class prediction confidence of the SVM classifier. More specifically if n is the total number of classes, the final BID is the label of class with the highest confidence label and an unordered combination of the top $t = \frac{\alpha}{2}$ class labels which are listed with decreasing confidence levels. For an attacker to guess the BID, given the SVM classes, the number of choices is $n + \binom{n}{t}$ resulting in the final number of bits as $\log_2(n + \binom{n}{t})$. Considering the FAR for the primary class the final number of bits would be $\text{MIN}[\log_2(n), -\log_2(\text{FAR})] + \log_2(\binom{n}{t})$. We typically consider the total number of classes $n > 69$ which leads the number of choices to be $> 2^{64}$, thus making it computationally hard for the attacker to guess the right BID.

3. EXPERIMENTS

In this section we summarize the experimental results we conducted to assess the accuracy and robustness of our approach. We carried out extensive tests for different biometrics, to demonstrate that the relevant criteria required for the security, repeatability and uniqueness of the BID are met. All experiments have been conducted using Microsoft Windows XP Professional 2002 Service pack 1 operating system, with Intel(R) Pentium(R)4 3.20GHz and memory of 512MB.

3.1 Dataset and Experimental Setup

We tested our hashing algorithm (Algorithm 1, Section 2.2) on fingerprint, iris and face data. Summary information about the data used and the obtained results is reported in Table 2. For fingerprints we used FVC [34] databases. The FVC dataset used consists of overall 324 fingerprint images of 59 individuals collected using thermal sweeping and optical sensors. We also used 50 images of 10 individuals generated using the synthetic fingerprint generator SFingeGe v3.0 [7]. Regarding the iris data, the UBIRIS iris Database3 [44] was used which consists of 1695 images of 339 individuals' eyes. Finally for the face data we used the Yale Database of Faces [20] containing 100 images of 10 individuals and the AT&T Database of Faces [1, 46] containing 400 images of 40 individuals. We evaluated our results using the SVM classification algorithm, with K-fold cross validation (CV). Based on the CV accuracy, the False Acceptance Rate (FAR) was calculated. The FRR is calculated as $1 - \text{CV Accuracy}$, whereas the FAR is calculated as the number of false accepts divided by the number of tries.

The values used in the experiments for the key parameters of Algorithm 1 are reported in Table 1, where n is the size of the image in pixels, p is the number of sub-images, m is the size in pixels for each of the sub-images, and J is the secondary image.

To assess the optimal values for p and m , we ran experiments with various possible combinations of the values and used the one which provided the maximum accuracy. For example for the fingerprint database FVC2004 DB3_B, the value of p was varied between

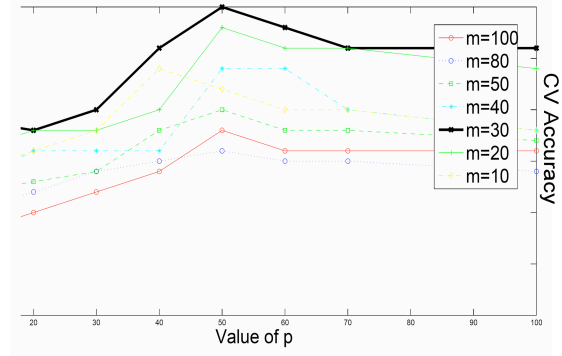


Figure 4: Plot of different values of number of sub-images (p); the image size of sub-images (m); and the corresponding CV accuracy.

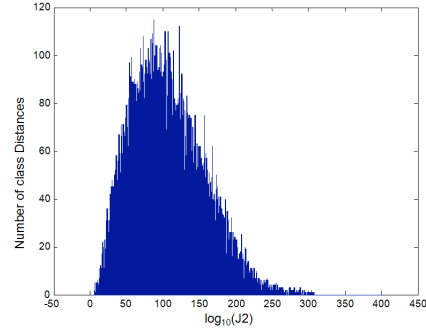


Figure 5: J2 histogram of iris classification.

[10, . . . , 100] and the value of m between [10, . . . , 100] (See Figure 4); the highest accuracy was found for $p = 50$ and $m = 30$.

The code for implementing the various steps is written in MATLAB and the `rand()` function of MATLAB is used as the pseudo random function used in step 21 and 35 of Algorithm 1. The size of the secondary image J is 30×100 leading to the size of $\vec{u}_j = 30 \times 1$ and $\vec{v}_j = 100 \times 1$, thus resulting in a hash vector $\vec{H} = \{\vec{u}_j, \vec{v}_j\}$ of 130 dimensions.

For the SVM classification we adopted the LIBSVM [8] package to generate the hash vectors and build the final classifier model. This uses the RBF as the kernel function. Based on experimental analysis, C was set to the range $\{2^5, \dots, 2^{15}\}$ and γ to $\{2^{-5}, \dots, 2^3\}$. All combinations C and γ were tried using grid search to select the best CV accuracy based on the input data.

Image type	n	p	m	J size	\vec{H} size
Fingerprint/Iris/Face	128	50	30	30×100	130

Table 1: Parameter values for experiments on Algorithm 1.

3.2 Experimental Results

We now discuss the results of the experimental evaluation of our approach. First, regarding the time performance, on the average, the hash vector from any given image is generated in 0.9597 seconds. The generation of SVM model for about 220 persons' hash vectors takes 3 or 4 hours. At the testing stage, once the model is

#	Biometric Type	Database Name	Description	# Images	# Persons	CV Accuracy %	FRR %	FAR %
1.	Finger-print	FVC2004, DB3_B	300 × 480, Thermal Sweeping Sensor	54	9	92.59	7.41	9.26×10^{-03}
2.	Finger-print	FVC2004, DB3_A	300 × 480, Thermal Sweeping Sensor	150	30	97.33	2.67	9.21×10^{-04}
3.	Finger-print	FVC2004, DB2	328 × 364, Optical Sensor	120	20	85.83	14.17	7.46×10^{-03}
4.	Finger-print	SFingGe v3.0, Synthetic Generator	288 × 384	50	10	88	12	1.33×10^{-02}
5.	Iris	UBIRIS.v1 Sessao_1	800 × 600 – 24 bit color	1100	220	87.73	12.27	5.6×10^{-04}
6.	Iris	UBIRIS.v1 Sessao_2	800 × 600 – 24 bit color	595	119	97.65	2.35	1.99×10^{-04}
7.	Face	The Yale Face Database B	640 × 480 – 8 bit gray scale	100	10	99	1	1.11×10^{-03}
8.	Face	AT & T Databases of Faces	92 × 112 – 256 bit gray scale	400	40	98.25	1.75	4.49×10^{-04}

Table 2: Summary of the experimental results of all biometric data types.

generated, it takes approximately 0.001 second to classify the test images.

Regarding the experimental results, the obtained results largely confirm the correctness of our algorithm: in each of the test cases, the accuracy was above 85% cross validation. False acceptance rates were within the interval $[1.99 \times 10^{-04}, 1.33 \times 10^{-02}]$, which translates into the assurance that the chances of accepting an incorrect biometric image are low. The worst observed FAR value is 1.33×10^{-2} , which interestingly is obtained for the images generated by the synthetic fingerprint generator, where the conditions for biometric generation were generally better controlled (e.g., there was no unexpected noise because of human interaction). Regarding FRR, the worst observed FRR value was in conjunction with the worst accuracy results since the FRR result is dependent on the accuracy (see previous section). The worst rate amounts to 14% (test case n. 3) and it is still acceptable, as it is in the same order of similar biometric key generators [24]. Additional insights specific to the different types of tested biometrics are discussed in what follows.

Fingerprint. Two types of Fingerprint Verification Competition (FVC) databases [34] corresponding to two types of sensors were used for the fingerprint biometric experiments. The sensors highly influence the quality of fingerprint images. We define the *quality* of the fingerprint image according to three criteria [28]: (i) high contrast between ridges and valleys, (ii) the image area foreground, and (iii) little scar or latency. As shown by the results, the CV cross validation is above 85% for each data set considered, which confirms the validity of our approach. A first important consideration suggested by the experimental results is that the algorithm performs better in case of large data set (as in the test case n. 2 in Table 2), most likely because of the more accurate training and testing during the configuration phase which helped in finding the optimal configuration parameters. We also notice that on average our algorithm performs better when using the thermal sensor than when using the optical sensor because the thermal sensor captures better quality fingerprint images. We can explain this result by elaborating more on how the quality is affected, in that the quality of the fingerprint image is affected by several human factors such as skin humidity and pressure. If the skin humidity is lower, the image quality of

the optical sensor degrades. The skin humidity does not affect the image quality of the thermal sensor because it is the sweeping type. Moreover, regarding pressure, for optical sensor the foreground image is smaller for low pressure, while the fingerprint is smeared for high pressure. This is again not true for thermal sweeping sensor where the image quality is not significantly affected.

Note that the last data set was composed of artificially generated images. We experimented with synthetic fingerprint images as they potentially supply non-biased images and can be created at a low cost. It was difficult to control the randomness which lowered the cross validation classification accuracy to 88%. We believe the results could be improved using synthetic generator version which generates several samples corresponding to a single individual, maintaining the invariant features of an individual for all samples.

Iris. We used the UBIRIS.v1 Sessao_1 (Session 1) and UBIRIS.v1 Sessao_2 (Session 2) [44, 43] iris databases. For the first image capture session, noise factors, such as reflections, luminosity and contrast, were minimized. In the second session the capture place was changed to introduce a natural luminosity factor. Images collected in the second session simulated the ones captured by a vision system without or with minimal active participation from the subjects, adding possible noise to the resultant images. Note that when capturing iris images, some pre-processing is performed. A sequence of images is obtained rather than a single image. Not all images in the input sequence are clear and sharp enough for recognition. The images may be out of focus, or contain interlacing lines caused by eye motion or have severe occlusions caused by eyelids and eyelashes. Therefore, only high quality images from an input sequence are included in the final database.

Face. We used two databases for these experiments. The first one collected good quality images, in that photos were taken with subjects in frontal pose. Thus the resulting cross validation accuracy was 99%. The second set of tests was performed on images taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position with

tolerance for some side movement. Despite this, the overall cross validation accuracy of this database was 98.25% although the false rejection rate increased by .75%.

4. ANALYSIS

We start with proving some key properties related to uniqueness and repeatability and security properties of the BID generation algorithms. Based on such results we analyze privacy aspects and discuss how to prevent from possible attacks.

4.1 Uniqueness and Repeatability

A criterion frequently used for assessing uniqueness and repeatability in classification is the J_2 function [32]. The key idea of the J_2 function is to compare the *within-class* distance of the various hash vectors (or elements being classified) belonging to a given class, with the *between-class* distance among the various classes. There are two key steps to be taken while evaluating J_2 .

The first step is to evaluate the *within-class* scatter matrix S_w : $S_w = \sum_{i=1}^M S_i P_i$ where M is the total number of classes; $S_i = E[(x - \mu_i)(x - \mu_i)^T]$ is the covariance matrix⁴ for a class denoted by w_i where E is the expected value function, x is any vector in class w_i and μ_i is the mean vector of class w_i ; and, $P_i = n_i/N$ where n_i is the number of samples in class w_i and N is the total number of samples in all the classes.

The second step is to evaluate the *between-class* scatter matrix S_b : $S_b = \sum_{i=1}^M P_i (\mu_i - \mu_o)(\mu_i - \mu_o)^T$ where $\mu_o = \sum_{i=1}^M P_i \mu_i$ is the global mean vector of all the classes.

From the above a covariance matrix of feature vectors with respect to the global mean is evaluated as $S_m = S_w + S_b$. Finally the J_2 criterion is calculated as: $J_2 = \frac{|S_m|}{|S_w|}$. As it is evident from the equation, for good repeatability of correct classification (small within-class distance), and uniqueness (large between-class distance) the value of J_2 should be large.

We carried out additional experiments on all the datasets to estimate J_2 and obtained average values of J_2 for fingerprint as 1.2712×10^{81} , iris as 1.5242×10^{303} and face as 3.7389^{103} . These values of J_2 and the corresponding classification accuracy (See Table 2) provide empirical evidence that the algorithm satisfies the uniqueness requirement on the biometric hashes generated based on the biometric datasets provided.

For clarity, we provide an example of a J_2 histogram for the Iris Session 1 database in Figure 5 (data corresponding to test case n. 5 in Table 2). Note that the J_2 metric requires the calculation of *within class* and *between class* distances of all the possible pairs of data elements. The y axis in the histogram presents the values of $\log(J_2)$ class distances between any two classes. For instance for a value (120(x-axis),100(y-axis)) means that there are 100 class distances which have the J_2 value of 120. If there are all together $|C|$ number of total classes then the possible permutations of the distances to be tested are $\frac{|C| \times (|C|-1)}{2}$.

4.2 Biometric Image Keyed Hashing

We analyze the one-way security property of the SVD based biometric image hashing algorithm. More specifically, we show that

⁴Covariance is the measure of how much two random variables vary together. A covariance matrix is a matrix of covariances between elements of a vector.

Type	n	spurious	η	# bits
Fingerprint	69	-	2.84×10^{19}	64
Fingerprint	139	69+1	2.36×10^{40}	134
Iris	220	-	4.52×10^{64}	214
Iris	119	-	2.43×10^{34}	114
Face	101	50+1	1.01×10^{29}	96

Table 3: Summary of number of SVM classes and entropy.

it is computationally hard, given the BID hash vector \vec{H} to reconstruct the original biometric image. We prove this result by the following two theorems. First, we prove that it is hard to construct the secondary image from the vector, which is required for reconstructing the original biometrics. The result (Theorem 2) shows that even if the second image is constructed or attacked, it is still hard to obtain the original biometric image I . Our results are based on the combination of mathematical properties of the SVD and the employed hashing technique.

THEOREM 1. *Let \vec{u}_J and \vec{v}_J be the vectors which form the final hash value $H(u_J, v_J)$, and let λ_i be non-zero eigen values of the matrix $J^T J$ where J is the secondary image. If there is no λ_i that is dominant, then it is computationally hard to construct the secondary image from $H(u_J, v_J)$.*

Because in our theoretical results the assumption that there is no dominant eigenvalue is crucial, we have carried out extensive an experimental analysis on the biometric images to assess whether such assumption holds. Our experimental results show that such assumption holds because of the smoothness of the secondary image. A proof sketch of the theorem is reported in Appendix A.

THEOREM 2. *Given the secondary image it is computationally hard to obtain the original image I .*

Proof Sketch in Appendix A.

As a final remark we note that even if the attacker is able to retrieve the biometric image, it cannot reconstruct the hash vector without the knowledge of the secret random value needed during the selection of the p sub-images and to pseudorandomly combine them to form the secondary image J .

4.3 SVM Classes and BID Space

From the empirical analysis during the classification experiments provided in Section 3, we observe that if n is the number of classes, and these classes are listed in decreasing order of their confidence level, the highest confidence class is the same and the unordered set of the following t classes where $(n-1) \geq t \geq \frac{n}{2}$ is the same for the multiple testing rounds in the K-fold validation. In general, for most SVM classification experiments for all three biometrics, the ordering of several of the t classes was swapped with the neighboring classes. Therefore for the final label which denoted the final BID value, we use the class with the highest confidence followed by an unordered combination of the next t classes. For an attacker to guess the right key based on the classifier model, the number of choices would be $\eta = n + \binom{n}{t}$, under the assumption that each class has the same likelihood. Based on the uniqueness analysis from the J_2 metric we observe that the samples considered have

large inter-class distances, thus avoiding centroid formations that would narrow down the attacker’s number of choices. As part of future work, we plan to further investigate inference-based attacks on the SVM model, which could potentially help the attacker make better guesses about the combination of classes used for generating the BID.

As noted from the experiments n in our case ranges in the interval [69, 220]. Based on the value of n , the resulting η ranges in the interval $[2^{64}, 2^{214}]$. η is proportional to the number of bits needed to encode the BID. More precisely the number of bits, considering the FAR for the primary class, is $MIN[\log_2(n), -\log_2(FAR)] + \log_2(\binom{n}{t})$. This results in the number of bits ranging in the interval [64, 214]. A summary of the experimental data corresponding to the biometric type, n , η and final number of bits of the BID is provided in Table 3.

4.4 Privacy and Security Analysis

We now analyze the relevant privacy and security properties of our technique, based on the above results. In addition we briefly analyze how our commitment technique is employed in the multi-factor approach to identity verification.

4.4.1 Privacy Analysis

Privacy in our context includes the following properties: unlinkability of the BID to the source biometric image, anonymity and confidentiality.

Unlinkability: Unlinkability refers to the impossibility of linking the BIB with a source biometric image. This property holds in our approach as a consequence of the irreversibility results of Theorems 1 and 2. The one-way nature of the BID generation process guarantees that there is no way to reconstruct the biometric image from the BID.

Confidentiality: Confidentiality refers to keeping the biometrics confidential throughout all the processing steps of the BID lifecycle. We protect confidentiality of the image as follows. First, once the biometric image is captured, the conversion phase only requires the hashing secrets and the SVM classifier model (referred to as the meta-data). Specifically, only the classifier model is permanently recorded by the system. During the verification phase, only the hash values obtained after processing the biometric images are used. Clear text images and templates are not required, so as to minimize information exposure. Therefore the only code that needs to be trusted to assure confidentiality of the biometric image is the code that given the initial image generates the hash value. Such code must be trusted not to leak the image and to discard the image once the hash value has been generated; the code is small and thus can be easily verified. We remark that confidentiality is preserved even in case an attacker gains partial information related to the BID. Since the BID and the biometric image are unlinkable, the confidentiality of the biometric image is preserved, as given the BID, given the unlinkability of the BID with the biometric image.

Anonymity: Anonymity refers to the property that prevents an individual to be identifiable within a set of subjects [42]. Our approach also assures anonymity, provided that no other identifying information is used in combination with the BID ZKPK proofs needed for verification. The generated BID, in fact, does not reveal any unique physiological information about the user’s identity which is one of the key problems in typical matching based biometric verification. Also it follows from the unlinkability and

confidentiality properties that the attacker cannot recreate the hash values given the biometric image and also cannot link a BID to an actual individual.

4.4.2 Security Analysis

Security in our system is given by the difficulty of perpetrating impersonation attacks.

We make two key assumptions in order to achieve a high-assurance BID generation. First, we assume that the sensor which captures the biometric image is able to detect *live* images and does not leak the image or information about the image. Second, we assume that the pseudorandom hashing secret used in Phase 1 is not compromised. If at least one of the two assumptions holds, then the BID cannot be compromised, as elaborated further in the analysis below.

We now focus on an attacker trying to impersonate a given user based on the BID and show how our approach withstands these types of attacks. We analyze the attackers’ options by considering each of the secrets involved in the system.

The various possible points of attack include (A) biometric image; (B) hashing secrets; (C) classifier model used in Phase 2 (see Figure 2); (D) BID and possibly additional secrets and components depending on other cryptographic components used. The secrets of the system are the hashing secrets used in Phase 1 and the random commitment secret which is used together with the BID to create the cryptographic commitment. The classifier model is not assumed to be secret. Precisely, the classifier model can be revealed without jeopardizing the protocol security if the number of classes n is greater than 69. This is because $n > 69$ (69 is the minimum sample size used in our experiments) would make the number of possibilities greater than 2^{64} thus ensuring computational hardness. As described in Section 4.3, increasing the value of n by adding classes increases the keyspace; making it computationally hard for an attacker to perform a brute force attack.

	A	B	C	D	E	Attack Prevention Summary
1	×					BID cannot be created without hashing secrets.
2	×	×				BID cannot be created without classifier model.
3	×		×			The classifier model does not allow inference of the hashing secret needed construct BID.
4	×	×	×	×		The BID is compromised, but the commitment secret prevents from creating ZKPK.
5				×		The BID is compromised, but the commitment secret prevents from creating ZKPK. No other secrets are leaked.
6		×	×		×	All stored information is compromised but the BID cannot be created without biometric image.

Table 4: Possible security attacks [key: (A) biometric image (B) hashing secrets (C) classifier model (D) BID (E) commitment secret; ×: the value is known to the attacker].

To succeed in an impersonation attack the attacker needs to know all the secrets required to create the BK. In order to gather the other secrets, the attacker would have to pass the verification methods and compromise the system. Bypassing the cryptographic ZKPK protocol is computationally hard [18, 5]. Additionally, the cryptographic ZKPK protocol prevents replay attacks: the attacker cannot use the proofs created during a given biometric verification process

in any another verification process. Table 4 provides a summary of the various cases in which one or more secrets are compromised, and reports possible security implications. Case 1, 2 and 3 address the cases in which the biometric image is known to the attacker, but not the meta-data, which includes the hashing secret and classifier model, nor the random secret in the BID commitment, which are stored by the user. Thus, in these cases the attacker is not able to generate the BID. However, if the attacker knows the BID, then to perform successful verification it also needs the commitment secrets. This scenario is summarized by case 4. As noted earlier the knowledge of the BID does not reveal any information about the biometric image or the secrets involved as shown in case 5.

Finally, an interesting case is when the stored information including the meta-data and the commitment secret are compromised (case 6). In this case, the attacker’s best choice as a source of information is the SVM model. However as we show in Section 4.3, for number of classes $n > 69$, the number of choices $> 2^{64}$ which makes it computationally hard for the attacker to guess the right BID.

5. RELATED WORK

Biometrics-based key generation has been extensively investigated in the past years. As mentioned earlier, the biometrics-based key generation is characterized by two stages. At the first stage certain biometric features are used to compute a bit string representing that biometrics. The bit string is then used in the second stage to generate a unique cryptographic key with the help of stored meta data. If two instances of the bit strings are sufficiently similar then the cryptographic key generated is the same. In most approaches, the second stage is independent of the biometrics being used, whereas the first is mostly biometric-specific.

The first approach to biometrics-based key generation is by Soutar *et al.* [50, 49, 48]. They developed methods for generating a repeatable cryptographic key from fingerprints using optical computing and image processing techniques. Following Soutar’s work several strategies have been proposed for improving the second-stage of the key generation. Davida *et al.* [15] described a second-stage strategy using error correcting codes (ECC) and how it could be used with first-stage approaches for generating a bitstring representing iris scans [14]. The second-stage approach was significantly improved by Juels *et al.* [26, 27]. The underlying intuition behind the error correction and similar schemes can be understood based on Shamir’s secret sharing scheme [47]. The hardness of Shamir’s secret sharing scheme is based on the *polynomial reconstruction* problem which is a special case of the Reed-Solomon list decoding problem [4]. In fuzzy vault scheme proposed by Juels [27] based also on ECC, the user adds spurious *chaff* points which make it infeasible for an attacker to reconstruct the polynomial representing the BK.

Since the introduction of the fuzzy vault scheme, several researchers have implemented it in practice [11, 57, 17, 10, 19, 51, 40]. In particular the most recent work is by Nandakumar *et al.* [40] where the fuzzy vault implementation is based on the location of minutia points in a fingerprint. They generated 128 bit keys and obtained an accuracy rate of 91% for high quality images and 82.5% for medium quality images. The FRR was approximately 7% which shows an improvement over several other implementation of this scheme (where the average FRR was from 20-30%). From the experimental point of view, we generate 134 bit keys with the accuracy of 94.96% for high quality images and 86.92% for medium quality images. The FRR was on an average 9.06% which is com-

parable to the above scheme. From the algorithmic point of view, we use a similar concept of chaff points while adding spurious classes to make it hard for the attacker to guess the correct final key. We do not use ECC to retrieve the final key, but plan to investigate how ECC can be used while finding a list of SVM classes uniquely ordered by the confidence measures (See Section 4.3). A major difference of our approach with respect to the stage-one approaches of the various implementations of the fuzzy-vault is that their feature extraction is specific to the type of biometrics. Dependence on specific features has led to brute force attacks on several fuzzy vault implementations [35]. In our case, we instead use image analysis which can be used for several generic 2D biometric images such as fingerprint, iris and face.

Another scheme which makes use of the *polynomial reconstruction* problem in the second-stage is the scheme proposed by Monroe *et al.* which was originally used for hardening passwords using keystroke data [39] and then extended for use in cryptographic key generation from voice [38]. Let us consider the case when m biometric features are recorded at stage-one. When the system is initialized the main key κ and $2m$ shares of κ are generated using generalized secret sharing scheme. The shares are arranged within an $m \times 2$ table such that κ can be reconstructed from any set of m shares consisting of one share from each row. The selection is based on the biometric features recorded. Monroe *et al.* show that it is computationally infeasible for an attacker to guess the right shares because of the random or spurious shares present in the table. We also add spurious classes in the SVM classification model to make it infeasible for the attacker to guess the BID. Moreover, the features they capture in stage-one for key stroke [39] are durations and latencies, whereas for the voice [38] are the cepstral coefficients. Their experimental evaluation shows an average about 20-30% FRR. This biometric encoding of voices is not comparable with ours as we consider different biometrics which can be represented in 2D images.

Several of the techniques have been recently extended in the context of bio-hashing [33, 29, 12]. The approaches closest to ours are the bio-hashing techniques by Goh and Ngo [21, 41] who propose techniques to compute cryptographic keys from face bitmaps. *Bio-hashing* is defined as a transformation from representations which have a high number of dimensions and high uncertainty (example face bitmaps) to representations which have a low number of dimensions and zero uncertainty (the derived keys). Like our work, the goal of using the image hashing techniques is to extract bits from face images so that all similarly looking images will produce almost the same bit sequence. However, the work mainly focuses on the first stage of biometrics-based key generation and proposes the potential use of Shamir’s secret sharing techniques [47] in the second stage. With respect to the first stage, Goh and Ngo use principal component (PCA) analysis for analyzing the images. This is similar to our use of SVD, as both SVD and PCA are common techniques for analysis of multivariate data. There is a direct relation between PCA and SVD in the case in which principal components are calculated from the covariance matrix. An important capability distinguishing SVD and related methods from PCA methods is the ability of SVD to detect weak signals or patterns in the data which is important in our case as we propose to use our techniques for generic 2D biometric images. The methodologies we employ for stage-one also differs in that the biometric hash vector output from stage-one cannot be simply distinguished using straight forward implementation of hamming distance based analysis as proposed in [21, 41]. We instead combine stage-one and stage-two

with the use of SVM classifiers in stage-two which provides a way to analyze the properties such as inter and intra-class distance of the biometric hash vectors. We provide a detailed analysis of our approach which has not been developed in earlier bio-hashing work.

There are other biometric cryptosystems in which biometric authentication is completely decoupled from the key release mechanism. The biometric template is stored on the device and when the biometric match happens, the cryptographic key is released [52]. This approach however has several vulnerabilities and is not related to our key generation approach.

6. CONCLUSION

In this paper we have presented a novel approach for generating BIDs from 2D biometric images. These BIDs can be used together with other identity attributes in the context of multi-factor identity verification techniques. In the proposed approach the secure management of the BID's random secret is an important issue. To address such issue there are approaches that provide a secure and usable way to manage and store those random secrets. One such approach [56] uses cellular phones based on NFC (Near Field Communication) technology and allows users to store secrets on the phone as well as to split them among various phone components (including an external card) and also on an additional external device for increased security. From the user side, configuration is very easy in that the user has a menu with three security levels (low, medium, high) among which to choose. Each such level corresponds to a different splitting strategy. We refer the reader to [56] for more details.

In addition to the technical solution provided in the paper, we have also investigated organizational requirements based on the potential scenarios where our approach would be most likely used⁵. In particular, the security of the initial enrollment is crucial for the overall process. We have developed cases in which enrollment has high assurance and it is performed at controlled and secure enrollment points. By contrast, in a non-secure enrollment, additional verification steps are needed to attest the biometric key generation software and the storage medium used for storing the user secret keys. We have thus explored the possible media used to store the secrets and benchmarked them to identify the most suitable media. Similar considerations apply to the verification locations, which may be protected or unprotected. Such analysis has been instrumental for clarifying the relevant preconditions that need to be met to successfully apply our approach, and to identify possible non-technical limitations.

We plan to further investigate possible attacks on the classification model to see if guessing attacks can reduce the entropy of the biometric samples considered. The η provided in Section 4 assumes that there are no guessing attacks as the J_2 value is high. However, there may be additional attacks such as those discovered by Mihailescu in [35] relevant to Fuzzy Vault schemes where the entropy of the scheme was significantly reduced as a result of the attacks.

7. REFERENCES

- [1] AT & T Databases of Faces.
<http://www.cl.cam.ac.uk/research/dtg/attachive/facedatabase.html>.

⁵Details concerning the organizational requirements for our biometric verification protocols are reported in a technical report, which we are unable to refer because of the double blind review requirements.

- [2] K-Fold Cross Validation. <http://en.wikipedia.org/wiki/Cross-validation>.
- [3] A. Bhargav-Spantzel, A. C. Squicciarini, R. Xue, and E. Bertino. Practical identity theft prevention using aggregated proof of knowledge. Technical report, CS Department, 2006. CERIAS TR 2006-26.
- [4] D. Bleichenbacher and P. Q. Nguyen. Noisy polynomial interpolation and noisy Chinese remaindering. *Lecture Notes in Computer Science*, 1807:53–77, 2000.
- [5] J. Camenisch and A. Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In B. Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer Verlag, 2001.
- [6] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology – CRYPTO '04*, 2004.
- [7] R. Cappelli. SFinGe: an approach to synthetic fingerprint generation. In *International Workshop on Biometric Technologies (BT2004)*, pages 147–154, Calgary, Canada, June 2004.
- [8] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [9] D. Chaum and T. P. Pedersen. Wallet databases with observers. In *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pages 89–105, London, UK, 1993. Springer-Verlag.
- [10] Y. Chung, D. Moon, S. Lee, S. Jung, T. Kim, and D. Ahn. Automatic alignment of fingerprint features for fuzzy fingerprint vault. In *In Proceedings of Conference on Information Security and Cryptology*, pages 358–369, Beijing, China, Dec. 2005.
- [11] T. C. Clancy, N. Kiyavash, and D. J. Lin. Secure smartcard based fingerprint authentication. In *WBMA '03: Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications*, pages 45–52, New York, NY, USA, 2003. ACM Press.
- [12] T. Connie, A. Teoh, M. Goh, and D. Ngo. Palmhashing: A novel approach for cancelable biometrics. *Information Processing Letters*, 93(1):1–5, 2005.
- [13] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [14] J. Daugman. Biometric personal identification system based on iris analysis. In *United States Patent*, 1994.
- [15] G. Davida, Y. Frankel, and B. Matt. The relation of error correction and cryptography to an offline biometric based identification scheme. In *Proceedings of WCC99, Workshop on Coding and Cryptography, 1999.*, 1999.
- [16] R. Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *SOUPS '05: Proceedings of the 2005 Symposium on Usable Privacy and Security*, pages 77–88, New York, NY, USA, 2005. ACM Press.

- [17] Y. C. Feng and P. C. Yuen. Protecting face biometric data on smartcard with reed-solomon code. In *Proceedings of CVPR Workshop on Privacy Research In Vision*, page 29, New York, USA, June 2006.
- [18] U. Fiege, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. In *STOC '87: Proceedings of the nineteenth annual ACM conference on Theory of computing*, pages 210–217, New York, NY, USA, 1987. ACM Press.
- [19] M. Freire-Santos, J. Fierrez-Aguilar, and J. Ortega-Garcia. Cryptographic key generation using handwritten signature. In P. J. Flynn and S. Pankanti, editors, *Proceedings of SPIE: Biometric Technology for Human Identification III*, volume 6202, 2006.
- [20] A. Georghiadis, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.
- [21] A. Goh and D. C. Ngo. Computation of cryptographic keys from face biometrics. In *Communications and Multimedia Security*, volume 2828 of *LNCS*, pages 1–13, 2003.
- [22] K.-S. Goh, E. Chang, and K.-T. Cheng. Support vector machine pairwise classifiers with error reduction for image classification. In *MULTIMEDIA '01: Proceedings of the 2001 ACM workshops on Multimedia*, pages 32–37, New York, NY, USA, 2001. ACM Press.
- [23] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 1983.
- [24] F. Hao, R. Anderson, and J. Daugman. Combining crypto with biometrics effectively. *IEEE Transactions on Computers*, 55(9):1081–1088, 2006.
- [25] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile, 2002.
- [26] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security*, pages 28–36, 1999.
- [27] A. Juels and M. Wattenberg. A fuzzy vault scheme. In *Proceedings of IEEE International Symposium on Information Theory, 2002.*, 2002.
- [28] H. Kang, B. Lee, H. Kim, D. Shin, and J. Kim. A study on performance evaluation of fingerprint sensors. In *Audio and Video Based Biometric Person Authentication*, pages 574–583, 2003.
- [29] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You. An analysis of biohashing and its variants. *Pattern Recognition*, 39(7):1359–1368, 2006.
- [30] S. S. Kozat, R. Venkatesan, and M. K. Mihcak. Robust perceptual image hashing via matrix invariants. In *International Conference on Image Processing*, pages V: 3443–3446, 2004.
- [31] C. Li, L. Khan, and B. Prabhakaran. Real-time classification of variable length multi-attribute motions. *Knowledge Information Systems*, 10(2):163–183, 2006.
- [32] C.-C. Li and K. S. Fu. Machine-assisted pattern classification in medicine and biology. *Annual Review of Biophysics and Bioengineering*, 9:393–436, 1980.
- [33] A. Lumini and L. Nanni. An improved biohashing for human authentication. *Pattern Recognition*, 40(3):1057–1065, 2007.
- [34] D. Maio and D. Maltoni. FVC2004: third fingerprint verification competition. <http://bias.csr.unibo.it/fvc2004/>, 2004.
- [35] P. Mihalescu. The fuzzy vault for fingerprints is vulnerable to brute force attack. Technical report, University of Göttingen, 2007.
- [36] M. K. Mihçak and R. Venkatesan. New iterative geometric methods for robust perceptual image hashing. In *DRM '01: Revised Papers from the ACM CCS-8 Workshop on Security and Privacy in Digital Rights Management*, pages 13–21, London, UK, 2002. Springer-Verlag.
- [37] X. min Tao, F. rong Liu, and T. xian Zhou. A novel approach to intrusion detection based on SVD and SVM. *Industrial Electronics Society*, 3(2–6):2028–2033, November 2004.
- [38] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel. Cryptographic key generation from voice. In *SP '01: Proceedings of the 2001 IEEE Symposium on Security and Privacy*, page 202, Washington, DC, USA, 2001. IEEE Computer Society.
- [39] F. Monrose, M. K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. In *CCS '99: Proceedings of the 6th ACM conference on Computer and communications security*, pages 73–82, New York, NY, USA, 1999. ACM Press.
- [40] K. Nandakumar, A. K. Jain, and S. Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. In *IEEE Transactions on Information Forensics and Security, 2007 (To appear)*, 2007.
- [41] D. C. Ngo, A. B. Teoh, and A. Goh. Biometric hash: high-confidence face recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(6):771–775, June 2006.
- [42] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. pages 1–9. 2001.
- [43] H. Proença and L. A. Alexandre. UBIRIS: a noisy iris image database. In *ICIAP 2005: International Conference on Image Analysis and Processing*, volume 1, pages 970–977, 2005.
- [44] H. Proença and L. A. Alexandre. Toward non-cooperative iris recognition: A classification approach using multiple signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Special Issue on Biometrics*, 9(4):607–612, July 2007. ISBN 0162-8828.
- [45] A. Ross, A. K. Jain, and J.-Z. Qian. Information fusion in biometrics. In *Pattern Recognition Letters*, volume 24, pages 2115–2125, September 2003.
- [46] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *IEEE Workshop on Applications of Computer Vision*, Sarasota (Florida), December 1994.

- [47] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
- [48] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar. Biometric encryptionTM - enrollment and verification procedures. In *SPIE 98: In Proceedings of Optical Pattern Recognition IX*, volume 3386, pages 24–35, 1998.
- [49] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar. Biometric encryptionTM using image processing. In *SPIE 98: In Proceedings of Optical Security and Counterfeit Deterrence Techniques II*, volume 3314, pages 178–188, 1998.
- [50] C. Soutar and G. J. Tomko. Secure private key generation using a fingerprint. In *Proceedings of Cardtech/Securetech Conference*, volume 1, pages 245–252, May 1996.
- [51] U. Uludag and A. Jain. Securing fingerprint template: Fuzzy vault with helper data. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 163, Washington, DC, USA, 2006. IEEE Computer Society.
- [52] U. Uludag, S. Pankanti, S. Prabhakar, and A. Jain. Biometric cryptosystems: Issues and challenges. In *Proceedings of the IEEE, Special Issue on Enabling Security Technologies for Digital Rights Management*, 2004., volume 92, 2004.
- [53] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
- [54] S. Wang and Y. Wang. Fingerprint enhancement in the singular point area. *IEEE Signal Processing Letters*, 11(1):16–19, January 2004.
- [55] Y. Wang, Y. Sun, M. Liu, P. Lv, and T. Wu. Automatic inspection of small component on loaded PCB based on SVD and SVM. In *Mathematics of Data/Image Pattern Recognition, Compression, and Encryption with Applications IX.*, volume 6315 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, September 2006.
- [56] J. Woo, A. Bhargav-Spantzel, A. Squicciarini, and E. Bertino. Verification of receipts from m-commerce transactions on nfc cellular. In *10th IEEE Conference on E-Commerce Technology (CEC 08)*, July 2008.
- [57] S. Yang and I. Verbauwhede. Automatic secure fingerprint verification system based on fuzzy vault scheme. In *ICASSP '05: Proceedings of the Acoustics, Speech, and Signal Processing*, volume 5, pages 609–612, Philadelphia, USA, March 2005.
- [58] W. Zhang, Y.-J. Chang, and T. Chen. Optimal thresholding for key generation based on biometrics. In *ICIP '04: International Conference on Image Processing*, pages 3451–3454, 2004.

APPENDIX

Proof.[Theorem 1]

If only the final hash value is known to an adversary, then the first step is to approximate the secondary image J (See Figure 2). We

prove the hardness by analyzing the following equation which provides a possible approximation of the secondary images –

$$J = \sum_{i=1}^r \sqrt{\lambda_i} u_i v_i^T = \sqrt{\lambda_1} u_1 v_1^T + \underbrace{\sqrt{\lambda_2} u_2 v_2^T + \sqrt{\lambda_3} u_3 v_3^T + \dots + \sqrt{\lambda_r} u_r v_r^T}_{\text{...}}$$

where $r = 2p$; p is the number of sub-images created; and λ_i , $1 \leq i \leq r$ are non-zero eigen values of the matrix $J^T J$ such that $\lambda_1 > \lambda_2 > \dots > \lambda_r$. Note that J^T is the transpose matrix of J and a positive square root of λ_i is a singular value. The u_i 's and v_i 's, $i = [1, \dots, r]$, are eigenvectors of $J J^T$ and $J^T J$ respectively. Since the final hash value, $[u_J, v_J]$ are known to the adversary, the values which need to be guessed are λ_1 and $\{\lambda_2 u_1 v_1^T + \lambda_3 u_2 v_2^T + \dots + \lambda_r u_r v_r^T\}$. To guess λ_i 's there are infinitely many solutions as any nonnegative eigenvalues can lead to specific eigenvectors that are unitary (i.e. satisfy the definition). Any eigenvalue matrix resulting from this construction will give a solution to the equation and therefore it is computationally hard for the adversary to identify the original value.

If there is a case in which λ_1 is dominant such that the rest of the values $\lambda_2, \dots, \lambda_r$ are approximately equal to zero, then one could try to guess λ_1 and possibly approximate the secondary image by $J = \sqrt{\lambda_1} u_J v_J^T$. It is not trivial to theoretically predict the possible distribution of the values of λ_i 's because they are dependent on the type of image and the distribution of the pixel values of those images. Therefore we conducted experimental evaluation on the biometric images and found that the λ_i 's are distributed such that there is no one dominant eigenvalue because the secondary image J is a smooth image (i.e. the adjacent pixels of the image do not differ beyond a certain threshold which is determined by the algorithm parameters). We conclude that because of the hardness of guessing the eigenvalues and the lack of dominant eigenvalues the reconstruction of the secondary image J from the resultant hash vector \vec{H} is computationally hard for the biometric types considered. \square

Proof Sketch. [Theorem 2]

If J is known to the adversary, then the first step would be to form each sub-image matrix A_i , where $1 \leq i \leq p$. Note that a combination of all A_i eigenvectors were used to construct J . Each A_i is of the form $A_i = U_i S_i V_i^T$. As in the proof of Theorem 1, an infinite number of eigenvalues exist for constructing infinite A_i which would satisfy the relation. Moreover, using the same reasoning as before, there are no dominant eigenvalues as the p sub-images each of size $m \times m$ are overlapping. Because of the overlap most significant eigenvalues do not differ beyond a certain threshold as determined by the algorithm parameters p and m . In addition the largest eigenvectors (i.e. the left most and the right most vectors of the U_i and V_i matrices respectively) of each sub-image A_i are pseudorandomly combined to form J resulting in the number of choices the attacker would need to try as $p!$. This motivates the need for large values of p (~ 50). As a result guessing the order of each sub-image A_i and hence creating the original image I is computationally hard.