Topcu *et al. EURASIP Journal on Advances in Signal Processing* (2016) 2016:100
DOI 10.1186/s13634-016-0396-1

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Practical security and privacy attacks against biometric hashing using sparse recovery

Berkay Topcu[1,2*], Cagatay Karabat[1], Matin Azadmanesh[2] and Hakan Erdogan[2]

Abstract

Biometric hashing is a cancelable biometric verification method that has received research interest recently. This method can be considered as a two-factor authentication method which combines a personal password (or secret key) with a biometric to obtain a secure binary template which is used for authentication. We present novel practical security and privacy attacks against biometric hashing when the attacker is assumed to know the user's password in order to quantify the additional protection due to biometrics when the password is compromised. We present four methods that can reconstruct a biometric feature and/or the image from a hash and one method which can find the closest biometric data (i.e., face image) from a database. Two of the reconstruction methods are based on 1-bit compressed sensing signal reconstruction for which the data acquisition scenario is very similar to biometric hashing. Previous literature introduced simple attack methods, but we show that we can achieve higher level of security threats using compressed sensing recovery techniques. In addition, we present privacy attacks which reconstruct a biometric image which resembles the original image. We quantify the performance of the attacks using detection error tradeoff curves and equal error rates under advanced attack scenarios. We show that conventional biometric hashing methods suffer from high security and privacy leaks under practical attacks, and we believe more advanced hash generation methods are necessary to avoid these attacks.

Keywords: Biometric verification, Biometric hashing, Advanced attack model, Rainbow attack

1 Introduction

Biometric recognition provides an alternative to the traditional authentication mechanisms based on passwords or tokens such as ID cards due to the inalienable and distinctive nature of biometric traits. Biometric recognition systems enable fast, reliable, and secure electronic authentication; however, their large-scale deployment in real-world applications causes privacy and security concerns [1–3]. Biometric systems are not foolproof and a critical vulnerability that is unique to biometrics systems is the acquisition of the stored templates by adversaries [4]. Biometric data might reveal sensitive information such as race, gender, and certain medical conditions. Since biometric traits are supposed to be permanent and unique to an individual, stolen templates can be used as unique identifiers to link information across different applications. Moreover, biometric modalities are limited in number, and they cannot be easily revoked to obtain another template as seen in the use of passwords. Therefore, it is essential to ensure the security of biometric templates and to protect biometric data. In the literature, several biometric template protection methods have been proposed [5] (e.g., fuzzy commitment scheme [6] and biohashing [7]) to overcome these concerns by securing biometric templates (e.g., face and fingerprint). Biometric template protection methods store a modified version of the biometric template and reveal as little information about the original biometric trait as possible without losing the capability to identify a person.

*Correspondence: berkay.topcu@tubitak.gov.tr
[1]Informatics and Information Security Research Center (BILGEM), The Scientific and Technological Research Council of Turkey (TUBITAK), Gebze, 41470 Kocaeli, Turkey
[2]Faculty of Science and Natural Engineering, Sabanci University, Orhanli Tuzla, 34956 Istanbul, Turkey

1.1 Biometric template protection and biohashing

Template protection methods can be categorized into two groups: (i) biometric cryptosystems [5] (i.e., fuzzy commitment [6], fuzzy vault [8]) and (ii) transformation-based methods/salting [9] (i.e., biohashing [10]). Biometric cryptosystems either bind secrets into biometric data to form a secure biometric template or generate secrets from biometric data with the help of some auxiliary data. The secrets can be successfully retrieved during a genuine verification attempt. The helper or auxiliary data does not reveal significant information about the biometric or the key. On the other hand, transformation-based approaches distort or randomize biometric data with the use of non-invertible functions so that the original data cannot be reconstructed from transformed templates. Biometric templates are transformed based on parameters derived from external information such as user keys or passwords.

Biohashing or biometric hashing [10] is one of the transformation-based methods, in which the biometric template of the user is transformed into a protected binary string through multiplication with a pseudo-random projection matrix and quantization. Due to increased inter-class variation and preservation of intra-class variation, biohashing significantly improves verification accuracy when the secret key is kept secure and unknown to the adversaries. In this paper, we use the terms biohashing and biometric hashing synonymously, even though we think biometric hashing is a more descriptive name.

In addition to the increased performance of the protected templates when the secret key of a user is kept safe, another advantage of biometric hashing lies in the ease of revoking a transformed template by changing the associated secret key. Furthermore, using the same biometric data, a user can be authenticated to different services through different biohashes generated from distinct secret keys. This way, two records that are presented to two different systems cannot be linked and activities of the user is kept private (Fig. 1).

Biometric hashing uses a unique secret key in order to randomize biometric template of each user. It is a two-factor authentication system in which both the biometric modality and the secret key of a user have to be presented during authentication. Although biohashing methods have become very popular due to their high authentication performance and easy deployment into match-on-card applications, research recently showed that they might suffer from serious security and privacy problems [4, 11–13].

We believe that it is necessary to study the security and privacy preservation capabilities of biometric hashing especially when the secret key is compromised. If the key is always assumed to be kept secure, an authentication system which checks the accuracy of the entered key will achieve a zero verification error even without any need for biometric data.

The security performance of a biohashing scheme under the assumption of a known key is analyzed in [14, 15], and biohashing is concluded to be a good biometric randomization algorithm with a high risk of compromising the biometric information. If the secret key of a user is compromised, the security of the protected template is at stake and it is only dependent on the non-invertibility of the biohash (i.e., it should be hard for an adversary to approximate the biometric feature vector from the biohash and the secret key). The reconstruction of a sufficiently similar feature vector that provides a close biohash to the original

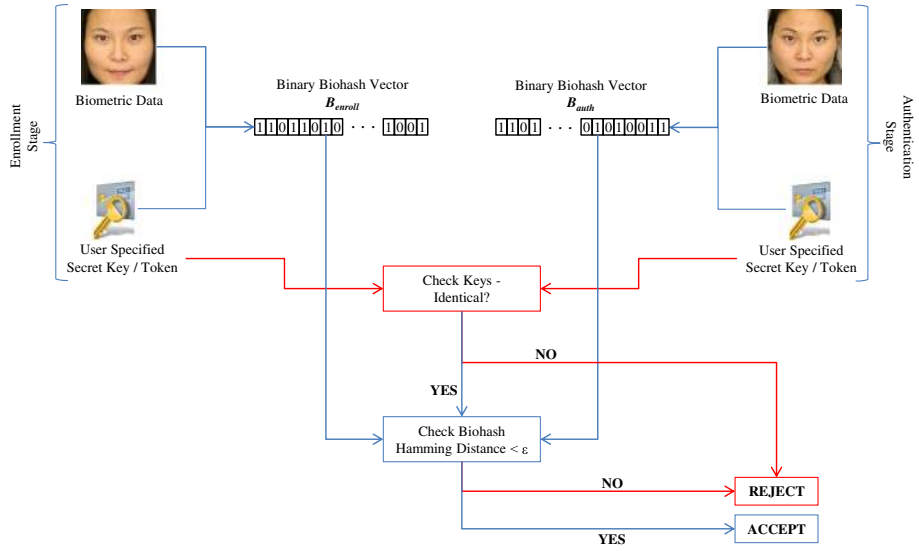

Fig. 1 Verification setup. Illustration of biometric hashing-based verification

one, called a pre-image attack (masquerade attack), is a major threat to the template protection capability of a biometric hashing scheme. It is not sufficient to make a function "lossy" (not one to one) in order to have a one-way function [16]. The biohashing method of Ngo et al. is presented as a one-way function [10]; however, we show that this is not the case (in the cryptographic sense) and biometric hashing is not pre-image attack resistant if the secret key that is used for generating a biohash is known to the adversary. Figure 2 briefly illustrates the inversion attack for biohashes. An adversary who possesses the biohash vector of a user and the corresponding secret key can invert the biohash and obtain a real-valued feature vector. This feature vector can be used to directly attack the system for unauthorized authentication. In addition, the adversary can generate an image of the biometric modality which could be used for both attacking the system and compromising the privacy of the user.

1.2 Attacks against biohashing—biohash inversion

In the first study that investigates the invertibility of a biometric hashing algorithm, it was assumed that the biohash of a user and the corresponding random projection matrix are available to an adversary. Each dimension of the biohash vector was mapped to the set $\{-1, 1\}$ (by mapping $[0] \rightarrow [-1]$ and $[1] \rightarrow [1]$) and the resulting vector was multiplied with the pseudo-inverse of the random projection matrix. A new biohash created from the estimated biometric feature vector was used to perform imposter attacks. A similar approach that uses the pseudo inverse of a random projection matrix was also presented in [17]. In [18], a new method was proposed to generate a biometric feature from biohashes using genetic algorithms. For each biohash in a database, the proposed genetic algorithm was applied to approximate the value of the biometric feature given the corresponding secret key.

A detailed analysis of irreversibility of biohashes was performed by Feng et al. [19] where the details of the random projection is solved using perceptron learning. It was assumed that the attacker does not have the secret key of the user and the parameters of the random projection are estimated using stolen biohashes and a local biometric database. The main difference of this study is that the method requires several stolen biohashes from several distinct subjects (68 subjects, 105 images/subject for one database and 350 subjects, 40 images/subject for another database) for parameter estimation. It was assumed that the whole system is available to the adversary as a black box and the matching scores could be eavesdropped. A local face dataset (3500 different local faces) was presented to the system along with a common token and every local binary template was matched against every stolen template. Using the matching scores and the stolen biohashes, local binary biohashes corresponding to the local face database were calculated, which were used for iterative perceptron learning to estimate the projection parameters. Once the parameters of the random projection were estimated, they could be used to generate synthetic real-valued features from a stolen biohash which is another perceptron problem. Our proposed methods cannot be compared with this method where the estimation of parameters with a single stolen biohash is not possible and several biohashes from different subjects are required. However, our methods require only a single biohash for the inversion.

Nagar et al. [4] have proposed a promising approach that is comparable to our proposed methods. In that approach, given the binary biohash vector of a subject and the transformation parameters, a close approximation to the original biometric features is recovered by formulating the problem as an optimization problem. A database of unrelated biometric features was used for optimization. For each unrelated biometric feature vector from the

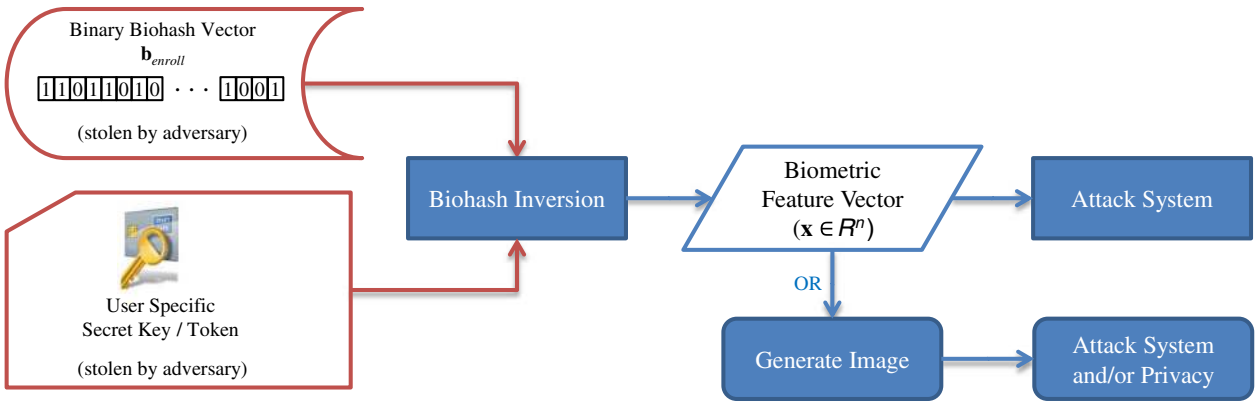

Fig. 2 Biohash inversion attacks. Illustration of the inversion attacks against biometric hashing

database, a new feature vector was estimated by minimizing the Euclidean distance between the new feature vector and the unrelated biometric feature vector subject to the consistency criterion (i.e., the new biohash created from the estimated feature vector exactly matches the original biohash). The estimated feature vector was computed by taking the weighted average of t number of trials where the weight was the Hamming distance between the original biohash and the estimated one. Since this approach attempts to invert biohashes in a similar setup with our proposed methods, we compared it with our algorithms in terms of verification errors and computation times.

1.3 Contributions of this paper

In this paper, we propose four different novel optimization-based methods that aim to predict the feature vector and/or the biometric image itself. Here, we assume that an adversary gains access to the biohash vector of a valid system user and the corresponding secret key and estimates a new real-valued feature vector from the binary biohash in order to authenticate to the system. Novel feature estimation methods are in the focus of this study. The first two proposed methods are based on 1-bit compressive sensing approach and related feature reconstruction algorithms. Compressive sensing is a new signal acquisition technology with the potential of reducing the number of measurements required to acquire signals that are sparse or compressible in some domain. Rather than uniformly sampling the signal, compressive sensing computes inner products with a randomized dictionary of test functions. The signal is then recovered by a convex optimization which ensures that the recovered signal is consistent with the measurements. One-bit measurements is a more restricted case in which only the sign information of the random measurements is preserved. In our framework, we solve the biohash invertibility problem by using two different reconstruction approaches, namely, linear programming [20] and binary iterative hard thresholding [21].

We also discuss minimum norm solutions for approximating feature vectors from biohashes and present L_2 and L_1 norm minimization for this problem. Finally, we describe the rainbow attack to compromise the security of a biometric hashing scheme. Rainbow attack is different from feature approximation methods and does not aim at predicting a new feature vector. With the help of a huge database of biometric features along with the biohash vector of a valid user and the corresponding secret key, a biometric image that creates a sufficiently close biohash to the desired one is found and used for illegitimate authentication.

We propose practical attacks and study their performances instead of using theoretical metrics. Furthermore, we analyze the privacy issues related to the invertibility of biohash templates, and as a case study, we visually inspect reconstructed face images of the subjects. Authentication performance of the reconstructed feature vectors in a conventional verification setup, in which the plain features are used for matching, is also investigated.

Our novel contributions regarding the reversibility of biohashes can be stated as follows. Practical security and privacy attacks against biohashes using 1-bit compressive sensing framework are introduced. Apart from that, minimum norm solutions are discussed in detail and L_1 norm minimization is introduced in addition to the L_2 norm minimization which appeared in the literature before. Finally, this study introduces a type of "rainbow attack" against biometric hashing systems. The differences between the existing attacks and our proposed attacks are given in Table 1 in terms of assumptions and related security and privacy issues.

First, we review the biometric hashing scheme in Section 2. The proposed feature approximation methods are presented in Section 3, which is followed by description of the rainbow attack in Section 4. Section 5 presents the experimental results of the proposed approaches and finally, we summarize our findings and conclusions in Section 6.

2 Biometric hashing

Biometric hashing schemes are simple yet powerful biometric template protection methods [22–26]. Biohash is a binary and pseudo-random representation of a biometric template (e.g., face or fingerprint), and biometric hashing schemes perform an automatic verification of a user based on her biohash which is a binary string. The two inputs to a biometric hashing scheme are (i) biometric template and (ii) user specific secret key. A biometric feature vector is transformed into another space using a pseudo-random set of vectors which are generated from the user's secret key. Then, the result is binarized to produce a pseudo-random bit string which is called the biohash. What is unique or specific to each user is the random projection matrix and it can be stored in a USB token or smartcard. In a practical system, user-specific random matrix is calculated based on a seed (user-specific secret key) that is stored in a USB token or smartcard microprocessor through a pseudo-random number generator. The seed is the same as those users recorded during the enrollment and is different among different users and different applications [7].

In an ideal case, the distance between the biohashes belonging to the biometric templates of the same user is expected to be relatively small. On the other hand, the distance between the biohashes belonging to different users is expected to be sufficiently high which enables higher recognition rates. The user is enrolled to the system at

Table 1 Existing biohash inversion attacks

Method	Assumptions	Security	Privacy
Multiply with the pseudo-inverse of the random projection matrix [17, 33]	- Random projection matrix is available - Threshold is fixed and it is 0 - Wavelet FMT face features	Attack with biohash from estimated features: - existing key - a new key is assigned and stolen again	
Genetic algorithms [18]	- Random projection matrix is available - Threshold is fixed and it is 0 - Fingercode features	1) Attack with biohash from estimated features: - existing key - a new key is assigned and stolen again 2) Average distance between real and approximated features	
Perceptron-learning with hill climbing and MLP modeling with customized hill-climbing [19]	- Several biohashes of various different subjects are available (other methods assume availability of a single stolen biohash) - Attacker can access the matching scores of the system - Secret key of the user is available	Identification scenario, where biohash generated from each synthetic face is matched against the stolen templates	Adversary has access to output of feature extractor given a face image and applies hill-climbing attack to generate synthetic face images
Solve a constrained minimization of distance between estimated features and unrelated feature vector [4]	- Random projection matrix is available - Threshold is available - A database of unrelated features - Eigenface features	Attack with biohash from estimated features: - existing key - a new key is assigned and stolen again	Reconstructed face images from estimated vector using PCA inversion
Methods proposed and discussed in this study: - Sparse recovery - Min-norm solutions	- Random projection matrix is available - Threshold is available - Eigenface features	1) Attack with biohash from estimated features: - existing key - a new key is assigned and is unknown - a new key is assigned and stolen again 2) Verification accuracy using the real features as gallery and approximated features as probe	Orthogonal linear face features (i.e., PCA, LDA): transformation matrix is known and its inverse is used to reconstruct face images

the enrollment stage. Then, the user again provides her biometric data and secret key to the system at the authentication stage in order to prove her identity.

In the below subsection, we describe the random projection (RP) based biohashing scheme proposed by Ngo et al. [10] for face verification.

2.1 Enrollment stage

Feature extraction At this phase, face images in the training set, which are collected during the enrollment stage, are used. The set has training face images belonging to the registered users, $\mathbf{I}_{i,j} \in \Re^{m \times n}$ where $i = 1, \ldots, K$ and K denotes number of users, $j = 1, \ldots, L$ and L denotes number of training images per user. Each face image is represented as a vector, $\mathbf{y} \in \Re^{(mn)\times 1}$. Then, principle component analysis (PCA) [27] is applied to the face images in the training set for feature extraction:

$$\mathbf{x} = \mathbf{A}(\mathbf{y} - \mu), \tag{1}$$

where $\mathbf{A} \in \Re^{k \times (mn)}$ is the PCA matrix trained by the face images in the training set, μ is the mean face vector, and $\mathbf{x} \in \Re^{k \times 1}$ is the vector containing PCA coefficients.

Random projection At this phase, a pseudo-random projection (RP) matrix, $\mathbf{R} \in \Re^{\ell \times k}$, is generated to transform the PCA coefficient vectors. The RP matrix elements are independent and identically distributed (*i.i.d*) and generated from a Gaussian distribution with zero mean and unit variance by using a pseudo-random number generator (PRNG) with a seed derived from the user's secret key. The RP matrix projects the PCA coefficients onto an ℓ-dimensional space:

$$\mathbf{z} = \mathbf{R}\mathbf{x}, \tag{2}$$

where $\mathbf{z} \in \Re^{\ell \times 1}$ is an intermediate biohash vector.

Quantization At this phase, the elements of the intermediate biohash vector $\mathbf{z}$ are binarized with respect to a threshold:

$$\mathbf{b}(k) = \begin{cases} 1, & \mathbf{z}(k) \geq \beta \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $\mathbf{b} \in \{0,1\}^{\ell}$ denotes the biohash vector of the user and β denotes the quantization threshold which can be 0 (sign operator) or mean value of the intermediate biohash vector $\mathbf{z}$, depending on the system design.

After enrollment, biometric hashes are stored in a database or in a smart card.

2.2 Authentication stage

At this stage, a claimer sends his face image $\tilde{\mathbf{I}} \in \Re^{m \times n}$ and his secret key to the system. Then, the system computes the claimer's test biometric hash vector by using the same procedures in the enrollment phase. The user is authenticated when the Hamming distance between $\mathbf{b}_{\text{enroll}}$ (which denotes the biohash of the user generated at the enrollment stage) and $\mathbf{b}_{\text{auth}}$ (which denotes the biohash of the user generated at the authentication stage) is below a pre-determined distance threshold ϵ as follows:

$$\sum_{k=1}^{n} \mathbf{b}_{\text{enroll}}(k) \oplus \mathbf{b}_{\text{auth}}(k) \leq \epsilon \tag{4}$$

where $\oplus$ denotes the binary XOR (exclusive OR) operator. The distance threshold ϵ is an integer between 0 and n (number of bits in a biohash). In a biometric hashing system, the selection of ϵ depends on system design, and it is chosen such that the desired false acceptance rate (FAR) and false rejection rate (FRR) are satisfied.

The system computes the Hamming distance between the test biometric hash vector and the claimed user's reference biometric hash vector stored in the database. If the Hamming distance is below the pre-determined distance threshold, the claimer is accepted; otherwise, the claimer is rejected (Fig. 1).

3 The proposed feature approximation methods from biohash

In this section, we introduce intrusion attacks via reconstruction of the biometric feature vector from biohashes. In this context, intrusion is defined as gaining access to a biometric recognition system by presenting falsified authentication data to the system [4]. We use the following notation throughout our analysis of biometric hashing scheme: $\mathbf{b}$ represents the biohash vector of a valid system user and it is obtained by an adversary to perform intrusion attacks through feature approximation, $\mathbf{R}$ is the user specific random projection matrix and known to the adversary, $\mathbf{x}$ is the original biometric feature vector that $\mathbf{b}$ is created from and it is neither known nor accessible by the attacker, and $\hat{\mathbf{x}}$ is the feature vector (or pre-image) that is approximated through inversion of $\mathbf{b}$. Note that, $\hat{\mathbf{x}}$ does not necessarily correspond to a valid biometric feature vector (i.e., PCA coefficients for faces or fingerprint minutiae information). However, using $\hat{\mathbf{x}}$, one can produce a biohash vector that allows unauthorized access to the biometric system (Fig. 3). Once $\hat{\mathbf{x}}$ is obtained, an attacker might also reconstruct the biometric modality and use it for illegitimate access to a system, i.e., in our case, this is the face image (it is also assumed that the attacker knows the PCA matrix used in feature extraction). In this study, we consider that the intrusion to the system can happen in two ways before the random projection step. An attacker either provides a digital face image (reconstructed face image) to the system prior to the feature extraction step or uses the approximated feature vector as input to the random projection.

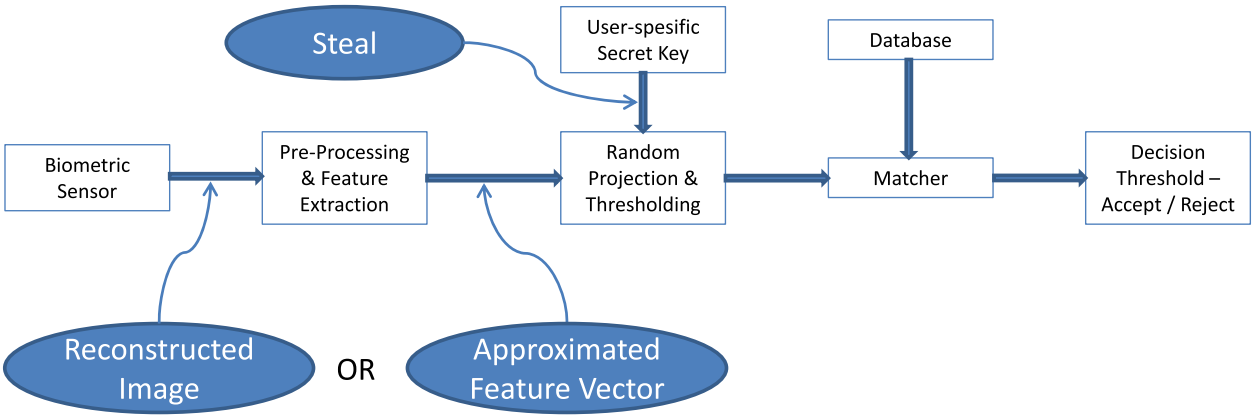

Fig. 3 Attacks. Illustration of the proposed attack

The success probability of such an attack to the biometric hashing system can be measured as $P(d(\text{sign}(\mathbf{R}\hat{\mathbf{x}}), \mathbf{b}) < \epsilon)$[1], where $d(\cdot)$ is the Hamming distance between two biohashes (i.e., the number of disagreeing bits). This metric is also called the intrusion rate due to inversion for the same biometric system (IRIS) by Nagar et al. [4]. In the next sections, we present various methods to obtain a feature vector $\hat{\mathbf{x}}$ that allows illegitimate access to a biometric system given $\mathbf{b}$ and the transformation parameters.

3.1 One-bit compressive sensing approach

One-bit compressive sensing studies efficient acquisition of sparse (or more structured) signals via linear measurement systems, and only 1-bit per measurement is retained. While the key application of this problem has been in the area of signal acquisition, it has also found applications in several learning related problems. Boufounos et al. [28] introduced the problem of 1-bit compressive sensing where only 1 bit of the linear measurement, specifically its sign, is observed. Random projection-based biometric hashing can be viewed in the same context as 1-bit compressive sensing. If the threshold used in quantization of the projected signal is 0 (such that the sign of the signal is kept), each bit of a biohash is the sign of the inner product of the feature vector ($\mathbf{x}$) with a measurement vector (in biometric hashing, each row of the random projection matrix ($\mathbf{R}$):

$$b_i = \text{sign}(\langle R_i, \mathbf{x} \rangle). \tag{5}$$

The biometric hashing procedure is compactly expressed using:

$$\mathbf{b} = \text{sign}(\mathbf{R}\mathbf{x}), \tag{6}$$

where $\mathbf{b}$ is the biohash vector, $\mathbf{R}$ is the matrix representing the random projection matrix (the measurement system), and the 1-bit quantization function sign(.) is applied element-wise to the vector $\mathbf{Rx}$.

For consistent reconstruction from 1-bit measurements, the measurements are treated as sign constraints that are enforced in the reconstruction to recover the signal. In the reconstruction, L_1 norm of the feature vector is minimized to obtain a sparse solution. When the PCA coefficients of face images are analyzed, it is noted that most of the coefficients are small in magnitude and only about 25 % of them is enough to obtain $\sim$ 70 % of the total energy as seen in Fig. 4. Therefore, it is reasonable to assume that PCA vectors are sparse. Also, as stated by Candes and Wakin [29], "compressive sampling exploits the fact that many natural signals are sparse in the sense that they have concise representations when expressed in the proper basis." Even if the original signal is not sparse, a basis can be found in which most coefficients are small, and the relatively few large coefficients capture most of the information and this allows for the use of sparse recovery in the problem of biohash inversion.

In addition, to enforce reconstruction at a non-trivial solution, one needs to artificially resolve the amplitude ambiguity. Thus, an energy constraint is imposed that the reconstructed signal lies on the unit L_2-sphere:

$$\|\mathbf{x}\|_2 = \left(\sum_i x_i^2 \right)^{1/2} = 1. \tag{7}$$

The signal on the unit sphere that is consistent with the measurements is found by solving:

$$\begin{aligned} &\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \\ &\text{s.t. } \text{sign}(\mathbf{R}\hat{\mathbf{x}}) \equiv \mathbf{b} \\ &\text{and } \|\hat{\mathbf{x}}\|_2 = 1. \end{aligned} \tag{8}$$

As the compressive sensing measurements are quantized to 1 bit, it is clear that the scale (absolute amplitude) of the signal is lost and it is not immediately evident that

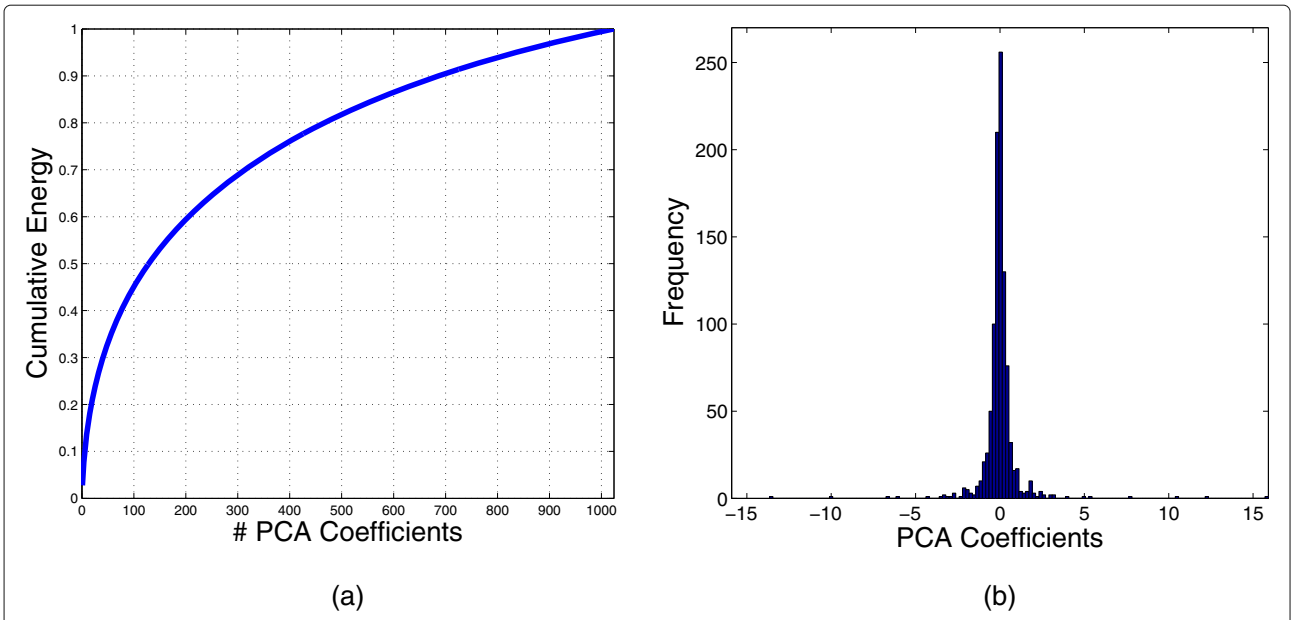

Fig. 4 PCA. **a** Cumulative energy contained in PCA coefficients. **b** Distribution of 1024 dimensional PCA coefficients of a sample from the database

the remaining information is enough for signal reconstruction. Nonetheless, there is a strong empirical evidence stating that signal reconstruction is possible [28]. One-bit compressive sensing by linear programming [20] and binary iterative hard thresholding [21] are two theoretical reconstruction methods that we implement separately for obtaining inverse images of biohashes and finding biometric feature vectors that provide biohash vectors which are acceptable by the verification system (i.e., with a distance to the original biohash vector that is less than a threshold).

3.1.1 One-bit compressive sensing by linear programming

The study in [20] has showed that $\mathbf{x}$ can be accurately estimated from extremely quantized measurement vector in (6). Note that $\mathbf{b}$ contains no information about the magnitude of $\mathbf{x}$ and only the normalized vector $\mathbf{x}/\|\mathbf{x}\|_2$ can be recovered. It has been shown that the signal can be accurately recovered by solving the following convex minimization program:

$$\begin{aligned} &\min \|\hat{\mathbf{x}}\|_1 \\ &\text{s.t. } \mathbf{B}\mathbf{R}\hat{\mathbf{x}} \geq \mathbf{0} \\ &\text{and } \|\mathbf{R}\hat{\mathbf{x}}\|_1 = m, \end{aligned} \tag{9}$$

where $\mathbf{B} = \text{diag}(\mathbf{b})$.

The first constraint, $\mathbf{B}\mathbf{R}\hat{\mathbf{x}} \geq \mathbf{0}$, keeps the solution consistent with the original biohash vector and it is defined by the relation $\langle R_i, \hat{\mathbf{x}} \rangle \cdot b_i \geq 0$ for $i = 1, 2, \ldots, m$ where R_i is the i^{th} row of the random projection matrix $\mathbf{R}$. The second constraint in the original problem definition (8) contains L_2-norm which is a quadratic term and can be replaced with the linear L_1-norm, so that the optimization becomes a linear program. The second constraint, $\|\mathbf{R}\hat{\mathbf{x}}\|_1 = m$, serves to prevent the program from returning zero solution, and it is linear as it can be represented as one linear equation $\sum_{i=1}^{m} b_i \langle R_i, \hat{\mathbf{x}} \rangle = m$, where m is the length of the biohash vector. Therefore, (9) is a convex minimization problem and can easily be represented as a linear program (see Algorithm 1).

Algorithm 1 Approximate biometric feature vector $\hat{\mathbf{x}}$ using Linear Programming

Input: $\mathbf{b}$, $\mathbf{R}$
Output: $\hat{\mathbf{x}}$
calculate $\mathbf{A}$ such that $\mathbf{A}\hat{\mathbf{x}} \geq \mathbf{0}$ using $\mathbf{b}$ and $\mathbf{R}$
calculate $\mathbf{A}_{eq}$ such that $\mathbf{A}_{eq}\hat{\mathbf{x}} = m$ using $\mathbf{R}$
set f to calculate L_1 norm of $\hat{\mathbf{x}}$
use simplex method to solve for $\hat{\mathbf{x}}$

3.1.2 Binary iterative hard thresholding

Binary iterative hard thresholding (BIHT) [21] is a modification of iterative hard thresholding (IHT) which is a real-valued algorithm designed for compressive sensing [30]. Proposed for the recovery of K-sparse signals, IHT algorithm consists of two steps. The first step is a gradient descent to reduce the least squares objective $\|\mathbf{y} - \mathbf{R}\mathbf{x}\|_2^2/2$. At each iteration, IHT proceeds by setting $\mathbf{a}^{l+1} = \mathbf{x}^l + \mathbf{R}^T(\mathbf{y} - \mathbf{R}\mathbf{x})$. The second step imposes a sparse signal model by selecting the K elements of $\mathbf{a}^{l+1}$ that are largest in magnitude.

BIHT algorithm modifies the first step of IHT and minimizes a consistency-enforcing objective. Given an initial estimate $\mathbf{x}^0 = \mathbf{0}$ and 1-bit measurements $\mathbf{b}$, at each iteration l, BIHT computes:

$$\mathbf{a}^{l+1} = \mathbf{x}^l + \frac{\tau}{2}\mathbf{R}^T\left(\mathbf{b} - \text{sign}\left(\mathbf{R}\mathbf{x}^l\right)\right) \qquad (10)$$
$$\mathbf{x}^{l+1} = \eta_K\left(\mathbf{a}^{l+1}\right),$$

where τ is a scalar that controls the gradient descent step size, and the function η_K computes the best K-term approximation of $\mathbf{a}^{l+1}$ (see Algorithm 2). In our experiments, we choose K as 25 % of the feature vector length, i.e., $K = 50$ for 200 dimensional feature vectors and $K = 256$ for 1024 dimensional feature vectors. As stated in Section 3.1, 25 % of the PCA coefficients captures 70 % of the total energy. In addition, we analyzed PCA coefficients of natural face images (see Fig. 4) and concluded that 25 % of the coefficients that are largest in magnitude are enough to reconstruct a typical face image that is visually similar to the original face image.

Algorithm 2 Approximate biometric feature vector $\hat{\mathbf{x}}$ using BIHT

Input: $\mathbf{b}$, $\mathbf{R}$, K
Output: $\hat{\mathbf{x}}$
initialize $\mathbf{x}^0$ all zeros
while $|\mathbf{b} - \text{sign}(\mathbf{R}\mathbf{x}^l)|_1 > 0$ **do**
$\mathbf{a}^{l+1} = \mathbf{x}^l + \frac{\tau}{2}\mathbf{R}^T(\mathbf{b} - \text{sign}(\mathbf{R}\mathbf{x}^l))$
sort elements of $\mathbf{a}^{l+1}$ and set the all but the largest K components to 0,
end while
set $\hat{\mathbf{x}} \leftarrow \mathbf{a}^{l+1}$

3.2 Minimum L_1 and L_2 norm solutions

In this section, we present and discuss minimum norm-based feature reconstruction methods for biohashes in addition to the solutions we propose in 1-bit compressive sensing framework.

Biohash vector is obtained through quantization from an intermediate vector $\mathbf{z}$ which is the output of a random projection, i.e., $\mathbf{z} = \mathbf{R}\mathbf{x}$. If one can estimate the quantization step and find an intermediate vector $\hat{\mathbf{z}}$ by inverting the biohash vector, a minimum norm solution can be used to estimate the biometric feature vector ($\hat{\mathbf{x}}$) as:

$$\min \|\hat{\mathbf{x}}\|_n \text{ s.t. } \hat{\mathbf{z}} = \mathbf{R}\hat{\mathbf{x}}. \qquad (11)$$

In this work, we study minimum norm solutions for $n = 1$ and $n = 2$, namely L_1 and L_2 norms.

3.2.1 Inversion of the quantization step

Solutions in a 1-bit compressive sensing framework implicitly handle the quantization of the randomly projected feature $\mathbf{z}$ within the optimization process. However, L_1 and L_2 norm-based reconstruction requires an explicit inversion of the thresholding step of the biometric hashing scheme.

In order to invert the quantization process, an adversary who possesses the biohash ($\mathbf{b}$) of a valid system user and corresponding random projection matrix ($\mathbf{R}$) uses an arbitrary biometric feature vector $\mathbf{x}_f$ to simulate the biometric hashing procedure through random projection and obtain an intermediate vector $\mathbf{z}_f = \mathbf{R}\mathbf{x}_f$. Next, the sample mean and standard deviation of $\mathbf{z}_f$ are calculated, μ and σ, respectively. Mapping the elements of the compromised biohash vector $\mathbf{b}$ from {0,1} to $\{-1, 1\}$ is performed as:

$$\hat{b}(i) = \begin{cases} 1, & b(i) = 1, \\ -1, & b(i) = 0, \end{cases} \qquad (12)$$

where $\hat{\mathbf{b}}$ is the mapped biohash vector. Finally, using the values calculated from the arbitrary biometric features, the intermediate vector $\hat{\mathbf{z}}$ is estimated as:

$$\hat{z}(i) = \mu + \hat{b}(i)\sigma. \qquad (13)$$

To be consistent with the solutions described in a 1-bit compressive sensing approach, we assume that the signs of the elements of the intermediate vector $\mathbf{z}$ is used to obtain the biohash (i.e., the threshold at the quantization step is 0). However, various quantization methods and thresholding mechanisms are proposed in the literature for biometric hashing, one of them being the mean value of the intermediate vector and another one being its median value. If the system uses the mean value of the intermediate vector as the quantization threshold, the mean value of the $\mathbf{z}_f$ can be calculated. In our experiments, the threshold equals to 0; thus, the mean value is not used, and the intermediate vector is computed as $\hat{z}(i) = \hat{b}(i)\sigma$.

3.2.2 Minimum L_2 norm solution

Once an adversary creates an intermediate vector $\hat{\mathbf{z}}$, the following L_2 norm minimization provides an estimate feature vector $\hat{\mathbf{x}}$ that is consistent with the observation $\mathbf{b} = \text{sign}(\mathbf{R}\hat{\mathbf{x}})$.

$$\min \|\hat{\mathbf{x}}\|_2 \text{ s.t. } \hat{\mathbf{z}} = \mathbf{R}\hat{\mathbf{x}}. \qquad (14)$$

The closed form solution that gives the minimum L_2 norm for the estimated feature vector is given by the MoorePenrose pseudo-inverse. For linear systems $\mathbf{A}\mathbf{x} = \mathbf{b}$ with non-unique solutions (i.e., under-determined systems), the pseudo inverse is used to reconstruct the solution of minimum Euclidean norm $\|\mathbf{x}\|_2$ among all

solutions. So the solution to the above minimization problem to estimate the feature vector from biohash $\mathbf{b}$ is calculated as $\hat{\mathbf{x}} = \mathbf{R}^{\dagger}\hat{\mathbf{z}}$.

3.2.3 Minimum L_1 norm solution

Similar to the minimum L_2 norm solution, minimum L_1 norm solution aims at solving the following minimization problem.

$$\min \|\hat{\mathbf{x}}\|_1 \text{ s.t. } \hat{\mathbf{z}} = \mathbf{R}\hat{\mathbf{x}}. \tag{15}$$

In a 1-bit compressive sensing approach by linear programming, L_1 norm of the estimated feature vector is minimized according to the constraints that include the quantization step. However, minimum L_1 norm solution handles the quantization step separately, and the minimization is carried out over the intermediate real-valued vector $\hat{\mathbf{z}}$. The minimization problem still has linear constraints and minimization of L_1 norm can easily be expressed as a linear program and solved accordingly.

For both L_1 and L_2 norm minimizations, if the PCA dimension is less than the biohash length (i.e., if the random projection step does not reduce the dimension), the linear system is over-determined and an exact solution might not possibly exist (i.e., solutions could be inconsistent with the observations). Instead, it is possible to minimize the residual between the observation and the solution (i.e., $\|\hat{\mathbf{z}} - \mathbf{R}\hat{\mathbf{x}}\|_n$) and to obtain a feature vector that provides biohashes that is close to the original one.

3.3 Reconstructing the face image

As long as the feature extraction step uses an orthogonal transformation matrix, it is possible to invert the feature extraction process simply by using the pseudo inverse of the transformation matrix and a face image can be reconstructed easily. The principal component analysis uses an orthogonal transformation, which means that the columns of the PCA matrix are perpendicular to each other and hence one can reconstruct the face image $\hat{\mathbf{y}}$ from $\hat{\mathbf{x}}$ by using the property of an orthogonal matrix $\mathbf{A}^{\dagger} = \mathbf{A}^{T}$:

$$\hat{\mathbf{y}} = \mathbf{A}^{T}\hat{\mathbf{x}} + \mu, \tag{16}$$

where $\mathbf{A} \in \Re^{k\times(mn)}$ is the PCA matrix, $\mathbf{A}^{\dagger}$ is the pseudo-inverse of $\mathbf{A}$, and μ is the mean face vector.

3.4 Other thresholding methods—apart from the "sign" operator

In cases where the thresholding after the random projection step is not the sign operator, some alternatives can also be formulated within our proposed framework. Assuming that an adversary has the full knowledge of the system, i.e., the specific thresholding method, he can also invert the biohashes.

3.4.1 Fixed or user-specific threshold

Apart from using the sign operator, one can use a predefined fixed threshold or user specific threshold, i.e., $\mathbf{b} = \text{sign}(\mathbf{Rx} - \mathbf{T})$ where $\mathbf{T}$ denotes the threshold. Entries of $\mathbf{T}$ can be the same number or different numbers at each dimension. $\mathbf{T}$ can also be specific to each user (it is shown as $\mathbf{T}_i$ where i denotes the subject number). By augmenting the threshold vector to the random projection matrix, $\hat{\mathbf{R}} = \begin{bmatrix} \mathbf{R} & -\mathbf{T}_i \end{bmatrix}$, we can reformulate the biohashing operation as $\mathbf{b} = \text{sign}\left(\hat{\mathbf{R}}\begin{bmatrix} \mathbf{x} \, 1 \end{bmatrix}\right)$ and perform the same operations for inverting biohashes.

3.4.2 Mean value is the threshold

An alternative way of thresholding the intermediate vector is to use the mean value of the intermediate biohash vector $\mathbf{z} = \mathbf{Rx}$ as the threshold and to calculate the biohash vector as

$$\mathbf{b} = \text{sign}(\mathbf{Rx} - \text{mean}(\mathbf{Rx})). \tag{17}$$

Thresholding step can be integrated into the random projection step by using the modified random projection matrix $\hat{\mathbf{R}}$:

$$\hat{\mathbf{R}} = \left[\mathbf{R} - \frac{\underline{\mathbf{1}} \cdot \mathbf{R}}{N}\right], \tag{18}$$

where N is the biohash length, $\underline{\mathbf{1}}$ is a matrix of ones, and the biohash vector becomes $\mathbf{b} = \text{sign}(\hat{\mathbf{R}}\mathbf{x})$. An adversary can use the modified matrix $\hat{\mathbf{R}}$ and all inversion methods that we discuss are still valid in this setup.

4 Rainbow attack

In the previous section, we propose four different optimization methods for recovering features from an original biohash vector that is stolen by an attacker. Having the corresponding secret key and using the knowledge of system parameters, one can estimate a real-valued feature vector $\hat{\mathbf{x}}$ with the consistency criterion such that $\mathbf{b} = \text{sign}(\mathbf{R}\hat{\mathbf{x}})$ in order to gain illegitimate access to the biometric system. Rainbow attack is different from these methods in the sense that it does not aim at inverting a biohash vector to obtain a valid pre-image. Instead, using the knowledge of the system and the secret key of the user, with the help of a large database of biometric features, an adversary may find a face image which, when combined with the secret key of the user, result in a biohash vector that is sufficiently close to the original biohash $\mathbf{b}$.

In the cryptography literature, a rainbow table is a precomputed table for reversing cryptographic hash functions, usually for cracking password hashes. Any computer system that requires password authentication must contain a database of passwords, either hashed or in plaintext, and utilize different methods to store passwords.

Because the tables are vulnerable to thefts, storing passwords as plain texts is dangerous. Most databases therefore store a cryptographic hash of a user's password in the database. When a user enters his password for authentication, it is hashed and compared to the stored password entry of that user (which is also hashed before being stored in the database). If the two hashes match, the access is granted. A rainbow table is a large dictionary with precalculated hashes and the passwords from which they were calculated. When an attacker steals a long list of password hashes from the system, he can quickly check if any of them are in the rainbow table. If that is the case, the rainbow table will also contain the original string that they were hashed from.

A biometric authentication system that protects biometric templates using biometric hashing methods operates in a similar way; the biohash of a user is stored and compared to the query biohash during verification. If an adversary having a large database of biometric features of various users steals the biohash of a system user and knows his secret key, the adversary can compute biohashes of each biometric feature in the database using the random projection matrix of the user and create a table of biohashes and their corresponding feature vectors. If any of the biohashes in the table is sufficiently close to the stolen biohash (i.e., their Hamming distance is less than a threshold), the corresponding feature vector can be used for illegitimate access to the biometric system.

Different from previously described attacks which try to approximate a feature vector that gives a close biohash vector to the stolen one, the rainbow attack is a practical attack that aims to compromise the security of a biometric hashing scheme. Furthermore, assuming that one authentication factor (the secret key of a user) is known, the rainbow attack also provides privacy threat since look alike faces can be found.

5 Experiments and results

In this section, the performance of our proposed attack methods are analyzed and discussed. The database that is used and the experimental setup are described, and attack models and their corresponding error rates are given.

5.1 Database and experimental setup

In order to provide the performance analysis of the security of biohashes based on the feature approximation methods, we implement our proposed methods on a face verification setup.

We have obtained face verification results on BioSecure-ds2 [31] face database. Faces are detected in an automatic fashion using Viola-Jones face detector [32], and detected face images are resized to 64×64 pixels. In order to normalize a gray-scale face image, its mean intensity value is extracted from each pixel and each pixel is divided to its standard deviation. The resulting face images have zero mean and unit variance.

The BioSecure-ds2 face database consists of 210 users, equally balanced in female and men. There are two sessions for each person. For each person and for each session, there exist six colored images (two webcam acquisitions and four standard camera acquisitions—two with flash and two without flash). Standard camera acquisitions of 210 users, 8 images per person, are used in our experiments.

M-dimensional PCA coefficients are calculated for all 8 samples of 210 subjects (a total of 1680 (210×8) face images are used in our experiments). Two different PCA dimensions ($M = 200$ and $M = 1024$) are used in this study. $M = 200$ is a typical choice for PCA dimension of face images. We also analyze $M = 1024$ in order to see what extent the increased feature dimension affects inversion process. PCA training is done using the first session images only. Applying standard biometric hashing procedure, a bit string is created by inner product between the pseudo-random number and M-dimensional PCA coefficients and deciding each bit based on the sign of the each vector entry. Using random projection matrices of different sizes, one can obtain bit strings of various lengths. We present our results using bit strings of lengths 128, 256, 512 and 1024, in order to analyze how the proposed methods perform for different biohash lengths.

In a verification setting, we use all possible combinations for matching genuine pairs and the first sample of each subject is chosen for imposter matches (5880 ($210 \times 8 \times 7/2$) genuine comparisons and 21945 ($210 \times 209/2$) imposter comparisons) in order to evaluate the performance of the biometric hashing scheme. For validating the consistency of approximated features using the proposed methods, we compare the biohashes created from these features with the original biohashes leading to one imposter score for each sample in the database (1680 imposter matches). Equal error rates (EER) in each case are reported.

5.2 Performance of the biometric hashing scheme

First, we apply the general biometric hashing scheme described in Section 2 on the BioSecure-ds2 face database. For comparison, we also include face verification results of PCA vectors by using Euclidean distance as the matching method. The equal error rates for this method before applying biometric hashing to PCA vectors are 11.893 and 12.482 % for vectors of length 200 and 1024, respectively. Equal error rates for biohash vectors of various lengths are given in Table 2.

For all bit lengths, the performance of the biometric hashing scheme is better than the baseline PCA approach and lower EERs are obtained with the protected templates. In cases where an adversary steals the secret key of a

Table 2 Equal error rates (%) for biohash vectors of different lengths

	PCA 200		PCA 1024	
Bit length	Biohash	Biohash (stolen key)	Biohash	Biohash (stolen key)
128	6.295	12.571	6.593	13.565
256	4.570	11.457	4.813	12.216
512	4.137	11.595	4.328	11.634
1024	2.875	11.118	2.934	11.553

user but does not possess the claimed person's biometric information, the adversary sends his own biometric (or an arbitrary biometric) and the secret key of the genuine user in order to be authenticated. This is a serious threat to the system as the pseudo-random vectors generated using the secret key have a considerable influence on the generated bit string, therefore, on the matching score. However, even if the attacker knows the secret key, the verification accuracy of the biometric hashing system is still in the same range with the performance of the unprotected PCA vectors.

One obvious addition to the biometric hashing scheme is the direct comparison of secret keys (i.e., the one stored during enrollment and the one presented during authentication) prior to biohash comparison. This way 0 % (zero) EER is achieved if the attacker does not have the secret key of a valid user. The error rates presented in Table 2 are the results of biohash comparison, and if key checking mechanism is applied as illustrated in Fig. 1, the EERs for the first scenario would be 0 %. So that, here, we study the added security coming from the biometrics with the use of biohashes in cases where an attacker obtains the secret key.

5.3 Performance of the feature approximation from biohash methods

Since the database that we use has 1680 samples from 210 subjects, using their PCA coefficients and secret keys of each subject, we create 1680 biohashes, each corresponding to a different sample. It is assumed that an adversary obtains the biohash and the secret key of a user and with this knowledge he aims to find a feature vector by inverting the biohash. With this new feature vector, a new biohash can be calculated and used for authentication purposes. For each biohash in the database, we obtain a new feature vector and create its corresponding biohash. We use the new biohash to perform an imposter attack to the original one and we do not attack to other genuine samples. We use all possible combinations to match genuine pairs (5880 ($210 \times 8 \times 7/2$)), and the number of imposter comparisons is 1680 (one for each biohash). The performance of each method is reported in terms of the equal error rate (EER), and higher EER shows the success of the attacker (i.e., 100 % EER means that the inversion of all biohashes in the database is successful and the approximated features provide biohash that matches with the original one).

In order to evaluate the security that biometric hashing provides, we follow three consecutive scenarios:

Advanced attack model (AAM): The attacker, who knows the system details and possesses the biohash of a user and his secret key, calculates an estimate feature vector. Using this feature vector and the secret key of the subject, a new biohash is created and compared with the original one.

Security after key change (SAKC): Upon the detection of a security breach, the secret key of the user is changed by the system administrator. Using the previous biometric data, a new biohash is created from the new secret key and stored as the new gallery template in the system. The adversary does not have access to neither the new secret key nor the new biohash. The adversary makes an authentication attempt using the feature vector found in the advanced attack model and the previous (or an arbitrary) secret key. It should be noted that these errors are available only when the system does not perform key checking prior to biohash comparison. As the attacker does not know the secret key of the user, the EER in a key-checking system is 0 %.

However, for the sake of completeness, a no key-checking system is also considered and EERs in this case are also reported. EERs presented in Table 3 correspond to the attack in which the adversary has the true (original) biometric features but does not possess the associated secret key. These numbers provide a lower bound on the long-term security error, where the secret key of the user is changed and is not known to the attacker.

Attack in the long term (ALT): The adversary obtains the new secret key of the user but not the new biohash.

Table 3 Equal error rates (%) when the adversary has the true biometric features but does not possess the associated secret key

PCA dimension	Biohash length			
	128	256	512	1024
200	6.199	4.290	4.243	2.917
1024	6.497	4.902	4.375	3.044

Using the feature vector found in the advanced attack model and the new secret key, the adversary makes an authentication attempt. This is different from the advanced attack model in the sense that the biohash vector of the user is not known to the adversary and the authentication attempt is performed using the approximated feature vector which is obtained from the previous biohash of the user.

5.3.1 Results for 1-bit compressive sensing approaches

We use two different feature approximation methods, namely linear programming (LP) and binary iterative hard thresholding (BIHT), in the 1-bit compressive sensing framework. The success rates of both methods are presented in Tables 4 and 5. For the advanced attack model, the number of exact reconstructions, i.e., the number of estimated features that provide the exact same biohashes (such that the Hamming distance between the original biohash and the forged biohash is 0), is 1680 for all bit lengths. For every sample in the database, regardless of the PCA dimension, both methods are able to find a feature vector that provides the exact same biohash and that is also reflected by 100 % EERs.

In the security after key change scenario, when the secret key of the user is changed but not known to the adversary, EERs are in the same line with the cases where the adversary has access only to one of the factors, either true biometric or true secret key (see Tables 2 and 3). In the attack in the long-term (ALT) scenario, it is possible for the attacker to have unauthorized access to the system most of the time, especially if the PCA length is shorter and the biohash length is longer (see the ALT column in Tables 4 and 5).

Boundary conditions are issues of LP implementation, i.e., small $\mathbf{Rx}$ values before thresholding (for sign operator, values are close to zero). This leads to numerical inconsistencies about the inequality criteria of the linear program (i.e., $\mathbf{BRx} \geq 0$) and can be solved by replacing the inequality constraint with $\mathbf{BRx} \geq \epsilon$, where ϵ is the minimum positive number available in MATLAB (machine epsilon).

Table 5 Equal error rates (%) for 1-bit compressive sensing approaches—BIHT method

PCA	Bit length	AAM	SAKC	ALT
200	128	100.00	7.381	33.767
	256	100.00	4.851	49.388
	512	100.00	3.958	74.809
	1024	100.00	3.367	90.536
1024	128	100.00	6.667	16.314
	256	100.00	5.306	19.887
	512	100.00	4.252	28.759
	1024	100.00	3.474	47.653

5.3.2 Results for minimum norm solutions

The same set of experiments on the invertibility of biohashes is conducted using the proposed minimum norm solutions (see Tables 6 and 7). For biohashes created from PCA vector of length 1024, both methods are able to find a pre-hash vector that can be used to create the same biohash for each sample in the database. As in the 1-bit compressive sensing approach, the number of exact reconstructions is also 1680 in this case. However, when less number of PCA coefficients are used in the system (i.e., the PCA feature vectors are 200 dimensional), there is a slight decrease in the equal error rates. Biohashes created from the estimated feature vectors are not exactly same with the original ones (i.e., the Hamming distance between them is greater than zero) which is reflected by the slight deviation from 100 % EER.

In the SAKC scenario, the performances of minimum norm solutions are similar to the 1-bit compressive sensing solutions. If the new key of the user is stolen (the ALT scenario), 1-bit compressive sensing approaches provide significantly higher error rates which shows the success of the attack method.

Table 4 Equal error rates (%) for 1-bit compressive sensing approaches—linear programming (LP) method

PCA	Bit length	AAM	SAKC	ALT
200	128	100.00	7.262	48.333
	256	100.00	5.225	65.570
	512	100.00	4.018	78.958
	1024	100.00	3.308	89.987
1024	128	100.00	7.530	40.187
	256	100.00	5.128	53.342
	512	100.00	4.286	68.907
	1024	100.00	3.444	80.863

Table 6 Equal error rates (%) for minimum norm solutions—L_2 norm

PCA	Bit length	AAM	SAKC	ALT
200	128	100.00	7.113	31.233
	256	99.843	5.196	34.753
	512	99.239	4.018	72.513
	1024	98.444	3.219	86.599
1024	128	100.00	7.117	17.623
	256	100.00	5.544	21.003
	512	100.00	4.256	28.703
	1024	100.00	3.474	36.947

Table 7 Equal error rates (%) for minimum norm solutions—L_1 norm

PCA	Bit length	AAM	SAKC	ALT
200	128	100.00	6.815	30.965
	256	97.113	5.106	28.563
	512	92.491	3.839	61.173
	1024	92.751	3.431	77.564
1024	128	100.00	6.577	17.534
	256	100.00	5.723	20.765
	512	100.00	4.196	28.346
	1024	100.00	3.474	36.947

Figure 5 illustrates the detection error tradeoff curves for the attacks using the proposed methods under the ALT scenario (together with the results of the the study in [4]). Table 8 shows the corresponding FAR1000 values (false reject rates when the FAR $= 10^{-3}$). The attack performance of the reconstructed feature vectors from biohashes of 1024-bits can be compared among different methods. For brevity, we do not include all results for different biohash lengths.

A special case of solving the norm minimization problem is when the PCA feature vector dimension is equal to the length of biohash in bits. In approximating the 1024 dimensional PCA vector from biohash of length 1024 bits, there is a single unique solution. However, the condition number of the random projection matrix is so high and this leads to inaccurate solutions. We improve the solution by decreasing the condition number of the random projection matrix. In this common practice, 20 % of the maximum singular value of the matrix $\mathbf{R}$ is added to its all singular values. This way, the condition number of $\mathbf{R}$ decreases by $\sim 10^2$.

Table 8 FAR1000 values for the proposed methods under the scenario attack in the long term

Method	200 → 1024	1024 → 1024
LP	97.9932	95.9864
BIHT	97.6190	66.1565
L_2	94.1190	54.7789
L_1	89.3027	54.7789

5.3.3 *Computation times for the proposed feature approximation methods*

The proposed feature approximation methods are implemented in MATLAB and the experimental results are run on a 2.5 GHz with 64 GB of RAM PC using 64-bit Windows Server 2008 operating system. From a given biohash of length 1024-bits and the corresponding secret key, we estimate the PCA feature vectors with four proposed methods, for PCA dimensions of 200 and 1024, respectively (Table 9). It is intuitive that for all methods it is faster to estimate a 200-dimensional feature vector. Among the four proposed methods, L_2-norm minimization is the first to estimate a 200-dimensional feature vector from a biohash of length 1024-bits. On the other hand, when the feature vector to be estimated is 1024-dimensional, the BIHT method performs faster than other methods.

5.4 Results for the rainbow attack

The rainbow attack is different from feature approximation methods and its success mainly depends on the availability of a huge biometric database. In this study,

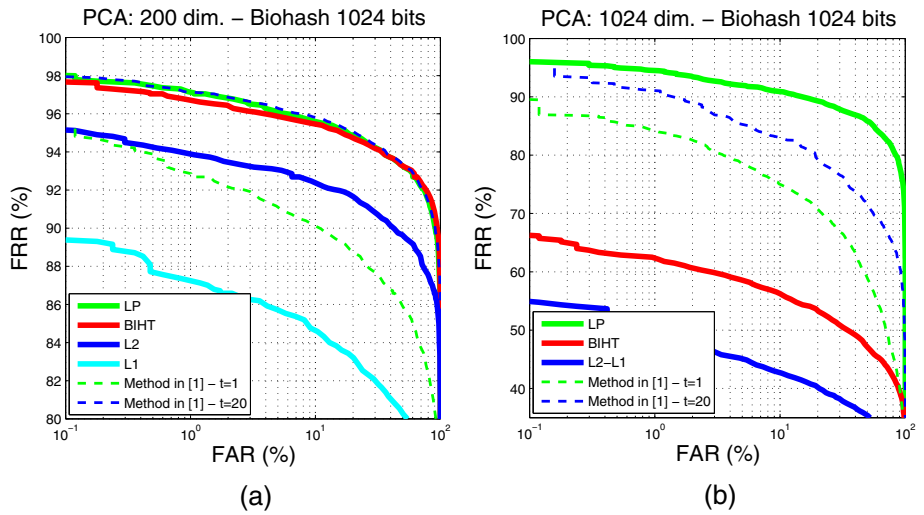

Fig. 5 DET curves for the proposed methods under the scenario Attack in the Long Term. **a** Reconstruction of 200 dimensional PCA feature vectors from biohash of length 1024 bits. **b** Reconstruction of 1024 dimensional PCA feature vectors from biohash of length 1024 bits

Table 9 Computation time required to estimate a feature vector from a given biohash (in seconds)

Method	1024 → 200	1024 → 1024
LP	12.681736	144.288818
BIHT	0.192342	0.294719
L_2	0.108523	1.681796
L_1	11.451703	26.453929
Method in [4] for $t = 1$	28.244039	185.517469
Method in [4] for $t = 20$	572.584385	4700.410120

we simulate the rainbow attack where an adversary has the secret key and the biohash of the user. We use the BioSecure-ds2 database and take the attacked user out of the set. We calculate the biohashes of the remaining face images with the secret key of the user and search for the one that is closest to the user's biohash. In this manner, we describe three different scenarios:

Collusion model (CM): Keys are known to the attacker and using an available database, he finds the faces that provide the closest biohash given the secret key of the valid user.

Security after key change (SAKC): Secret keys of users are changed by the system administrator. The attacker does not know the new key but uses the face found in the CM scenario.

Attack in the long term (ALT): The attacker obtains the new key. He uses the face found in the CM scenario and the new key to create biohashes.

The equal error rates for the rainbow attack on biohashes for these three scenarios are given in Table 10. Our visual inspection shows that faces which create close biohashes when combined with the same secret key are visually alike. This should also be regarded as a threat to the privacy of the user, as well as a threat to the security of the system (Fig. 6).

Table 10 Equal Error Rates (%) for the rainbow attack

PCA	Bit Length	CM	SAKC	ALT
200	128	53.597	6.964	38.571
	256	49.787	4.762	40.179
	512	47.177	4.043	41.820
	1024	46.054	3.342	43.469
1024	128	56.467	7.440	38.746
	256	51.786	5.795	41.417
	512	48.206	4.524	42.543
	1024	46.794	3.296	43.439

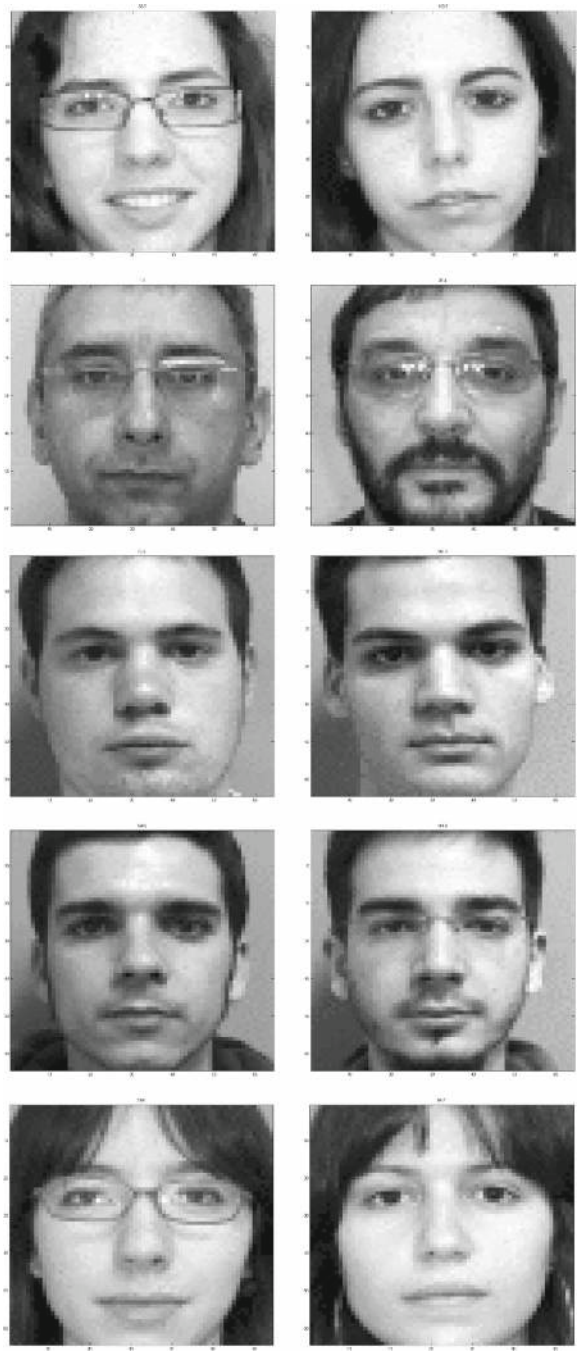

Fig. 6 Rainbow attack. Faces that provide close biohashes

5.5 Privacy assessment for the proposed methods

5.5.1 Visual results of the attacks

A critical implication of the reversibility of biohashes is the relation between the reconstructed feature vectors and the original biometric information (face) of the users. For assessing to what extent the privacy of the user is at stake if his/her biohash is inverted via our proposed methods, we compare face images reconstructed using the original

PCA vectors and the estimated features. Assuming the attacker knows the details of feature extraction (PCA matrix and mean vector), we reconstruct face images with the approximated feature vectors using (16). In the following figures (Figs. 7, 8, 9, and 10), we present the original face image of the user, the reconstructed face image from original PCA coefficients, and the four reconstructed face images from obtained PCA coefficients through L_2, L_1, LP, and BIHT methods, respectively.

The first two set of images (Figs. 7 and 8) belong to two different subjects from the database and the reconstruction is carried out on biohashes with length of 1024 bits which are obtained from 200-dimensional PCA features. All four methods provide face images that look similar to the subject's original face image.

Figures 9 and 10 illustrate the results for the same two subjects. The length of the biohashes used is 1024 bits; however, the only difference is the number of PCA coefficients used, which is 1024 instead of 200. It is immediately clear that estimating 1024-dimensional PCA features is harder than estimating 200-dimensional PCA features and the reconstructed face images show the difficulty in obtaining faces that are visually similar to the original face image. Among the four proposed methods, only LP solution stands out for obtaining face images that look alike the original face of the subject. Figure 11 illustrates the reconstruction of the face images using the LP method for various PCA feature vector dimensions and biohash bit lengths. It is clear that the reconstruction is visually more successful when the length of the PCA feature to be estimated is smaller and the biohash length is larger.

5.5.2 Cross-linking different systems

In addition to visually threatening the privacy of the system users, estimating feature vectors from biohashes might threaten their privacy in other biometric recognitions systems which use the same biometric characteristic (i.e., face information). To check whether reconstructed feature vectors are close to the original PCA feature vectors or not, we include face verification results of PCA vectors, (i.e., reconstructed feature vector is compared to corresponding original feature vector). The Euclidean distance is used to match two PCA vectors and each PCA vector is normalized in order to have zero mean and unit variance prior to comparison. The normalization step is required since the scale of the original PCA coefficients and the reconstructed ones might be different. We do not include all verification results for brevity, but the EERs for PCA-based face verification when 200-dimensional feature vectors are estimated from 1024-bit biohashes are given in Table 11 and the corresponding DET curves are shown in Fig. 12.

6 Conclusions

Biometric template protection is a critical problem that needs to be addressed to enhance the public acceptance of biometric technologies, and it is essential to develop a set of measures which can evaluate the strength of template protection techniques. Although biometric cryptosystems

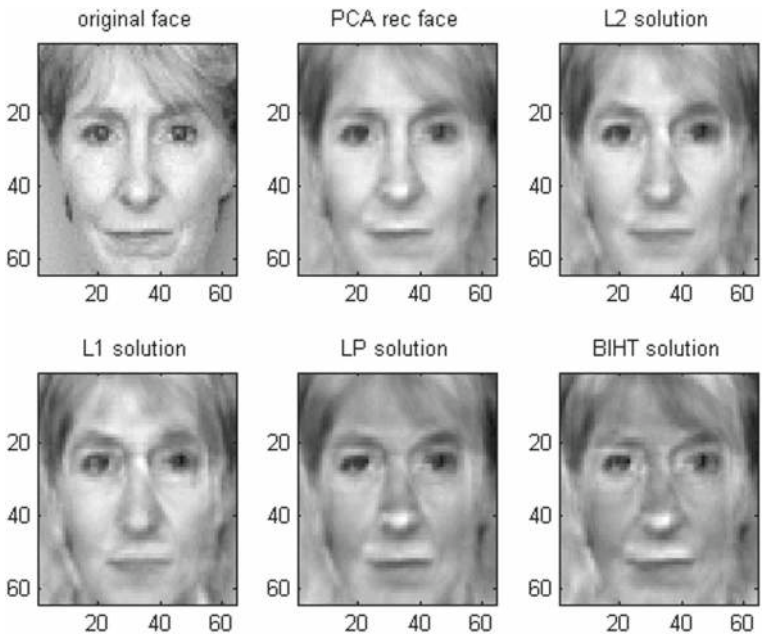

Fig. 7 Reconstructed face images. Reconstructed from biohashes of length 1024 bits—PCA dimension 200

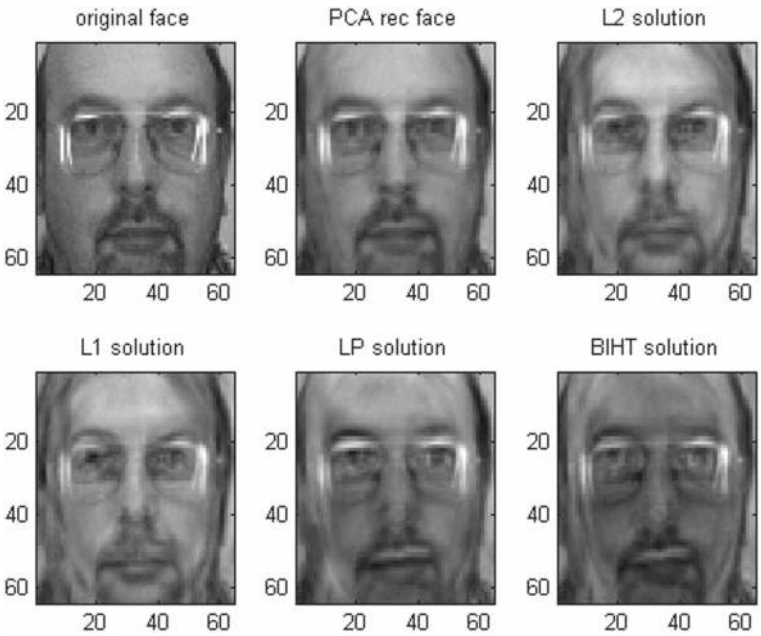

Fig. 8 Reconstructed face images. Reconstructed from biohashes of length 1024 bits—PCA dimension 200

can be analyzed using information theoretical metrics such as entropy and mutual information, the suitability of theoretical analysis of the transformation-based methods is based on the hardness of the invertibility of the transformation.

When a user's biohash is obtained by an adversary, it can seriously undermine the security of the biometric system and privacy of users. If the secret key of a user is known to the adversary, the biometric feature of the user can be reconstructed from the user's biohash which might harm the subject's privacy and lead to illegitimate authentication to a system. Biometric hashing is claimed to be irreversible due to the random projection and quantization steps; however, our study shows that an attacker is

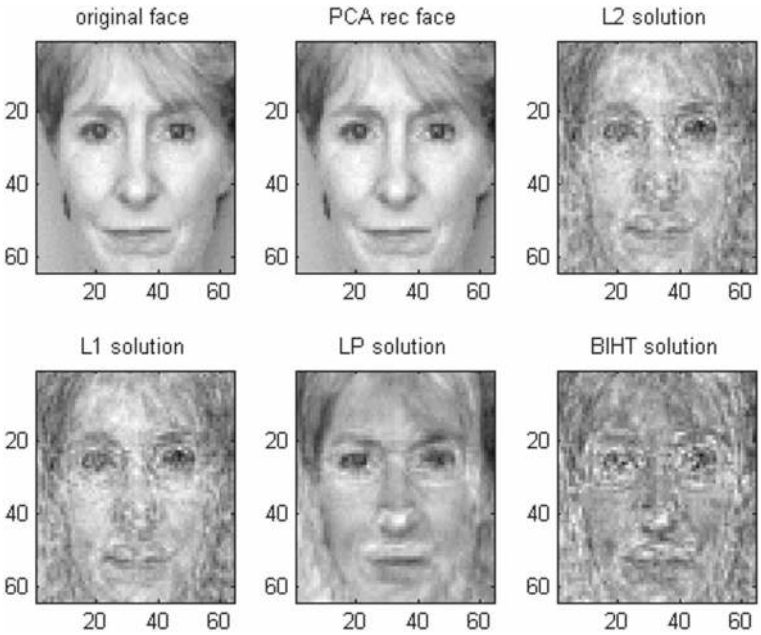

Fig. 9 Reconstructed face images. Reconstructed from biohashes of length 1024 bits—PCA dimension 1024

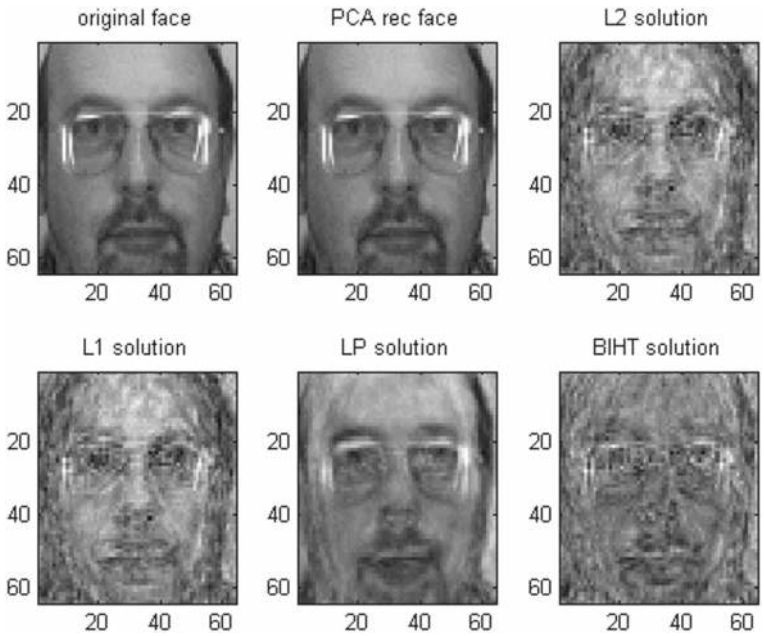

Fig. 10 Reconstructed face images. Reconstructed from biohashes of length 1024 bits—PCA dimension 1024

able to invert the transformed template to obtain a close approximation to the original biometric template.

This paper proposes four novel ways to approximate the original biometric feature from the transformed template in a biometric hashing scheme and reveals security and privacy problems concerning the associated biometric system. We define three different attack scenarios under which we analyze the protection capability of biohashing. From the security point of view, these attacks enable an adversary to recover a biometric template under realistic assumptions and perform intrusion attacks to the biometric system. This study is the first to analyze the inversion of biohashes in a 1-bit compressive sensing framework. Experimental results show the superiority of this approach

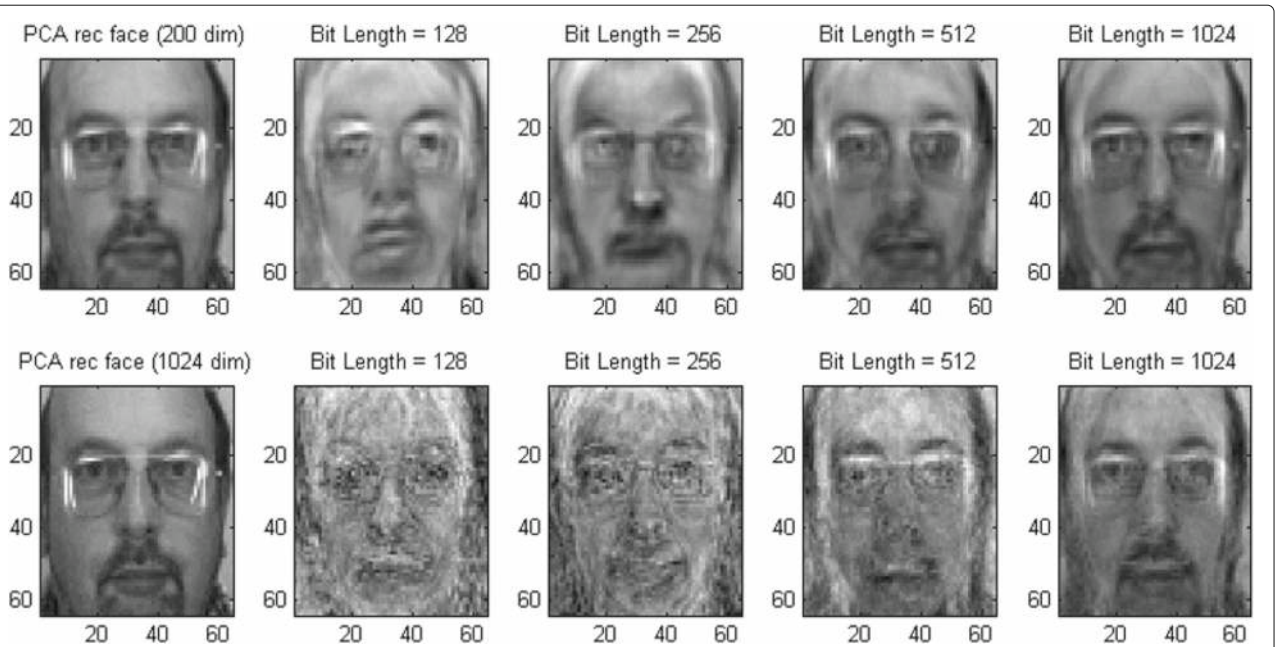

Fig. 11 Reconstructed face images. Reconstructed using LP method for different biohash bit lengths (128, 256, 512, and 1024)

Table 11 Equal error rates (%) for direct feature level comparisons—200-dimensional PCA feature vectors and biohash length = 1024 bits

LP	91.161
BIHT	91.773
L_2	88.338
L_1	78.720

over minimum norm solutions. Biohashes that are created from feature vectors obtained by using LP and BIHT solutions to the 1-bit compressive problem are equal to the original biohashes stored during enrollment, and this is a serious threat to the security of the system. In addition, this study introduces rainbow attack in order to find a biometric template from a biometric database and use it to obtain a biohash that is same with or close to the original biohash of a subject.

Biometric hashing scheme is a generic template protection scheme that can be applied to various types of biometric features. In this paper, we focus on an orthogonal linear transform of face images, namely PCA (i.e., Eigenfaces). Several other studies on biohashing also use PCA ([4, 10]) or LDA ([19]) (i.e., Fisherface) which is another orthogonal linear transform that is invertible. Using the knowledge of the linear transform and its inversion, we analyze the privacy issues by reconstructing face images.

If the adversary knows system details (i.e., the PCA matrix, user's secret key, and other parameters), the obtained feature vectors can be used to reconstruct face images of the subject which is a direct threat to the privacy of system users. The quality of the reconstructed images depends on the number of bits and length of the original feature vector, and the images illustrated in the last section visually confirm the success of the methods in reverting the biohash vectors. In this work, we study feature reconstruction and image reconstruction is carried out separately. Directly approximating images from biohash vectors may also be possible by integrating the PCA transformation with random projection matrix and solving the optimization problem by enforcing sparsity in the DCT or block-DCT domain. However, our initial experiments in this direction indicate that image level approximation approach lowers the performance both in security perspective (evaluated through EERs) and privacy perspective (evaluated through visual inspection of the reconstructed face images) due to the fact that the number of dimensions to be approximated is higher for images.

In the future, the effects of various improvements proposed for biometric hashing scheme might be investigated for security and privacy analysis by carrying out similar attacks on the improved versions of biometric hashing. In addition, weaknesses of the biometric hashing scheme should be explored and possible modifications should be introduced for better security and privacy protection capability in the light of the inversion attacks proposed in this study.

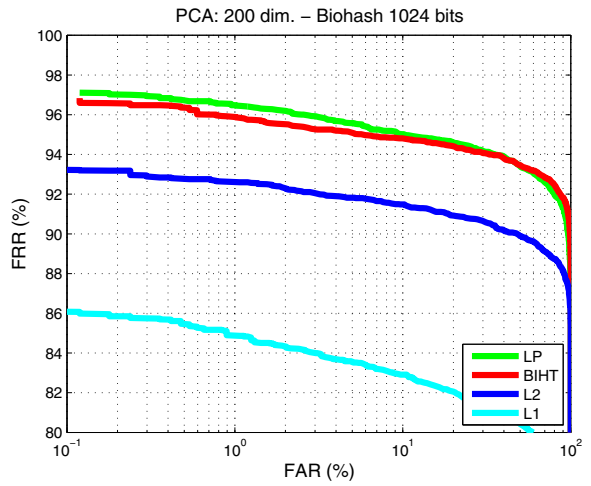

Fig. 12 Feature level comparison. DET curves for direct feature level comparisons—200-dimensional PCA feature vectors and biohash length = 1024 bits

Acknowledgements
This work has been partially supported by the BEAT project 7th Framework Research Programme of the European Union (EU), grant agreement number: 284989. The authors would like to thank the EU for the financial support and the partners within the consortium for a fruitful collaboration. For more information about the BEAT consortium please visit http://www.beat-eu.org.

Competing interests
The authors declare that they have no competing interests.

Received: 10 August 2015 Accepted: 2 September 2016
Published online: 15 September 2016

References

1. T Ignatenko, FMJ Willems, Information leakage in fuzzy commitment schemes. IEEE Trans. Inf. Forensics Secur. **5**(2), 337–348 (2010)
2. S Prabhakar, S Pankanti, AK Jain, Biometric recognition: security and privacy concerns. IEEE Secur. Priv. **1**(2), 33–42 (2003)
3. NK Ratha, JH Connell, RM Bolle, in *Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*. An Analysis of Minutiae Matching Strength, AVBPA '01 (Springer, London, 2001), pp. 223–228
4. A Nagar, K Nandakumar, AK Jain, in *Media Forensics and Security*, ed. by ND Memon, J Dittmann, AM Alattar, and EJ Delp. Biometric Template Transformation: A Security Analysis, SPIE Proceedings, vol. 7541 (SPIE, United States, 2010), p. 75410. doi:10.1117/12.839976
5. AK Jain, K Nandakumar, A Nagar, Biometric Template Security. EURASIP J. Adv. Signal Process. **2008**, 113–117 (2008). doi:10.1155/2008/579416
6. A Juels, M Wattenberg, in *Proceedings of the 6th ACM Conference on Computer and Communications Security*. A Fuzzy Commitment Scheme, CCS '99 (ACM, New York, 1999), pp. 28–36. doi:10.1145/319709.319714
7. ATB Jin, DNC Ling, A Goh, Biohashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern Recogn. **37**(11), 2245–2255 (2004). doi:10.1016/j.patcog.2004.04.011

8. U Uludag, S Pankanti, A Jain, in *Audio- and Video-Based Biometric Person Authentication*, ed. by T Kanade, A Jain, and N Ratha. Fuzzy Vault for Fingerprints, Lecture Notes in Computer Science, vol. 3546 (Springer, New York, 2005), pp. 310–319. doi:10.1007/11527923-32
9. D Maltoni, D Maio, AK Jain, S Prabhakar, *Handbook of Fingerprint Recognition*, 2nd edn. (Springer, London, 2009). doi:10.1007/978-1-84882-254-2
10. DCL Ngo, ABJ Teoh, A Goh, Biometric Hash: High-Confidence Face Recognition. IEEE Trans. Circ. Syst. Video Technol. **16**(6), 771–775 (2006). doi:10.1109/TCSVT.2006.873780
11. X Zhou, Privacy and security assessment of biometric template protection. IT - Inf. Technol. **54**(4), 197–200 (2012)
12. B Yang, C Busch, P Bours, D Gafurov, in *Media Forensics and Security*, ed. by ND Memon, J Dittmann, AM Alattar, and EJ Delp. Robust Minutiae Hash for Fingerprint Template Protection, SPIE Proceedings, vol. 7541 (SPIE, United States, 2010), p. 75410
13. K Kümmel, C Vielhauer, T Scheidat, D Franke, J Dittmann, in *Proceedings of the 11th IFIP TC 6/TC 11 International Conference on Communications and Multimedia Security*. Handwriting Biometric Hash Attack: A Genetic Algorithm with User Interaction for Raw Data Reconstruction, CMS'10 (Springer, Berlin, 2010), pp. 178–190
14. A Kong, KH Cheung, D Zhang, MS Kamel, J You, An analysis of biohashing and its variants. Pattern Recognit. **39**(7), 1359–1368 (2006). doi:10.1016/j.patcog.2005.10.025
15. X Zhou, T Kalker, in *Media Forensics and Security*, ed. by ND Memon, J Dittmann, AM Alattar, and EJ Delp. On the Security of Biohashing, SPIE Proceedings, vol. 7541 (SPIE, United States, 2010), p. 75410. doi:10.1117/12.839165
16. O Goldreich, *Foundations of Cryptography: Volume 1*. (Cambridge University Press, New York, 2006)
17. Y Lee, Y Chung, K Moon, in *IEEE Workshop on Computational Intelligence in Biometrics: Theory, Algorithms, and Applications, CIB*. Inverse Operation and Preimage Attack on Biohashing, (2009), pp. 92–97. doi:10.1109/CIB.2009.4925692
18. P Lacharme, E Cherrier, C Rosenberger, in *International Conference on Security and Cryptography, SECRYPT*, ed. by P Samarati. Preimage Attack on Biohashing (SciTePress, Reykjavík, 2013), pp. 363–370. doi:10.5220/0004524103630370
19. YC Feng, M Lim, PC Yuen, Masquerade attack on transform-based binary-template protection based on perceptron learning. Pattern Recogn. **47**(9), 3019–3033 (2014). doi:10.1016/j.patcog.2014.03.003
20. Y Plan, R Vershynin, One-bit compressed sensing by linear programming. Commun. Pur. Appl. Math. **66**(8), 1275–1297 (2013). doi:10.1002/cpa.21442
21. L Jacques, JN Laska, PT Boufounos, RG Baraniuk, Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. IEEE Trans. Inf. Theory. **59**(4) (2013). doi:10.1109/TIT.2012.2234823
22. C Karabat, H Erdogan, in *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. A Cancelable Biometric Hashing for Secure Biometric Verification System, (2009), pp. 1082–1085. doi:10.1109/IIH-MSP.2009.121
23. Z Bai, D Hatzinakos, in *11th International Conference on Control Automation Robotics Vision (ICARCV)*. Lbp-Based Biometric Hashing Scheme for Human Authentication, (2010), pp. 1842–1847. doi:10.1109/ICARCV.2010.5707216
24. Y Wai Kuan, ABJ Teoh, DCL Ngo, Secure hashing of dynamic hand signatures using wavelet-Fourier compression with BioPhasor mixing and 2N discretization. EURASIP J. Adv. Signal Process. **2007**(1), 32–32 (2007). doi:10.1155/2007/59125
25. C Rathgeb, A Uhl, in *20th International Conference on Pattern Recognition (ICPR)*. Iris-Biometric Hash Generation for Biometric Database Indexing, (2010), pp. 2848–2851. doi:10.1109/ICPR.2010.698
26. A Lumini, L Nanni, An Improved Biohashing for Human Authentication. Pattern Recognit. **40**(3), 1057–1065 (2007). doi:10.1016/j.patcog.2006.05.030
27. M Turk, A Pentland, Eigenfaces for recognition. J. Cogn. Neurosci. **3**(1), 71–86 (1991). doi:10.1162/jocn.1991.3.1.71
28. PT Boufounos, RG Baraniuk, in *42nd Annual Conference on Information Sciences and Systems, CISS*. 1-bit Compressive Sensing, (2008), pp. 16–21. doi:10.1109/CISS.2008.4558487
29. EJ Candes, MB Wakin, An introduction to compressive sampling. IEEE Signal Proc. Mag. **25**(2), 21–30 (2008). doi:10.1109/msp.2007.914731
30. T Blumensath, ME Davies, Iterative hard thresholding for compressed sensing. Appl. Comput. Harmon. Anal. **27**(3), 265–274 (2009). doi:10.1016/j.acha.2009.04.002
31. J Ortega-Garcia, J Fierrez, F Alonso-Fernandez, J Galbally, MR Freire, J Gonzalez-Rodriguez, C Garcia-Mateo, J-L Alba-Castro, E Gonzalez-Agulla, E Otero-Muras, S Garcia-Salicetti, L Allano, B Ly-Van, B Dorizzi, J Kittler, T Bourlai, N Poh, F Deravi, M Ng, M Fairhurst, J Hennebert, A Humm, M Tistarelli, L Brodo, J Richiardi, A Drygajlo, H Ganster, FM Sukno, S-K Pavani, A Frangi, L Akarun, A Savran, The multiscenario multienvironment BioSecure multimodal database (BMDB). IEEE Trans. Pattern. Anal. Mach. Intell. **32**(6), 1097–1111 (2010). doi:10.1109/TPAMI.2009.76
32. P Viola, MJ Jones, Robust Real-Time Face Detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004). doi:10.1023/B:VISI.0000013087.49260.fb
33. KH Cheung, AW-K Kong, J You, D Zhang, in *Proceedings of The 2005 International Conference on Imaging Science, Systems, and Technology: Computer Graphics, CISST*, ed. by HR Arabnia. An Analysis on Invertibility of Cancelable Biometrics Based on Biohashing (CSREA Press, Las Vegas, 2005), pp. 40–45