Jiang *et al. Cybersecurity* (2023) 6:4
https://doi.org/10.1186/s42400-023-00137-0

Cybersecurity

RESEARCH

Open Access

Cancelable biometric schemes for Euclidean metric and Cosine metric

Yubing Jiang[1,2], Peisong Shen[1*], Li Zeng[1,2], Xiaojie Zhu[3], Di Jiang[1] and Chi Chen[1,2]

Abstract

The handy biometric data is a double-edged sword, paving the way of the prosperity of biometric authentication systems but bringing the personal privacy concern. To alleviate the concern, various biometric template protection schemes are proposed to protect the biometric template from information leakage. The preponderance of existing proposals is based on Hamming metric, which ignores the fact that predominantly deployed biometric recognition systems (e.g. face, voice, gait) generate real-valued templates, more applicable to Euclidean metric and Cosine metric. Moreover, since the emergence of similarity-based attacks, those schemes are not secure under a stolen-token setting. In this paper, we propose a succinct biometric template protection scheme to address such a challenge. The proposed scheme is designed for Euclidean metric and Cosine metric instead of Hamming distance. Mainly, the succinct biometric template protection scheme consists of distance-preserving, one-way, and obfuscation modules. To be specific, we adopt location sensitive hash function to realize the distance-preserving and one-way properties simultaneously and use the modulo operation to implement many-to-one mapping. We also thoroughly analyze the proposed scheme in three aspects: irreversibility, unlinkability and revocability. Moreover, comprehensive experiments are conducted on publicly known face databases. All the results show the effectiveness of the proposed scheme.

Keywords Biometric template protection, Distance-preserving hashing, Many-to-one mapping

Introduction

Biometrics (such as fingerprint, iris, and face) has been a popular choice for authentication systems in areas such as military, finance, surveillance, and public security, enjoying a bright application prospect. Generally, a biometric authentication scheme includes two phases: enrollment and verification. In enrollment, a biometric sample (e.g. an image of the face, iris, or finger) of the user is captured by a sensor, then a feature extractor is used to generate a biometric feature vector (i.e., plain biometric template) from the biometric. In verification, the sensor captures a new biometric sample of the user and uses the same feature extractor to extract a fresh biometric template from this new image. Then the fresh biometric template is compared to the enrolled biometric template. If they are similar or close in some metric space, the verification is successful. In the past few decades, innumerable research has been conducted to improve the recognition accuracy of biometric authentication algorithms. In recent years, deep learning-based biometric authentication schemes have achieved a remarkable progress in recognition accuracy [e.g. face (Deng et al. 2020; Institute of Computing Technology 2020; Ranjan et al. 2019), voice (Baevski et al. 2021; Conneau et al. 2020; Xu et al. 2020), and gait (Fan et al. 2020; Chao et al. 2022)]. The state-of-the-art deep learning-based methods output a real-valued biometric feature vector, and the similarity between different biometric templates is usually measured by Euclidean distance or Cosine distance. Technically, the design goal of

*Correspondence:
Peisong Shen
shenpeisong@iie.ac.cn
[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] College of Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates

biometric feature extraction is to make the intra-class distance (i.e., the distance between different feature vectors of the same user) as small as possible, while making inter-class distance (i.e., distance between feature vectors of different users) as big as possible, in order to optimize false acceptance rate (FAR) and false rejection rate (FRR).

However, with large deployments of biometric authentication systems, privacy concern for sensitive biometric data is arising. Recent studies Mai et al. (2018), Gomez-Barrero and Galbally (2020) and Wang et al. (2021b) show that original biometric images can be easily reconstructed from plain biometric templates, which poses a serious threat to personal privacy. In recent years, many privacy-preserving biometric authentication schemes have been proposed to protect the plain biometric templates. Biometric Template Protection (BTP) scheme is a well-known method that generates a transformed (encrypted, projected) template from plain biometric feature vectors. BTP scheme can be seen as a distance-preserving and irreversible transformation of plain biometric feature vectors, so it can enhance the privacy protection of current biometric authentication systems while not downgrade the recognition accuracy. Note that the transform function needs a secret key (i.e., a token provided by the user) to generate the BTP from the biometric feature vector. Besides, a BTP scheme needs to be unlinkable and revocable. Revocable means once an attacker steals the BTP, the legitimate user can use a new token to release a new BTP that is independent of the original one. Unlinkability requires that two BTPs generated by different biometric features of the same user are independent.

Most of the current BTP schemes (Chin et al. 2006; Rathgeb et al. 2014; Pillai et al. 2011; Lai et al. 2017; Sadhya et al. 2019; Sadhya and Raman 2019) are designed for binary biometric templates in Hamming metric. Considering that most biometric recognition methods, in reality, generate real-valued templates with Euclidean or Cosine distance measures (e.g. face, voice, gait), it's urgent to promote the BTP research for Euclidean or Cosine metric. Up to now, only a few BTP schemes are proposed for Euclidean or Cosine metrics. For example, Biohashing (Jin et al. 2004) is the first BTP scheme for face images. It first applies random projection on plain biometric templates to realize dimension reduction and distance-preserving in Euclidean metric. Here, random projection is motivated by the Johnson-Lindenstrauss lemma (Goel et al. 2005) and the group of orthogonal basis vectors used for random projection are generated by user's random seed. Then thresholding is used to binarize the projected values. Further, the result of binary string can be hashed for user verification.

Pathak and Raj (2012) first introduced locality-sensitive hashing (LSH) to generate the cancelable biometric template for real-valued voice signals. The voice feature vector is first processed by Euclidean LSH, then the cryptographic hash value of the result vector is stored as the BTP. It's worth noting that, different from random projection which only realizes dimension reduction in Euclidean metric, LSH functions will discretize the Euclidean vectors and result vector is directly compared in hamming distance. LSH function guarantees a larger collision possibility for vectors with small Euclidean distance or Cosine distance. In this way, the distance is preserved to a certain extent. In recent years, Jin et al. (2018) and Sadhya et al. (2019) proposed "Indexing-of-Max" (IOM) Hashing, which is a special form of LSH, to construct cancelable biometric templates. Lai et al. (2021) also proposed a BTP generation scheme based on Cosine LSH (Charikar 2002).

Besides, some deep learning based BTP schemes (Mai et al. 2021; Hahn and Marcel 2021; Kumar Pandey et al. 2016) are proposed. These schemes usually train a multi-layer neural network to realize the mapping from the Euclidean vector to a randomly distributed codeword. However, compared to the LSH-based (or random projection based) BTP schemes which is conceptually simpler and data-independent, these deep learning based schemes may need to re-train the entire network when re-issuing a new template.

Currently, a major problem of distance-preserving hashing-based schemes (Jin et al. 2004; Pillai et al. 2011; Gomez-Barrero et al. 2014; Lai et al. 2017; Jin et al. 2018; Sadhya et al. 2019; Lai et al. 2021) is that they are vulnerable in stolen-token settings where the adversary can get the key of the user. Patrick et al. and Ghammam et al. (2020) successfully used linear equations and quadratic programming for cryptanalysis of Biohashing, URP-IoM, and GRP-IoM schemes. Without surprise, most BTP schemes usually leak the similarity scores between transformed templates and attackers can launch similarity-based attacks (Wang et al. 2021a; Chen et al. 2019; Dong et al. 2019; Lai et al. 2021) where one can build the target template based on the similarity score in an iterative way.

The problem of protecting biometric data from similarity-based attacks has become the most urgent research issue. Chen et al. (2019) proposed a secure quantization method for biometric templates based on the deep learning method, which obtains a small information leakage by minimizing the variance of inter-class distance in hash space. They proved their method has an excellent recognition performance through experiments on iris. However, in order to minimize the variance of the inter-class distance of hash value, they require the hash distances of different classes to be equidistant. This condition leads to the consequence that the coding space and coding length should be

large and long enough to construct datasets with more classes. Dang et al. (2020) proposed another learning-based approach for biometric template protection. Their proposed model is trained with one-shot and multi-shot enrollment, to encode the biometric data to a predefined output with high probability. However, both methods (Chen et al. 2019; Dang et al. 2020) are strongly data-dependent and need to be pre-defined and pre-trained. Re-training may be required if the scenario or database is changed. As pointed out by Chen et al. (2019), similarity-based attacks come into play when the distribution of similarity scores of different transformed templates exhibits the property of linearity. As a result, these attacks are invalid in the case of non-linearity. Lai et al. (2021) provide a solution for resisting the similarity-based attack, which achieves non-linearity by calculating the similarity inside the subset and then calculating the overall similarity through a threshold. However, if the adversary can know the algorithm details of template generation, the decimal template can be easily transformed back to the binary template. Therefore the relationship becomes linear and the adversary can conduct similarity-based attacks. In general, it's still unclear how to construct a secure BTP scheme against similarity-based attacks under a stolen-token scenario.

In this paper, we first propose a secure BTP scheme for real-valued biometric templates under the Euclidean metric and Cosine metric in stolen-token setting. In particular, a distance-preserving hashing function is first applied to a real-valued biometric feature vector to generate a hashcode vector. Then a many-to-one mapping function is designed and applied to each value in the hashcode vector to generate a protected template. In addition, a threshold-based matcher is designed to compare the input with the stored templates. Due to the design, irreversibility is achieved by introducing the many-to-one mapping mechanism and a novel combination of hash function and many-to-one mapping function. Consequently, the proposed BTP scheme is able to resist linear inequalities attacks in which an attacker can derive a set of linear inequalities from the protected template and then guess the original features by quadratic programming (Ghammam et al. 2020; Lacharme et al. 2013) and similarity-based attacks under the stolen-token scenario. To evaluate the performance, we conduct comprehensive experiments to investigate the effect of Euclidean LSH selection and modulo function selection and compare it with the state of the art in accuracy. Finally, we thoroughly conduct security and privacy analysis, especially in irreversibility, unlikability, and revocability.

In brief, we summarize our contributions below.

1. We propose a secure biometric template protection scheme against similarity-based attacks under the stolen-token scenario. Especially, our scheme instantiates the distance-preserving hashing by utilizing LSH functions and using a modulo function to implement a many-to-one mapping mechanism. Our scheme achieves the same security level as deep learning-based schemes while our scheme is data-independent and conceptually simpler.
2. We thoroughly evaluate the proposed BTP scheme in both theory and experiments. In theory, we analyze the irreversibility, unlinkability and revocability properties of the proposed scheme. In experiments, we conduct a large number of experiments on publicly known face databases, and the result shows that the proposed scheme is effective as claimed.

The rest of the paper is organized as follows. The relevant background knowledge is given in "Preliminaries" section. The scheme of the proposed BTP is described in "The proposed BTP scheme" section. The performance and security analysis are given in "Performance analysis" and "Security and privacy analysis" sections respectively. Finally, conclusion is given in "Conclusion" section.

Preliminaries

Locality sensitive hashing

Locality sensitive hashing (LSH) (Gionis et al. 1999) is primarily designed to solve *Nearest Neighbor Search* problems. Points that are closer in the original metric space have a higher probability of collision and vice versa. For a domain S of the points set, distance measure D, possibility p_1, p_2, vectors p, q and the hashed space U, the LSH family is defined as:

Definition 1 (Gionis et al. 1999). A family $\mathcal{H} = \{h : S \to U\}$ is called $(R; cR; p_1; p_2)$-sensitive for D, if for any $p, q \in S$, a random hash function $h \in \mathcal{H}$:

$$\begin{aligned} &\mathbb{P}_{r\mathcal{H}}[h(p) = h(q)] \geq p_1, \quad \text{if } D(p,q) \leq R \\ &\mathbb{P}_{r\mathcal{H}}[h(p) = h(q)] \leq p_2, \quad \text{if } D(p,q) \geq cR. \end{aligned} \tag{1}$$

According to the distance metric D, the construction of LSH can be divided into LSH under Hamming distance (Gionis et al. 1999), LSH under Euclidean distance (Datar et al. 2004) and LSH under Cosine distance (Charikar 2002). In this paper, we only use Euclidean LSH and Cosine LSH.

Euclidean LSH

Datar et al. (2004) defined a LSH under Euclidean metric. The hash function can transform a n-dimension vector w of real numbers into an integer. The definition is as follows:

$$h_{x,y}(w) = \left\lfloor \frac{x \cdot w + y}{c} \right\rfloor \tag{2}$$

where x is the n-dimensional vector, and each component is identically independent, following the standard normal distribution $\mathcal{N}(0, 1)$. The parameter c is defined as the width of hashing, and y is a real number that is randomly selected from the range of $[0, c]$.

Cosine LSH

Charikar (2002) defined an LSH under the Cosine metric. Its hash function can transform the n-dimension vector w of real numbers into a binary value. Let $\boldsymbol{x}$ and $\boldsymbol{w}$ be the two n-dimension vectors. The hash function $h(\cdot)$ in Cosine LSH is defined as follows:

$$h_x(w) = \begin{cases} 1 & \text{if } x \cdot w > 0 \\ 0 & otherwise. \end{cases} \tag{3}$$

Table 1 Notations and their descriptions

Notations	Descriptions
w	Feature vector
u	Hashcode vector
U	The set of sub-vectors of hashcode vector
d	The sub-vector length in Cosine LSH-based scheme
v	Protected template
n	The length of feature
N	N-to-one mapping
l	The length of template
k	The number of LSH times
I	The vector consisted of k LSH outputs
M	Prime number
A, B	Tokens (secret key) of the user
$h(\cdot)$	Locality sensitive hashing (LSH)
c	Parameter of Euclidean LSH
x, y	Parameters of LSH
τ	Threshold
hd_{nor}	The normalized hamming distance between protected templates
S	The similarity score between protected templates

The proposed BTP scheme

In this section, we introduce the secure biometric template protection (BTP) scheme for the Euclidean metric and Cosine metric. The proposed scheme consists of two phases: enrollment and verification. (1) In enrollment, a biometric feature vector is first extracted from the enrolled biometric image, and then this plain biometric feature vector is transformed into a protected template with the help of a secret token. Finally, the output template is stored as the enrolled template. (2) In verification, a fresh biometric feature vector is extracted from the biometric image of the user, and then this fresh biometric feature vector is transformed into a fresh protected template with the help of the secret token. Afterward, this new template is compared to the enrolled template of by a threshold matcher. Finally, the matcher outputs the authentication result based on the similarity of templates.

In more detail of the scheme shown in Fig. 1, the scheme includes four modules: feature extraction, distance-preserving hashing, many-to-one mapping, and matcher. All the frequent notations used in our work is shown in Table 1.

Feature extraction

State-of-art feature extractors usually output biometric feature vectors (such as the face, iris, voice, and gait) in Euclidean space and Cosine space. The effectiveness

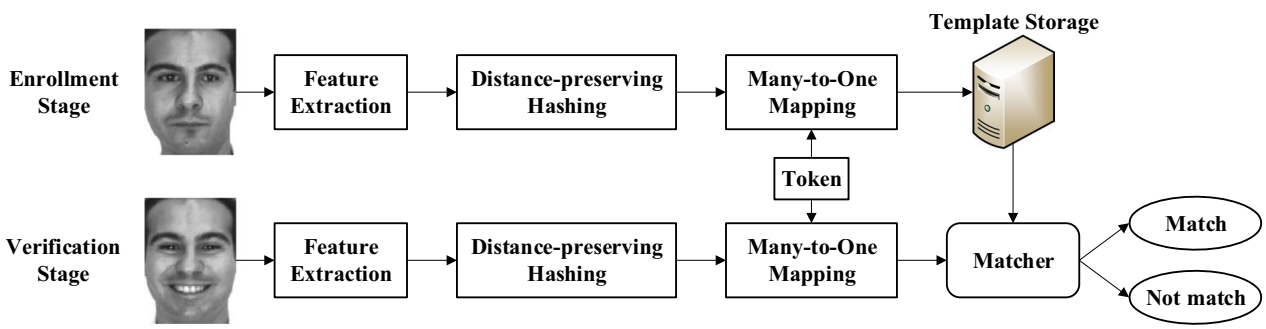


Fig. 1 An overview of our BTP scheme

of the feature extractor is measured by the deviation of intra-class distance distribution and inter-class distance distribution. Note that biometric images can be restored from feature vectors (Mai et al. 2018; Gomez-Barrero and Galbally 2020; Wang et al. 2021b), so the feature extractor almost provides no privacy protection.

LSH-based distance-preserving hashing

The distance-preserving hashing module transforms the biometric feature vectors in Euclidean or Cosine space into a smaller space (in Hamming metric) in a distance-preserving manner. The adjacent points in the original space will remain close after transformation. It's also a dimension-reduction process to guarantee irreversiblity. Currently, random projection (Jin et al. 2004) and LSH (Gionis et al. 1999; Datar et al. 2004) are commonly used to achieve distance-preserving hashing.

In detail, we use two LSH families to transform the real-valued biometric feature vectors into hashcode vectors. One method is based on Euclidean LSH (Gionis et al. 1999) and another is based on Cosine LSH (Datar et al. 2004). Although all LSH families achieve distance-preserving, they are designed for different metric. So the choice of LSH family depends on the metric space where plain biometric features reside. In the original biometric feature space, if the Euclidean metric is adopted for measuring the similarity between biometric features, we will use Euclidean LSH for distance-preserving hashing, otherwise the Cosine LSH is adopted. As shown in Fig. 2, we take the Cosine LSH-based distance-preserving hashing as an example. The detailed steps are as follows. First, k hash functions are randomly selected from the LSH family. Then, the k hash functions are applied to process the input of the biometric feature vector, outputting k hashcode (either 0 or 1) to constitute the hashcode vector.

Note that if only LSH is used in a BTP scheme, the adversary can easily reconstruct a preimage template by using a linear inequalities attack or similarity-based attack under a stolen-token scenario. As a result, we introduce the many-to-one module.

Modulo-based many-to-one mapping

The introduction of many-to-one mapping will further strenghen the irreversibility of transformed templates, preventing BTP scheme from linear inequalities attack (Lacharme et al. 2013; Ghammam et al. 2020) and similarity-based attacks (Lai et al. 2021). In this section, we use the modulo function to instantiate Many-to-one mapping.

Considering that Cosine LSH outputs binary bits while Euclidean LSH outputs integers, the design of many-to-one mapping for Euclidean metric and Cosine metric are slightly different. For the hashcode vector generated from Euclidean LSH, the modulo function is directly applied to each integer in the hashcode vector. However, for Cosine LSH, the generated hashcode vector is first divided into l units (sub-vectors), each unit contains d bits, and $k = l \times d$. We denote the set of these units as U, which is shown as Eq. 4.

$$U = \{U_i | i = \{1, \ldots, l\}, |U_i| = d\} \tag{4}$$

where U_i is an individual unit and $|\cdot|$ denotes its size. Then we convert every hashcode unit U_i into an integer u_i in range of $\{0, \ldots, 2^d - 1\}$. The transformed set of integers is represented as u, which is named as the *hashcode*. The process is shown in Eq. 5, where $C(\cdot)$ stands for the transformation of binary to decimal. Finally, the modulo function is applied to every hashcode unit, shown as Eq. 6, where the value range of u_i is N times as large as M. Therefore, the range of u_i is N times as large as the range of v_i and Eq. 6 realizes a N-to-one mapping. M is selected as the prime number to resist the irreversibility attack, which will be analyzed in detail in the later section. For Cosine LSH based scheme, M is the number closest to $\lfloor 1/\ N{\times}2^d \rfloor$ and for Euclidean-LSH based scheme, M is the number closest to $\lfloor 1/\ N{\times}u_{max} \rfloor$, where

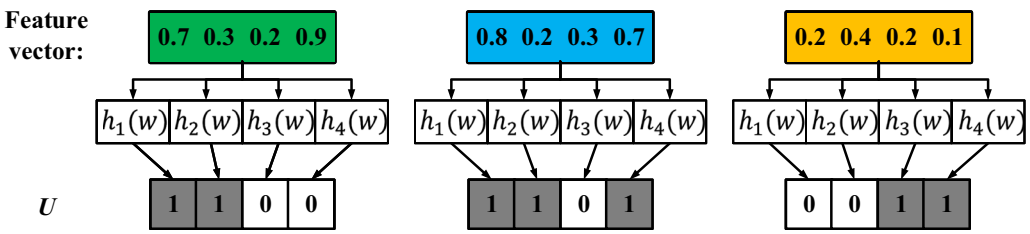


Fig. 2 An example of distance-preserving hashing with k = 4, i.e., four Cosine LSH functions are applied to each feature vector. w denotes the biometric feature vector, and $h(\cdot)$ denotes the Cosine LSH. The "green-colored" feature vector is closer to the "blue" one than the "orange" one. After the distance-preserving hashing, the left result is still closer to the middle result than the right result in Hamming space

u_{max} is the maximal value of u_i. $A = \{a_1, a_2, \ldots, a_l\}$ and $B = \{b_1, b_2, \ldots, b_l\}$ are two l-length random integer vectors stored as the the token of the user, where l is the length of the BTP template. The range of each integer of A or B is the same as u_i. $v = \{v_1, v_2, \ldots, v_l\}$ is the final protected template.

$$u = \{u_i | i = \{1, \ldots, l\}, u_i = C(U_i)\} \tag{5}$$

$$v_i = (a_i \times u_i + b_i) \mod M \tag{6}$$

Algorithms 1 and 2 present the detailed procedures of using Cosine LSH and Euclidean LSH to generate the protected template respectively. In Algorithm 1, we first execute $h_i(w)$ for i from 1 to k, where h_i is the Cosine LSH and k is the total number of hash functions. Then we separate I into l components $\{U_1, \ldots, U_l\}$, where I is the set consisting of $I_i (1 \leq i \leq k)$ and l is the template length. For each $U_j (1 \leq j \leq l)$, it is first converted to a hashcode u_j, then u_j is multiplied with a_j and plus b_j before the result is stored into v_j. Finally, v_j is modular to M and the result is stored back to v_j. All the above v_j is collected into a set v. In Algorithm 2, we first execute $h_i(w)$ for i from 1 to l, where h_i is the Euclidean LSH and l is the total number of hash functions. For each $I_i (1 \leq i \leq l)$, I_i is multiplied with a_i and plus b_i before the result is stored into v_i. Finally, v_i is modular to M and the result is stored back to v_i. All the above v_i is collected into a set v.

For further explaining the algorithms, Fig. 3 presents an example of using Cosine LSH (Algorithm 1) to generate the protected template.

Algorithm 1 BTP Generation: Cosine LSH-based.

Input: feature vector $\boldsymbol{w}$, key $A = \{a_{i\in[1,l]}\}$, $B = \{b_{i\in[1,l]}\}$, k Cosine LSH functions $h_{i\in[1,k]}$, and system parameter M
Output: BTP $v = \{v_1, \ldots, v_l\}$
for $i = 1$ to k **do**
  $I_i = h_i(\boldsymbol{w})$
end for
Separate I into l parts: $\{U_1, U_2, ..., U_l\}$
for $j = 1$ to l **do**
  Convect U_j into decimal u_j
  $v_j = u_j * a_j + b_j$
  $v_j = v_j \mod M$
end for
$v = \{v_1, v_2, \ldots, v_l\}$

Algorithm 2 BTP Generation: Euclidean LSH-based.

Input: feature vector $\boldsymbol{w}$, key $A = \{a_{i\in[1,l]}\}$, $B = \{b_{i\in[1,l]}\}$, k Euclidean LSH functions $h_{i\in[1,k]}$, and system parameter M
Output: BTP $v = \{v_1, \ldots, v_l\}$
for $i = 1$ to l **do**
  $I_i = h_i(\boldsymbol{w})$
  $v_i = I_i * a_i + b_i$
  $v_i = v_i \mod M$
end for
$v = \{v_1, v_2, \ldots, v_l\}$

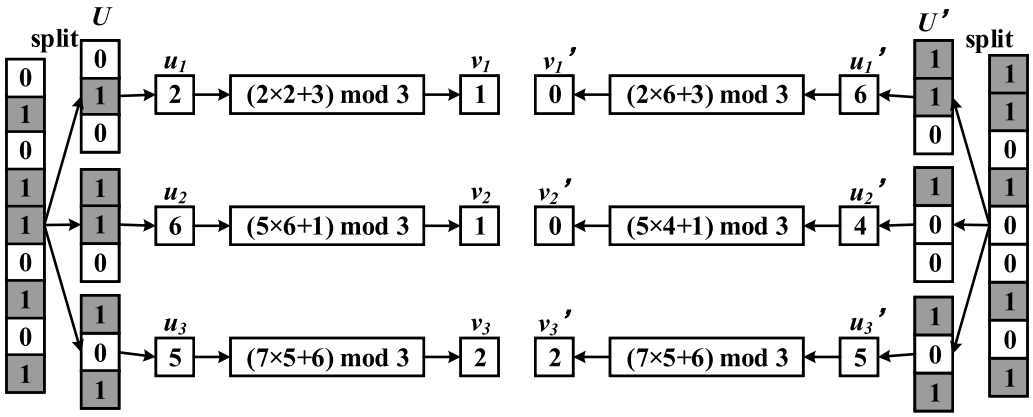


Fig. 3 An example of many-to-one mapping for two different Cosine LSH-based hashcode vectors. In the example, the left hashcode vector is $\{0, 1, 0, 1, 1, 0, 1, 0, 1\}$ and $U = \{\{0, 1, 0\}, \{1, 1, 0\}, \{1, 0, 1\}\}$. Then the integer set $u = \{2, 6, 5\}$ is computed through U. The modulo function is set in the format of $(a \times u_i + b) \mod M$, where $a \in A = \{2, 5, 7\}, b \in B = \{3, 1, 6\}, |U_i| = 3, n = 2$, and $M = 3$ is the prime number closest to $\lfloor 2^{|U_i|}/n \rfloor$. The result of the modulo operation is $v_i \in v = \{1, 1, 2\}$. Similarly, in the right side, we get the $v' = \{0, 0, 2\}$. From the result, we can observe that 2 appears in both v and v' and with 1/2 probability to infer the u_i or u'_i correctly due to the modulo-based many-to-one mapping

Matcher

The matcher module computes the similarity score (i.e. length l divided by hamming distance) between the verified template and enrolled template, and then compares the similarity score with a preset threshold in order to verify the identity of users. Technically, a large score implies a small hamming distance between templates, which further implies more collisions between hashcode vectors, this means plain templates are similar. If the similarity score S between templates exceeds the threshold τ, the two templates will be recognized as the same person, otherwise as different persons. The determination of threshold value will be explained in detail in the experiment section.

The normalized hamming distance between protected templates $v = \{v_1, \ldots, v_l\}$ and $v' = \{v'_1, \ldots, v'_l\}$ is calculated by Eq. 7. In Eq. 7, the exclusive xor is defined as follows: $v_i, v'_i \in \mathbb{Z}$, if $v_i = v_i'$, $v_i \oplus v'_i = 0$; else, $v_i \oplus v'_i = 1$. Therefore, the similarity score S is calculated by hamming distance hd_{nor} according to Eq. 8.

$$hd_{nor} = \frac{\sum_{i=1}^{l} v_i \oplus v'_i}{l} \tag{7}$$

$$S = 1 - hd_{nor} \tag{8}$$

Performance analysis

In this section, to verify the effectiveness of our BTP scheme on real-value biometric templates, we demonstrate the accuracy performance of our BTP scheme on public face databases under various parameter settings. Firstly, the determination of threshold is introduced. Specifically, three parameters are discussed: Euclidean LSH width c (not required in Cosine LSH), parameter N in N-to-1 mapping and BTP template length l. Besides, we show the difference between cosine LSH-based scheme and Euclidean LSH-based scheme.

For the metric of accuracy, we use the typical EER (equal error rate: when FAR is equal to FRR) to evaluate the accuracy of the algorithm (Jin et al. 2004, 2018; Lai et al. 2017; Chen et al. 2019).

Experiment setup

Our experiments are conducted on four databases: AR Face (Martinez and Benavente 1998), CASIA-FaveV5 (http://biometrics.idealtest.org), ORL (https://cam-orl.co.uk/facedatabase.html), and LFW (Huang et al. 2008). Table 2 shows the detail information of the four face databases and the number of template comparisons of inter-class and intra-class in the experiment. In this section, we apply SeetaFace2 (Institute of Computing Technology 2020) in feature extraction to evaluate the performance of our proposed biometric protection template scheme. The cardinality of the face feature vector is 1024.

Table 2 Information of the face databases and the comparison number of intra-class and inter-class in the experiment

Database	Persons	Face images	Intra-class comparison	Inter-class comparison
AR Face	100	2357	26,791	2,747,375
CASIA	500	2491	4961	3,096,330
ORL	40	400	1791	78,000
LFW	5749	13,233	242,257	83,707,271

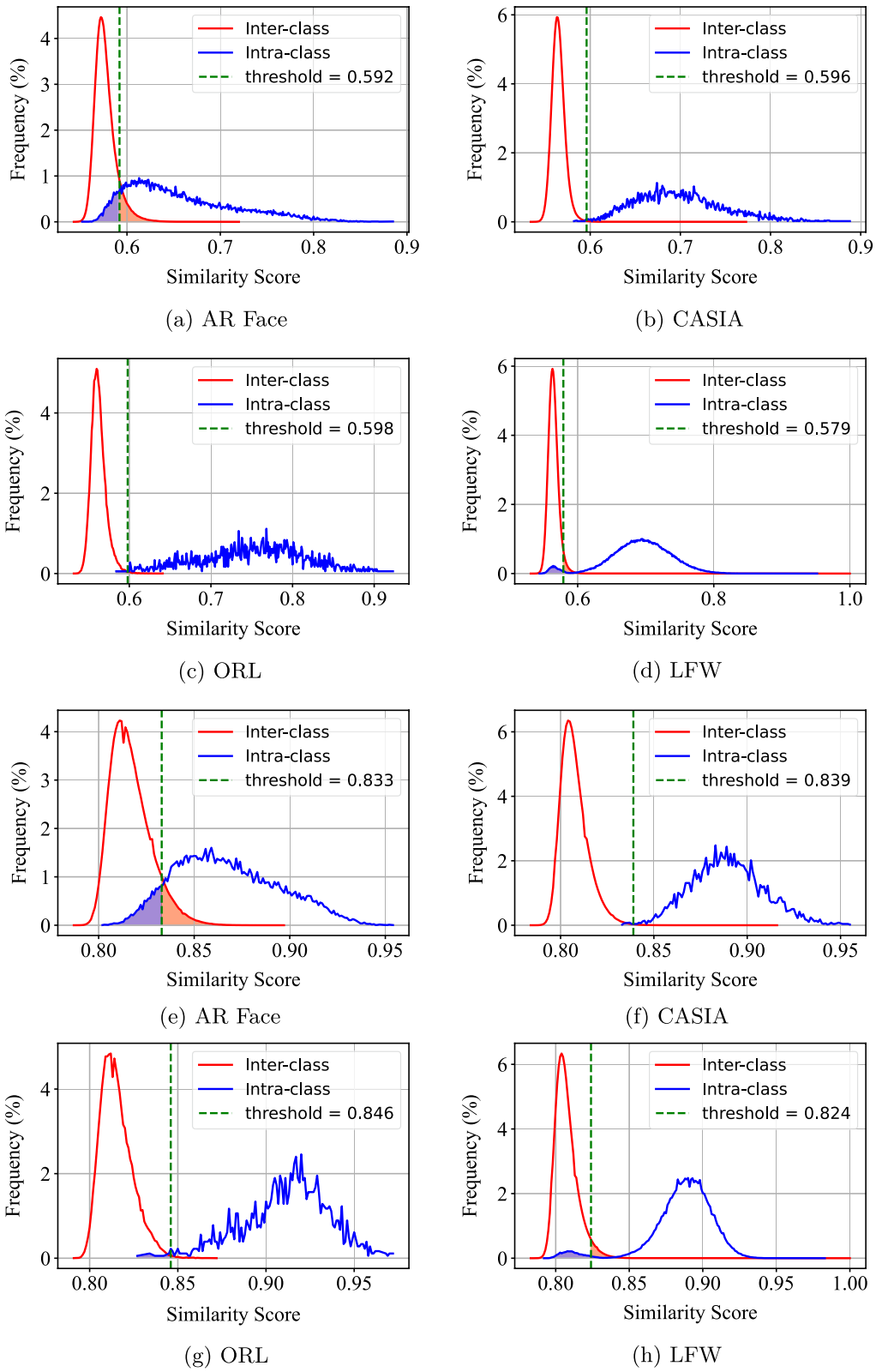


Fig. 4 Threshold of schemes with Euclidean LSH under $k = 5000, n = 2, c = 1.0$ and Cosine LSH under $k = 10{,}000, N = 2, d = 2$. **a**–**d** show the result of Euclidean LSH-based scheme, while **e**–**h** show the result of Cosine LSH-based scheme

Note that our analysis is conducted under a stolen-token setting where the adversary is assumed to can learn tokens of all users. In other words, all users apply the same token in the following experiments. For the genuine-token scenario where the adversary can not know the token, the token will increase the recognition power of cancelable face templates, thus EER of current LSH-based schemes is nearly 0%.

Threshold for similarity scores

In the template matching phase, threshold value plays an important role in deciding whether two templates come from the same person. If the similarity score between two templates exceeds the threshold, the templates are recognized as coming from the same person. Meanwhile, if the similarity score is below the threshold, the templates are recognized from different persons. In this paper, we adopt the common practice (Jin et al. 2018; Lai et al. 2021) that set the threshold value at the point when FAR is equal to FRR. Figure 4 shows the similarity score distributions of templates between different persons (i.e. inter-class comparisons) and between same person(i.e. intra-class comparisons). The area of red-colored region represents FAR and the area of blue-colored region represents FRR. The crossing point of green vertical line and X-asis is set as the threshold. If threshold value increases, the green vertical line moves from left to right on the X-axis, we can see that FRR increases and FAR decreases.

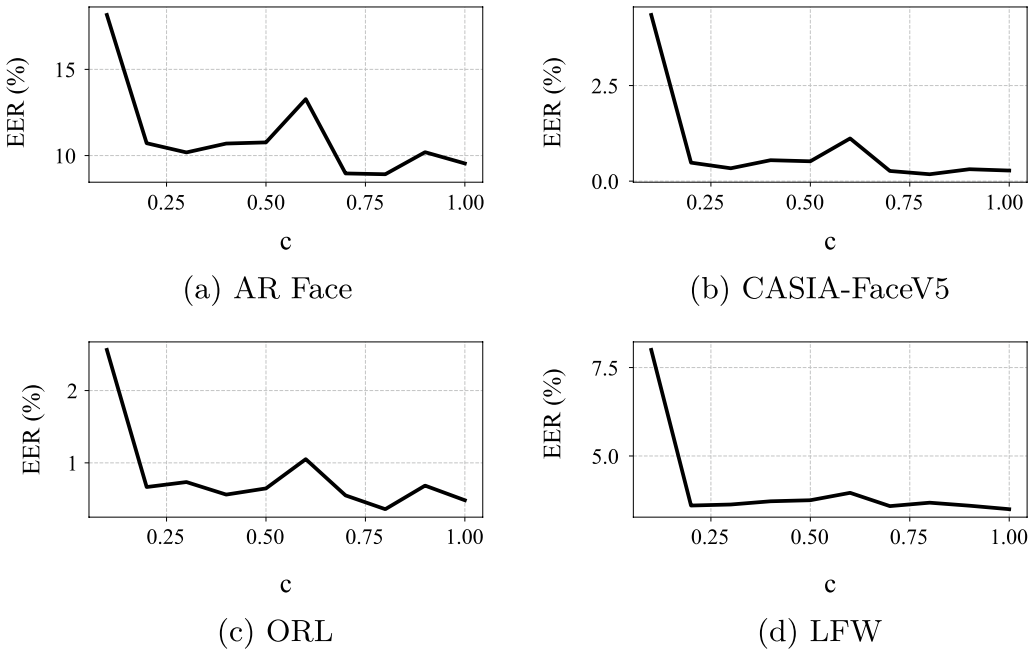


Fig. 5 EER change with Euclidean LSH parameter c under $k = 10{,}000$ and $N = 2$

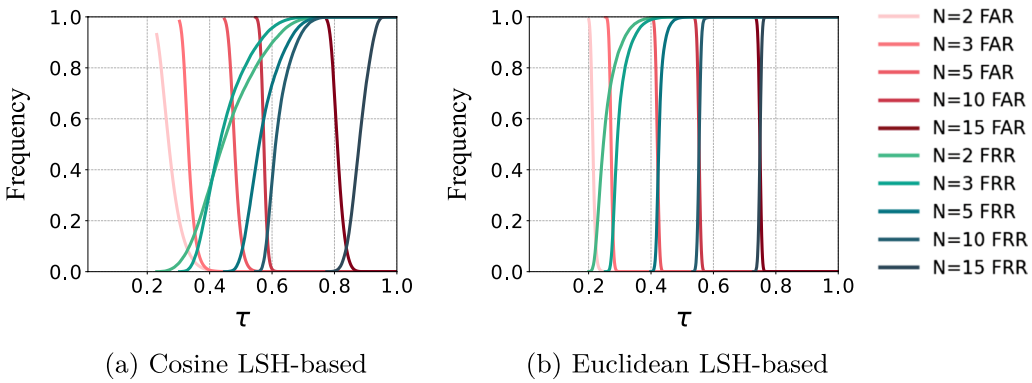


Fig. 6 FAR, FRR, and EER change with various N

When FAR is equal to FRR, the corresponding similarity score is set as the threshold.

Effect of Euclidean LSH width *c*

The width of Euclidean LSH affects the equal error rate (EER) of distance-preserving hashing. The relation between EER and width c is shown in Fig. 5. It can be observed that EER decreases when c increases and $c < 0.2$. When c exceeds 0.2, there is no significant change in EER. The reason is as follows: Based on Eq. 2, one can get: When c increases, the output range of single Euclidean LSH function decreases i.e., u_{max} decreases. When c increases and $c < 0.2$, for intra-class comparisons, the output value of LSH has a larger collison possibility. However for inter-class comparisons, because the output value of LSH is still large, the collsion possibility is not improved too much. Therefore, FRR decreases and FAR rarely increases. As a result, EER will decrease. When c increases and $c > 0.2$, for inter-class comparisons, because the output value of LSH is relatively small, the collsion possibility will also improve, just like the intra-class case. Therefore, FRR decreases and FAR increases. As a result, EER will not change much.

Effect of parameter *N* in *N*-to-1 mapping

In this section, we analyze the effects of the many-to-one mapping parameter N, where N denotes the number of inputs being mapped to the same output. Technically, when N increases, more mapping collisions will appear, leading to a decrease in FRR and an increase in FAR.

We performed our experiments by varying $N \in \{2, 3, 5, 10, 15\}$ while simultaneously setting $k = 10{,}000$, $d = 5$ for Cosine LSH-based scheme and setting $k = 5000$, $c = 0.3$ for Euclidean LSH-based scheme. The EERs of AR face for all settings are presented in Fig. 6. τ indicates the threshold for authentication. From Fig. 6, the result shows that for the same threshold τ, FAR increases, and FRR decreases with the increasing N.

Effect of template length *l*

In our scheme, template length l is related to k (the number of hash functions used in distance-preserving hashing). In the Euclidean LSH-based BTP scheme, template length is equal to the number of hash functions (i.e., $l = k$). In Cosine LSH-based BTP scheme, the number of hash functions is d times as large as the template length (i.e., $l = k/d$). In our experiments, d is set to 2. Intuitively, if the template length is longer, then more LSH functions are required to generate a hashcode vector which results in better distance-preserving. Due to that, the template length affects the performance accuracy of the BTP scheme significantly.

Figure 7 shows the EERs with different LSH output lengths k on different databases. It presents that when k increases, EER first decreases sharply, then maintains stable both for schemes based on Euclidean LSH and Cosine LSH. The longer the template, the longer the

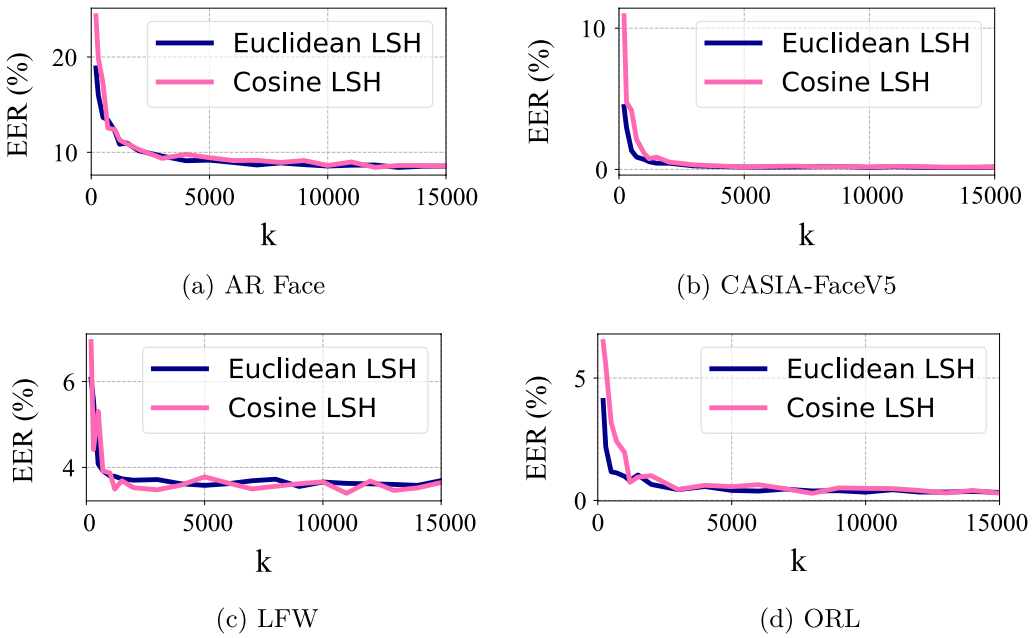


Fig. 7 EER changes with LSH length k under $c = 1.0, d = 2$ and $N = 2$

Table 3 Equal error rate (EER %) comparison with related works

Method	AR	CASIA	ORL	LFW
Before BTP in Euclidean metric	8.298	0.125	0.328	3.599
JHL+18 (Jin et al. 2018) GRP-IoM	8.691	0.230	0.449	**3.426**
JHL+18 (Jin et al. 2018) URP-IoM	8.589	0.176	0.434	3.759
LJW21 (Lai et al. 2021)	9.716	0.239	0.456	3.739
Our Euclidean LSH-based	**8.559**	**0.154**	**0.339**	3.660
Our Cosine LSH-based	8.616	0.179	0.450	3.663

The lowest value of EERs of different schemes are marked in bold

computation time required for template generation. Considering that the time complexity of generating BTP is proportional to LSH length k, according to Fig. 7, we recommend setting $k = 10{,}000$.

Accuracy comparison

To comprehensively demonstrate the accuracy of the proposed scheme, we present a comparison of our BTP scheme with state-of-art ones (Jin et al. 2018; Lai et al. 2021). As shown in Table 3, the EER of our proposed scheme is better than all the schemes except the GRP-IoM scheme, which sacrifices template generation time for better accuracy. Roughly speaking, the template generation time of GRP-IoM is at least four times more than ours and we will show it in the next section. From Table 3, we can find that there is almost no increase in the EER of our scheme after biometric template protection. Besides, Fig. 8 shows the FAR and FRR curves of our scheme when the threshold value changes. When the threshold increases, FAR will decrease and FRR will increase gradually. The value at the intersection point of the two curves represents EER.

Besides, we present the ROC (receiver operating characteristic) curve of our scheme in Fig. 9. From the above result, we can conclude that our scheme achieves better privacy protection in the stolen-token scenario while not degrading the accuracy.

Time cost evaluation

We measure the computational complexity of BTP scheme by template generation time. In our proposed BTP instantiation, the computational cost is mainly from locality-sensitive hashing function and many-to-one mapping function. Table 5 compares the average time cost of generating a BTP template from a 1024-dimension face feature vector on a desktop running 64-bit windows with Intel (R) Core (TM) i5-9500 CPU @ 3.00 GHz and 16GB RAM. The protected template generation of biohashing and our proposed scheme is about 5 ms, while GRP-IoM based scheme is 23.38 ms and URP-IoM based scheme is 10.50 ms. The result shows that the proposed scheme has the same complexity as Biohashing, and is more efficient than other schemes.

Comparison of token length

Our proposed schemes regard the parameters of many-to-one mapping in Eq. 6 as the token (secret key of the user). The token is the vector a and b and the token length is two times of the template length (i.e., $L_{token} = 2 \times l_{template}$). In most schemes (Jin et al. 2004, 2018; Lai et al. 2021), token is set as the projection parameter, and the token length is the result of template length multiplied by feature vector length (i.e., $L_{token} = n \times l_{template}$). n denotes feature length. Taking the feature extraction algorithm of seetaface2 as an example, when the feature-length is 1024, our scheme requires approximately 512 times less in token storage with the same length of the template.

Experiments on different choices of LSH

In this section, we conduct an experiment to demonstrate the difference of Cosine LSH-based scheme and Euclidean LSH-based scheme. In this experiment, SeetaFace (Institute of Computing Technology 2020) and InsightFace (Deng et al. 2019) are both used for face feature extraction. Table 4 shows the equal error rate (EER) of plain template and protected template. From Table 4, we find that SeetaFace is more suitable for feature extraction in Euclidean metric, as the EER of plain template in Euclidean metric is less than that of Cosine metric. On the contrary, InsightFace is more suitable for feature extraction in Cosine metric. Then we apply Euclidean LSH-based BTP scheme and Cosine LSH-based BTP scheme on these plain templates. As the result shows, for templates generated by SeetaFace, EER of protected templates of Euclidean LSH-based scheme is less than that in Cosine LSH-based scheme. For templates generated by InsightFace, EER of protected templates in Cosine LSH-based scheme is less than Euclidean LSH-based scheme. These results prove our intuition that one should choose the same LSH family as corresponding metric space where plain template reside, in order to achieve a better recognition performance. Besides, from Table 5, we can find that the Cosine LSH-based scheme has a faster speed of template generation. Therefore, Cosine LSH-based scheme should be considered with priority when faster calculation speed is demanded.

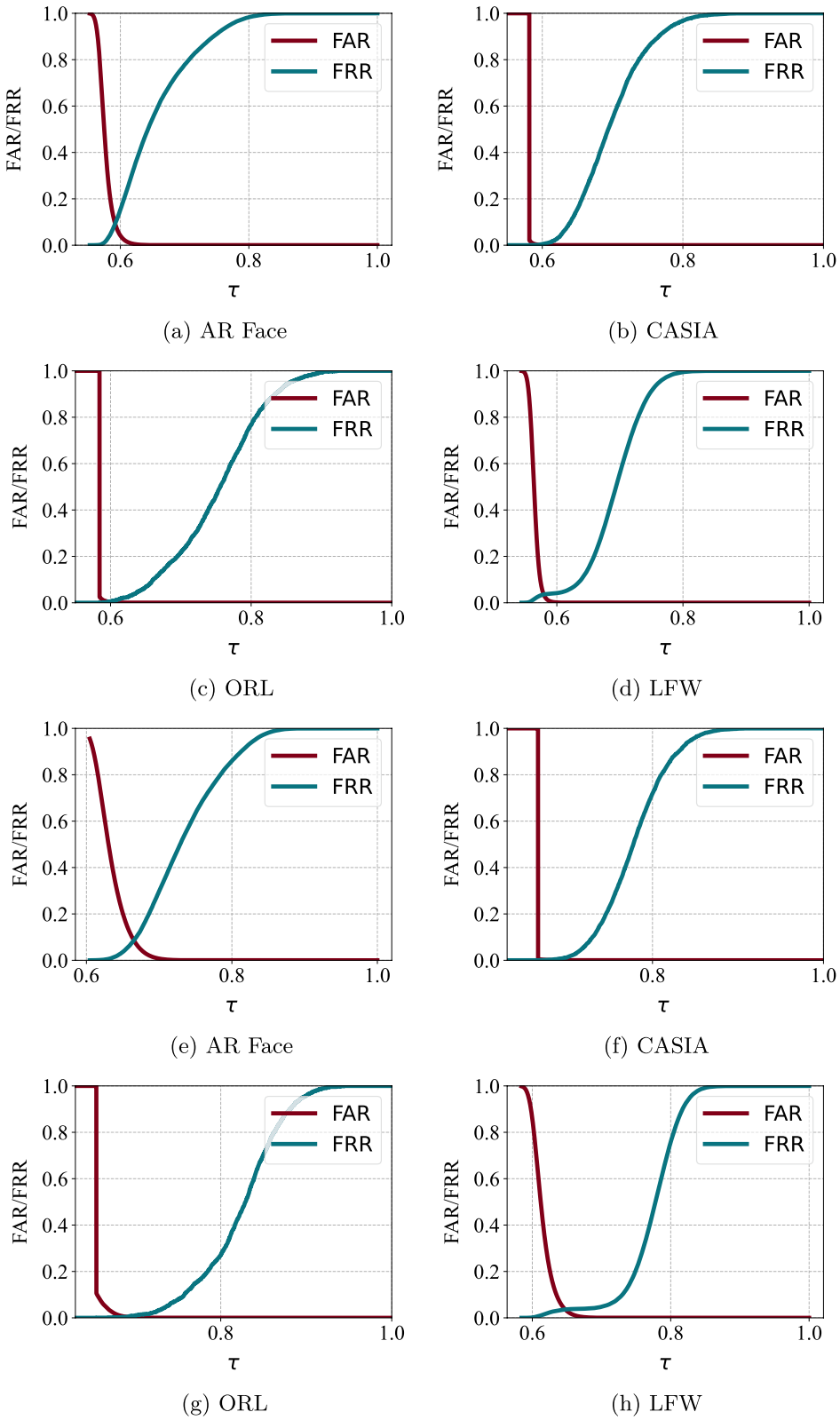


Fig. 8 FAR and FRR change with threshold τ for Euclidean LSH under $k = 5000, N = 2, c = 1.0$ and Cosine LSH under $k = 10{,}000, N = 2, d = 2$. **a**–**d** show the result of Euclidean LSH-based scheme, while **e**–**h** show the result of Cosine LSH-based scheme

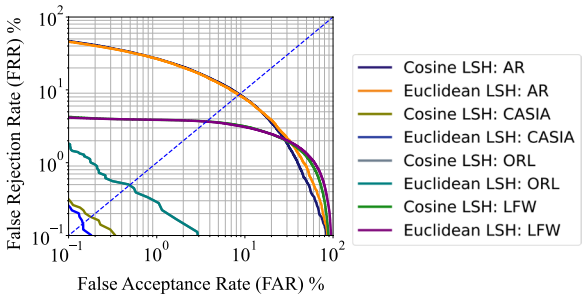


Fig. 9 ROC curves of our scheme over various databases under stolen-token scenario

Table 4 Equal error rate (EER) of plain template and protected template

	Plain template		Protected template	
	Euclidean distance	Cosine distance	Euclidean LSH-based	Cosine LSH-based
SeetaFace (Institute of Computing Technology 2020)				
AR Face	**8.298**	8.882	**8.559**	8.616
CASIA	**0.125**	0.159	**0.154**	0.179
ORL	**0.328**	0.377	**0.339**	0.450
LFW	**3.599**	3.751	**3.660**	3.663
InsightFace (Deng et al. 2019)				
AR Face	2.710	**2.696**	6.332	**3.129**
CASIA	0.161	**0.041**	0.051	**0.018**
ORL	0.034	**0.028**	0.069	**0.042**
LFW	4.993	**4.541**	4.781	**4.601**

The lower value between EER in Euclidean metric and EER in cosine metric is marked in bold

Security and privacy analysis

In this section, we systematically analyze the security of BTP by evaluating the properties of irreversibility, unlinkability, and revocability, following the typical analysis in this area (Jin et al. 2004, 2018; Sadhya and Raman 2019).

Irreversibility

Irreversibility requires that the original face feature vector can not be recovered from the cancelable face template even if the key (token) to generate protected template is leaked.

Ghammam et al. (2020) and Lacharme et al. (2013) pointed out that BTP generated only by LSH is vulnerable to pre-image attacks. The attacker who has got templates of the user and LSH parameters can easily use linear and geometric programming methods to find a fake feature vector that is near to the original feature vector of the legitimate user. To overcome the weakness, our BTP generation scheme not only introduces many-to-one mapping but also novelly combines LSH hashing with many-to-one mapping to achieve irreversibility.

Table 5 Generation time comparison of different template generation techniques (milliseconds)

Scheme	Biohashing	GRP-IoM	URP-IoM	Cosine LSH-based	Euclidean LSH-based
Generation time	4.74	23.38	10.50	4.68	5.88

We adopt the analysis method from Jin et al. (2018) and Sadhya and Raman (2019). In our scheme, vectors to construct linear inequalities are the precondition for an adversary to analyze the irreversibility. The adversary can construct a series of inequalities if knowing the hashcode vector when the token is leaked. However, we use N-to-one mapping to hide the hashcode vector. The adversary will need to find the input of N-to-one mapping in every group correctly with the possibility of $1/N$, so the possibility of finding all hashcode input of all templates is $(1/N)^l$. When l is large enough, our scheme can satisfy the demand of irreversibility.

We discuss the irreversibility against five distinct attacks which have been suggested in previous works (Jin et al. 2018; Sadhya and Raman 2019)—Attack via Single Template, Attack via Record Multiplicity (ARM), Brute Force Attack, False Accept Attack, and Similarity-based Attack. Our model is evaluated under two standard settings - *genuine-token* and *stolen-token*.

(1) Attack via single template This refers to the adversarial ability in restoring the original face feature from a single BTP.

Attack in case of stolen-token In a stolen-token scenario, the adversary can know the templates and tokens of the user. Because the hashcode vector generated by LSH is vulnerable to linear and geometric programming, the best attack strategy for the adversary is to guess the value of the decimal hashcode vector (u). The possibility of successfully guessing the hashcode vector with one guess attempt is $1/\ N^l$. Here l denotes the length of BTP and N denotes the N-to-one mapping. For Cosine LSH-based BTP, we set $k = 10{,}000$, $l = 5000$, $N = 2$. For Euclidean LSH-based BTP, we set $k = l = 5000$. So the probability of successfully guessing the feature vector from BTP is $1/2^{5000}$.

Attack in case of genuine-token In this case, the adversary can only know templates, so the possibility of guessing hashcode vector (u) with one attempt is $1/2^{d*l}$

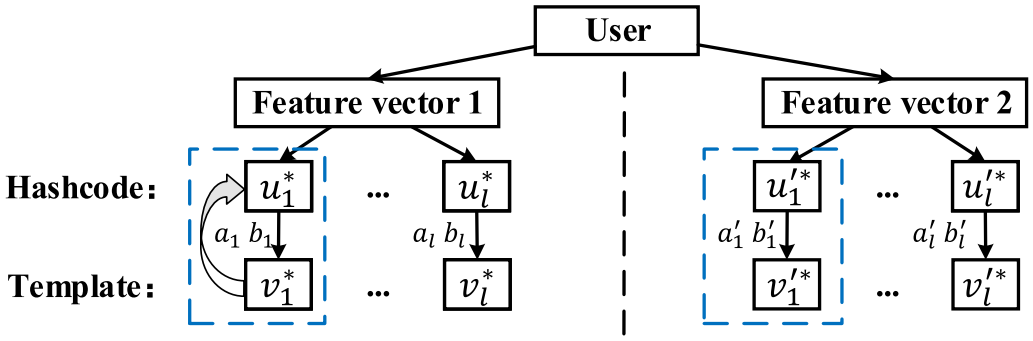


Fig. 10 An explanation of ARM attack to our proposed scheme. The two sides of the black dotted line represent the template generations by same distance-preserving hashing of feature vectors of the same user with different tokens

(Cosine LSH-based BTP) or $1/(u_{max}+1)^l$ (Euclidean LSH-based BTP). Here, d denotes the sub-vector length in the Cosine LSH-based scheme, and u_{max} indicates the maximum value in the range of u. In our setting, $d = 2$, $l = 5000$, $u_{max} = 6$. So the probability of successfully guessing the feature vector from BTP is $1/2^{10,000}$ (Cosine LSH-based BTP) or $1/7^{5000}$ (Euclidean LSH-based BTP), which is totally not feasible in real-world execution.

(2) Attack via record multiplicity (ARM) ARM is a more dreadful attack, which refers to the adversarial ability in restoring the original face feature from multiple BTPs of the target user. We first analyze the attack in the case the adversary can know two templates and tokens of one user and then generalize it to multiple. Figure 10 explains the ARM attack on our proposed scheme. In Fig. 10, the adversary tries to obtain the hashcode $\{u_1^*, \ldots, u_l^*\}$, in the case of knowing templates $\{v_1^*, \ldots, v_l^*\}$ and $\{u_1'^*, \ldots, u_l'^*\}$. In our scheme, the generation of each entry of the template $(v_1^*, \ldots, v_l^*)$ is independent, so the analysis of reversing one entry of template can be generalized to the whole template. Next, we analyze that the adversary reverses the first entry of the hashcode vector through the first entry of templates which is the part framed by the blue dotted box in Fig. 10.

If an adversary can gain one compromised protected template v_1^*, corresponding token $\{a_1, b_1\}$, and public scheme parameter M, according to Eq. 6, the adversary can be capable to infer the hashcode u through Eq. 9.

$$a_1 \times u + b_1 \equiv v_1^* \mod M \tag{9}$$

In more detail, to infer u, there are two cases:

(1) a and M are not coprime. Since M is a prime number, when a and M are not mutually prime, a can only be a multiple of M (i.e., $a = m \times M, m \in \mathbb{Z}$). Equation 9 can be rewritten to Eq. 10, which can be further simplified into Eq. 11.

$$m \times M \times u + b_1 \equiv v_1^* \mod M \tag{10}$$

$$b_1 \equiv v_1^* \mod M \tag{11}$$

In this case, u can be any value of its domain, and the value of one entry of template v_1^* is determined by the value of token b. If a_1 and M are not coprime, one entry of the template is determined by b_1. However, it should be noticed that the possibility that a and M are not coprime in every entry is low. In our experiment, $M = 3$ and the $l = 5000$, the value range of a is $\{1, \ldots, 10\}$, therefore the possibility is $(\frac{3}{10})^{5000}$. Therefore, the adversary is not able to gain an advantage in constructing the congruence equations.

(2) a and M are coprime. According to the relationship of hashcode and template, let u_1^* denote the real hashcode calculated from the feature vector by LSH and it can be expressed as Eq. 12, thus u_1^* must be one of the solutions of Eq. 9. Because a and M are coprime, all the solutions of Eq. 9 are $u = \{u_i | u_i = u_1^* + iM, i \in \mathbb{Z}, u_i \in \mathbb{S}\}$. $\mathbb{S}$ is the range of LSH output.

$$v_1^* = (a_1 \times u_1^* + b_1) \mod M \tag{12}$$

In ARM attack, the adversary can know multiple templates and their corresponding tokens and then the adversary can construct Eq. 13:

$$\begin{cases} v_1^* &= (a_1 \times u_1 + b_1) \mod M \\ v_1'^* &= (a_1' \times u_1 + b_1') \mod M \end{cases} \tag{13}$$

Templates v_1^*, $v_1'^*$ are generated by hashcode u_1^*, $u_1'^*$, so u_1^* must be a solution of the first equation of Eq. 13 and $u_1'^*$ must be a solution of the second equation of Eq. 13.

u_1^* and $u_1'^*$ are outputs of same LSH of the near biometric features collected by the same user in different times. If $u_1^* = u_1'^*$, the solution of Eq. 13 is

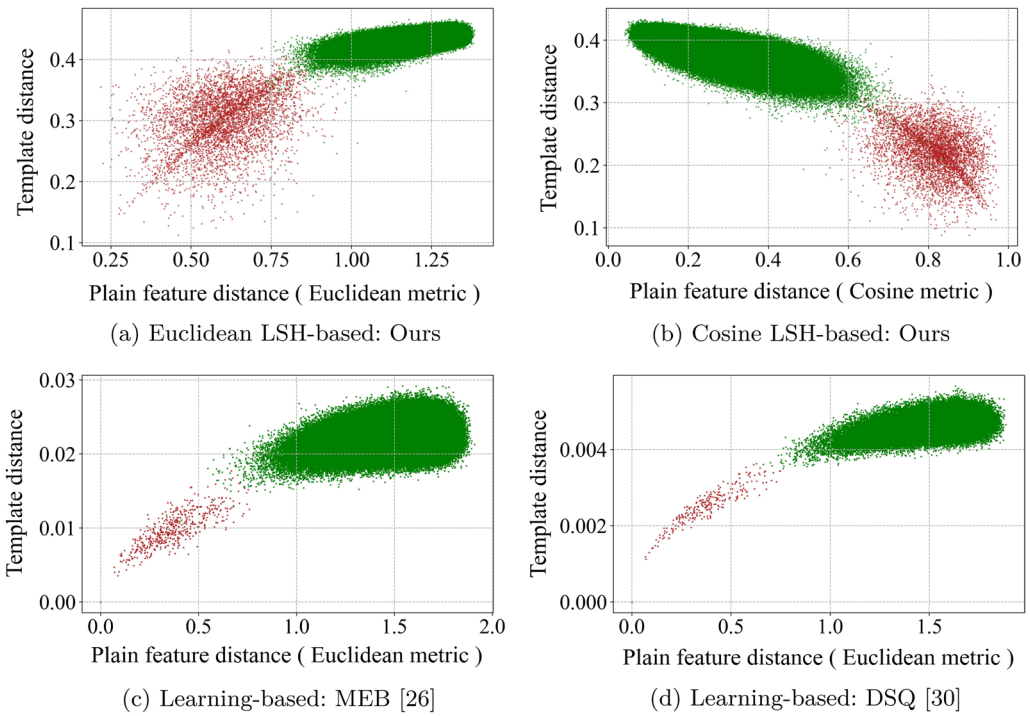
$u = \{u_i | u_i = u_1^* + iM, i \in \mathbb{Z}, u_i \in \mathbb{S}\}$. If $u_1^* \neq u_1'^*$, the adversary will have no additional information to obtain the hashcode u. The attack result of the adversary with one template is the same as the attack result of the adversary with multiple templates. For the ARM attack on one entry of hashcode, the adversary cannot determine which solution is the real hashcode, so the adversary can only randomly guess the above solutions. For many-to-one mapping, there is no additional benefit to attacking with multiple face templates. Hence, the attack complexity is the same as an attack via a single template.

(3) Brute force attack Brute force attack on the feature: First, count the features extracted from four face databases, and the minimum value and the maximum value are 0.00 and 0.34 respectively. Since the feature is a real-value vector, we set the precision to 0.01. Then the possible value number of each entry of the feature is $(0.34 - 0.00)/0.01 = 34$. The length of the feature is 512, so the difficulty of a brute force attack on the feature is $34^{512} = 2^{2605}$.

Brute force attack on template: For the Euclidean-LSH-based scheme, under the best accuracy parameter setting $M = 3$, the maximum value and the minimum value of protected templates is 2 and 0. The length of the template is 5000 (for Cosine LSH-based) and 10,000 (for Euclidean LSH-based), so the difficulty of brute force attack on the template is $3^{5000} = 2^{7925}$ (for Cosine LSH-based) and $3^{10,000} = 2^{15,850}$ (for Euclidean LSH-based).

(4) False accept attack False accept attack (dictionary attack) requires farless number of attempts to gain illegitimate access (Tams et al. 2015). In this attack, access would be granted as long as the comparison score succeeds the pre-defined threshold τ. In our template protection scheme, the template length is l, so the adversary can pass the verification by guessing the template with length $l \times \tau$. τ and l are 0.836, 5000 respectively in the scheme based on Euclidean distance, so the success possibility is $1/2^{(l\times\tau)} = 1/2^{4180}$ for one attempt of guess. Moreover, τ and l are 0.592, 10,000 respectively in the Cosine LSH-based scheme, so the success possibility is $1/2^{(l\times\tau)} = 1/2^{5920}$ for one attempt of guess.

(5) Similarity-based attack Next, we analyze the irreversibility of our scheme against similarity-based attack which uses the information leakage on the distance of projected template to approximate plain template in an iterative manner. In this section, we prove the proposed scheme can resist the similarity-based attack based on the following two facts: (1) we conduct an experiment to show that the intra-class and inter-class distance distributions before and after transformation under the



(a) Euclidean LSH-based: Ours (b) Cosine LSH-based: Ours

(c) Learning-based: MEB [26] (d) Learning-based: DSQ [30]

Fig. 11 The relationship between plain feature distance and template distance of different BTP schemes. Here, "red" point denotes intra-class. The "green" point denotes inter-class

stolen-token scenario is non-linear. (2) We propose a simplified similarity-based attack and shows that our scheme can resist this attack while state-of-art ones can not.

To resist similarity-based attacks, the distance of plain templates comparisons and distance of protected template comparisons should exhibit a certain extent of non-linearity, as suggested by Chen et al. (2019) and Lai et al. (2021). Figure 11 shows the relationship between plain feature distance and template distance of our schemes and two deep-learning-based BTP schemes (Kumar Pandey et al. 2016; Chen et al. 2019). Our experiment is conducted on CASIA database where each subject has multiple facial images, 60% of which are used for training and 40% for testing (Note that training is only essential for deep learning based schemes). The X-axis shows the distance of plain template comparisons and Y-axis shows the distance of protected template comparisons. Generally speaking, our BTP scheme achieves the same level of irreversibility against similarity-based attacks as learning-based BTP schemes (Kumar Pandey et al. 2016; Chen et al. 2019). Besides, our LSH-based BTP scheme is simpler and more efficient than learning-based BTP schemes in template re-generation.

Besides, we give a similarity-based attack, its procedures are shown in Algorithm 3.

In this attack, the attacker knows all information about BTP parameters and tokens. The attacker's goal is to use this information to reconstruct a fake face template that is similar enough to the authentic one. In detail, the attack works as follows: First, the adversary chooses a guessing initial feature vector (all zeros), and the initial vector will generate the initial template. Then Z number of disturbing noises are generated and each noise is added to s number of positions respectively, where the magnitude of the noise is t. Next, the adversary uses the same parameter and key to generate Z number of guessing templates from guessing feature vectors with noise. Then, the adversary compares the distance between the guessing template and the real template (intercepted by an adversary), and the guessing feature vector corresponding to the template with minimized distance is updated as the initial vector in the next guessing round.

In our experiment, we choose seetaface2 (Institute of Computing Technology 2020) as the feature extraction algorithm and extract the normalized feature vector, so the initial guessing vector is the origin point. The parameter setting is as follows: $Z = 50$, $s = 20$, $t = 0.01$ and the iteration number $T = 500$.

Figure 12 shows the attack results of our attack scheme on the CASIA-FaceV5 database. The results show that we have good attack effects on the schemes of Lai et al. (2021) and Jin et al. (2018), and our proposed scheme has the best attack resistance effect.

Algorithm 3 The proposed similarity-based attack.

Input: template vector $\boldsymbol{v} = \{v_1, \ldots, v_l\}$, key $A = \{a_{i\in[1,l]}\}$, $B = \{b_{i\in[1,l]}\}$, k LSH functions $h_{i\in[1,k]}$
Output: attack result $w^* = \{w_1^*, \ldots, w_n^*\}$

initialize $w_0 = \{0, \ldots, 0\}$, $d_0 = d(v, BTP(A, B, h, w_0))$
for $i = 1$ to T **do**
  for $j = 1$ to Z **do**
    random generate $pos_{i,j} \in \{0, 1\}^n$, $|pos_{i,j}| = s$
    random generate $sym_{i,j} \in \{-1, 1\}^n$
    $e_{i,j} = pos_{i,j} * sym_{i,j} * t$
    $w_{i,j} = e_{i,j} + w_{i-1}$
    $d_{i,j} = d(v, BTP(A, B, h, w_{ij}))$
  end for
  $d_{i,min} = min(d_{i,1}, \ldots, d_{i,Z})$
  if $d_{i,min} < d_{i-1,min}$ **then**
    $w_i = w_{i,min}$
  else
    $w_i = w_{i-1}$
  end if
end for
output $w^* = w_N$

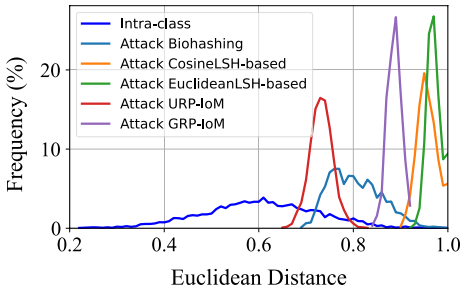


Fig. 12 Our proposed similarity-based attack results towards advanced cancelable biometrics schemes on CASIA-faceV5 database

Unlinkability

A user can enroll one of his biometric traits for multiple applications. Unlinkability states that the template generated from a biometric feature must not allow crossing matching among other templates generated from the same feature.

We follow the method (Jin et al. 2018) to conduct the unlinkability analysis. Pseudo-imposter score distribution and pseudo-genuine score distribution are compared to verify the unlinkability of our proposed scheme. If both distributions are overlapped, then the adversary can not distinguish whether the two templates generated by BTP come from the same user. Therefore, cancelable templates are unlinkable.

Pseudo-imposter score The comparison score between templates generated from the same original feature of same user and different tokens. In our experiment, 50 random number tokens are used to generate 50 templates. Comparing the first template with the rest 49 templates, 49 pseudo-imposter Scores are computed for each feature.

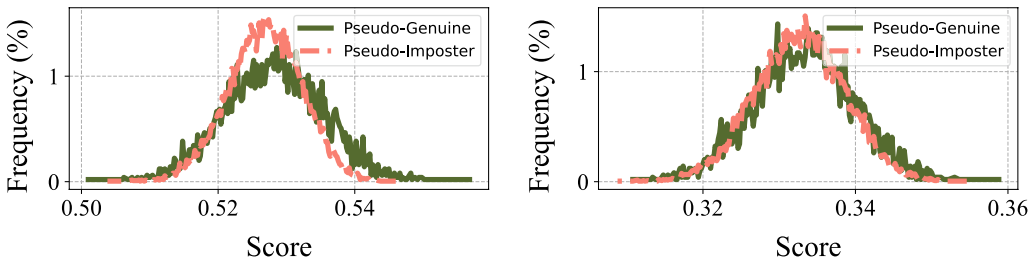
Pseudo-genuine score The comparison score between templates generated from different features and different tokens.

Table 6 shows the mean value and standard deviation variance of pseudo-imposter and pseudo-genuine of our BTP schemes. Taking the CASIA-FaceV5 database as an example, the distributions of pseudo-imposter and pseudo-genuine are presented in Fig. 13. As expected, distributions of pseudo-imposter scores and pseudo-genuine scores are highly overlapped. Hence, the unlinkability of our proposed template protection scheme is verified.

Gomez-Barrero et al. (2017) proposed a general scheme for the quantitative evaluation of biometric templates' unlinkability. The unlinkability of the protected templates can be more thoroughly analyzed by the mated H_m (i.e., different templates from the same user) and non-mated H_{nm} (i.e., different templates from different users) score distributions. The authors proposed the notion of $D_{\leftrightarrow}^{sys}$ to measure the system's global unlinkability, which ranges from 0 (completely unlinkable) to 1.0 (completely linkable). In this section, we adopt this measure to evaluate the scheme's unlikability. Table 7 compares the $D_{\leftrightarrow}^{sys}$ of the state-of-art cancelable biometric schemes with our scheme on CASIA-FaceV5 database. As Table 7 shows, our BTP schemes have better unlinkability. From Table 7, we can find that the unlinkability of Eculidean LSH-based scheme is better than that of Cosine LSH-based scheme. Therefore, Euclidean LSH-based scheme is prefered when unlinkability is considered in priority.

Table 6 The mean values and standard deviation variances of genuine, imposter, pseudo-imposter and pseudo-genuine of proposed scheme based on Euclidean LSH and Cosine LSH

	Genuine		Imposter		Pseudo-imposter		Pseudo-genuine	
	Mean	var($\times 10^{-3}$)	mean	var($\times 10^{-5}$)	Mean	var($\times 10^{-5}$)	Mean	var($\times 10^{-5}$)
AR Face								
Euclidean LSH	0.655	3.401	0.333	4.463	0.331	3.619	0.333	4.486
Cosine LSH	0.919	0.452	0.526	5.024	0.533	3.597	0.528	5.103
ORL								
Euclidean LSH	0.765	4.080	0.334	4.696	0.331	0.386	0.333	4.407
Cosine LSH	0.949	0.254	0.526	5.327	0.529	5.420	0.529	5.332
CASIA								
Euclidean LSH	0.700	2.357	0.334	4.436	0.333	3.507	0.333	4.315
Cosine LSH	0.935	0.195	0.526	5.003	0.527	2.895	0.528	5.106
LFW								
Euclidean LSH	0.696	2.052	0.334	4.432	0.334	3.497	0.333	4.463
Cosine LSH	0.930	0.318	0.526	4.946	0.536	2.962	0.528	5.030



(a) Cosine LSH-based (b) Euclidean LSH-based

Fig. 13 The pseudo-genuine and pseudo-imposter score distributions of CASIA database with $k = 10{,}000, d = 2, n = 2, N = 2$

Table 7 Quantitative evaluation of cancelable biometrics of CASIA-FaceV5 database

Scheme	URP-IoM	GRP-IoM	Euclidean LSH-based	Cosine LSH-based
$D_{\leftrightarrow}^{sys}$	0.118	0.144	0.038	0.079

Revocability

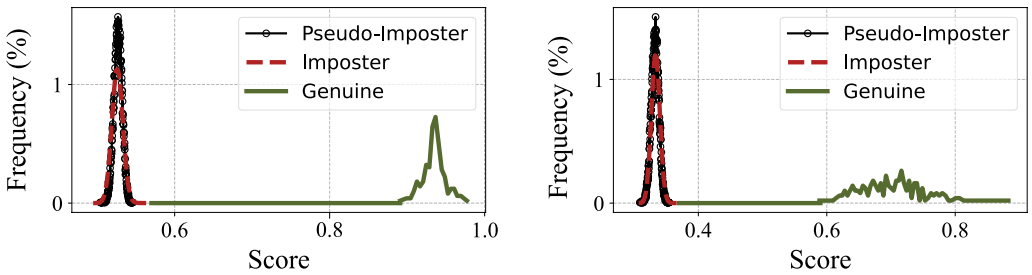
Revocability requires that if the protected template is leaked, the stored template can be revoked and reissued, and the new template is totally independent from the leaked template.

We follow the common measurement of revocability in Jin et al. (2018, 2012). In detail, we evaluate three score distributions: genuine score (comparison scores of BTPs generated from different features of same user), imposter score (comparison scores of BTPs generated from different users with different tokens), and pseudo-imposter score (comparison scores of same feature with different tokens). According to Jin et al. (2018, 2012) and Cho and Teoh (2017), revocability needs to match two properties. First, the distribution of imposter score and pseudo-imposter score is overlapped. Second, the distribution of genuine score and pseudo-imposter score is distinguishable.

Table 6 shows the experimental result of mean value and standard deviation, running on AR Face, ORL, CASIA-FaceV5, and LFW databases. From Table 6, the mean value and standard deviation of the imposter distribution are almost the same as the pseudo-imposter distribution, while the mean value and standard deviation of the genuine distribution are quite different from the pseudo-imposter distribution. To vividly show the above result, Fig. 14 is given. From the above analysis, we can draw a conclusion that our scheme meets the requirement of revocability.

Conclusion

In this paper, we propose a succinct scheme of secure biometric template protection which is secure against powerful similarity-based attacks under stolen-token scenario. In the scheme, we introduce not only a many-to-one mapping mechanism but also a novel combination of distance-preserving hashing and many-to-one mapping to overcome the weakness of the existing BTP scheme. Moreover, we instantiate the scheme by adopting the LSH function to realize distance-preserving hashing and designing a modulo function to implement many-to-one mapping. Finally, we conduct a comprehensive theory and experiment analysis of the instantiation.



(a) Cosine LSH-based (b) Euclidean LSH-based

Fig. 14 The pseudo-imposter, imposter and genuine score distributions of CASIA-FaceV5 database with $k = 10{,}000, d = 2, N = 2$

Author information
Peisong Shen is an assistant researcher in State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences. His research interest includes applied cryptography and privacy protection. He got his bachelor degree from University of Science and Technology of China in 2012 and graduted from School of Cyber Security, University of Chinese Academy of Sciences in 2018. Since then, he joined current laboratory as research assistant.

Acknowledgements
Portion of the research in this paper use the CASIA-FaceV5 collected by Chinese Academy of Sciences' Institute of Automation (CASIA).

Author contributions
All authors have contributed to this manuscript and approve of this submission. YJ participated in all the work and drafting the article. PS, XZ and Prof. CC have made many contributions to the technical route, designing research, and revising the article. LZ and DJ contributed to the analysis and interpretation of experimental data. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The data that support the findings of this study are available from the corresponding author, upon reasonable request .

Declarations

Competing interests
We confirm that none of the authors have any competing interests in the manuscript.

Received: 3 October 2022 Accepted: 30 December 2022
Published online: 02 February 2023

References

AT &T Laboratories Cambridge. The ORL database of faces. https://cam-orl.co.uk/facedatabase.html
Baevski A, Hsu W-N, Conneau A, Auli M (2021) Unsupervised speech recognition. Adv Neural Inf Process Syst 34:27826–27839
Chao H, Wang K, He Y, Zhang J, Feng J (2022) Gaitset: cross-view gait recognition through utilizing gait as a deep set. IEEE Trans Pattern Anal Mach Intell 44(7):3467–3478
Charikar MS (2002) Similarity estimation techniques from rounding algorithms. In: Proceedings of the thiry-fourth annual ACM symposium on theory of computing, pp 380–388
Chen Y, Wo Y, Xie R, Wu C, Han G (2019) Deep secure quantization: on secure biometric hashing against similarity-based attacks. Signal Process 154:314–323
Chin CS, Jin ATB, Ling DNC (2006) High security iris verification system based on random secret integration. Comput Vis Image Underst 102(2):169–177
Chinese Academy of Sciences Institute of Automation: CASIA-FaceV5. http://biometrics.idealtest.org
Cho S, Teoh ABJ (2017) Face template protection via random permutation maxout transform. In: Proceedings of the 2017 international conference on biometrics engineering and application, pp21–27
Conneau A, Baevski A, Collobert R, Mohamed A, Auli M (2020) Unsupervised cross-lingual representation learning for speech recognition. arXiv preprint arXiv:2006.13979
Dang TM, Tran L, Nguyen TD, Choi D (2020) Fehash: full entropy hash for face template protection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp 810–811
Datar M, Immorlica N, Indyk P, Mirrokni VS (2004) Locality-sensitive hashing scheme based on p-stable distributions. In: Proceedings of the twentieth annual symposium on computational geometry, pp 253–262
Deng J, Guo J, Xue N, Zafeiriou S (2019) Arcface: additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4690–4699
Deng J, Guo J, Ververas E, Kotsia I, Zafeiriou S (2020) Retinaface: single-shot multi-level face localisation in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern fingerprint recognition, pp 5203–5212
Dong X, Jin Z, Jin ATB (2019) A genetic algorithm enabled similarity-based attack on cancellable biometrics. In: 2019 IEEE 10th international conference on biometrics theory, applications and systems (BTAS). IEEE, pp 1–8
Fan C, Peng Y, Cao C, Liu X, Hou S, Chi J, Huang Y, Li Q, He Z (2020) Gaitpart: temporal part-based model for gait recognition. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 14213–14221
Ghammam L, Karabina K, Lacharme P, Thiry-Atighehchi K (2020) A cryptanalysis of two cancelable biometric schemes based on index-of-max hashing. IEEE Trans Inf Forensics Secur 15:2869–2880
Gionis A, Indyk P, Motwani R et al (1999) Similarity search in high dimensions via hashing. In: VLDB, vol 99, pp 518–529
Goel N, Bebis G, Nefian A (2005) Face recognition experiments with random projection. In: Biometric technology for human identification II, vol 5779. SPIE, pp 426–437
Gomez-Barrero M, Galbally J (2020) Reversing the irreversible: a survey on inverse biometrics. Comput Secur 90:101700
Gomez-Barrero M, Rathgeb C, Galbally J, Fierrez J, Busch C (2014) Protected facial biometric templates based on local Gabor patterns and adaptive bloom filters. In: 2014 22nd international conference on pattern recognition. IEEE, pp 4483–4488
Gomez-Barrero M, Galbally J, Rathgeb C, Busch C (2017) General framework to evaluate unlinkability in biometric template protection systems. IEEE Trans Inf Forensics Secur 13(6):1406–1420
Hahn VK, Marcel S (2021) Biometric template protection for neural-network-based face recognition systems: a survey of methods and evaluation techniques. ArXiv arXiv:2110.05044
Huang GB, Mattar M, Berg T, Learned-Miller E (2008) Labeled faces in the wild: a database forstudying face recognition in unconstrained environments. In: Workshop on faces in 'real-life' images: detection, alignment, and recognition
Institute of Computing Technology (2020) SeetaFace2. https://github.com/seetafaceengine/SeetaFace2
Jin ATB, Ling DNC, Goh A (2004) Biohashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern Recognit 37(11):2245–2255
Jin Z, Teoh ABJ, Ong TS, Tee C (2012) Fingerprint template protection with minutiae-based bit-string for security and privacy preserving. Expert Syst Appl 39(6):6157–6167
Jin Z, Hwang JY, Lai Y-L, Kim S, Teoh ABJ (2018) Ranking-based locality sensitive hashing-enabled cancelable biometrics: index-of-max hashing. IEEE Trans Inf Forensics Secur 13(2):393–407
Kumar Pandey R, Zhou Y, Urala Kota B, Govindaraju V (2016) Deep secure encoding for face template protection. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 9–15
Lacharme P, Cherrier E, Rosenberger C (2013) Preimage attack on biohashing. In: 2013 International conference on security and cryptography (SECRYPT). IEEE, pp 1–8
Lai Y-L, Jin Z, Teoh ABJ, Goi B-M, Yap W-S, Chai T-Y, Rathgeb C (2017) Cancellable iris template generation based on indexing-first-one hashing. Pattern Recognit 64:105–117
Lai Y, Jin Z, Wong K, Tistarelli M (2021) Efficient known-sample attack for distance-preserving hashing biometric template protection schemes. IEEE Trans Inf Forensics Secur 16:3170–3185
Mai G, Cao K, Yuen PC, Jain AK (2018) On the reconstruction of face images from deep face templates. IEEE Trans Pattern Anal Mach Intell 41(5):1188–1202
Mai G, Cao K, Lan X, Yuen PC (2021) Secureface: face template protection. IEEE Trans Inf Forensics Secur 16:262–277
Martinez A, Benavente R (1998) The AR face database. Tech. Rep. 24 CVC Technical Report
Pathak MA, Raj B (2012) Privacy-preserving speaker verification as password matching. In: 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 1849–1852

Pillai JK, Patel VM, Chellappa R, Ratha NK (2011) Secure and robust iris recognition using random projections and sparse representations. IEEE Trans Pattern Anal Mach Intell 33(9):1877–1893
Ranjan R, Patel VM, Chellappa R (2019) Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Trans Pattern Anal Mach Intell 41(01):121–135
Rathgeb C, Breitinger F, Busch C, Baier H (2014) On application of bloom filters to iris biometrics. IET Biom 3(4):207–218
Sadhya D, Raman B (2019) Generation of cancelable iris templates via randomized bit sampling. IEEE Trans Inf Forensics Secur 14(11):2972–2986
Sadhya D, Akhtar Z, Dasgupta D (2019) A locality sensitive hashing based approach for generating cancelable fingerprints templates. In: 2019 IEEE 10th international conference on biometrics theory, applications and systems (BTAS). IEEE, pp 1–9
Tams B, Mihăilescu P, Munk A (2015) Security considerations in minutiae-based fuzzy vaults. IEEE Trans Inf Forensics Secur 10(5):985–998
Wang H, Dong X, Jin Z, Teoh ABJ, Tistarelli M (2021a) Interpretable security analysis of cancellable biometrics using constrained-optimized similarity-based attack. In: 2021 IEEE winter conference on applications of computer vision workshops (WACVW), pp 70–77
Wang H, Wang S, Jin Z, Wang Y, Chen C, Tistarelli M (2021b) Similarity-based gray-box adversarial attack against deep face recognition. In: 2021 16th IEEE international conference on automatic face and gesture recognition (FG 2021), pp 1–8
Xu Q, Likhomanenko T, Kahn J, Hannun A, Synnaeve G, Collobert R (2020) Iterative pseudo-labeling for speech recognition. arXiv preprint arXiv:2005.09267

Publisher's Note