Practical Construction of Face-Based Authentication Systems with Template Protection Using Secure Sketch

Tran Dang, Quynh Truong, Tran Dang

Practical Construction of Face-based Authentication Systems with Template Protection Using Secure Sketch

Tran Tri Dang, Quynh Chi Truong, Tran Khanh Dang

Faculty of Computer Science & Engineering, Ho Chi Minh City University of Technology
{tridang, tqchi, khanh}@cse.hcmut.edu.vn

Abstract. Modern mobile devices (e.g. laptops, mobile phones, etc.) equipped with input sensors open a convenient way to do authentication by using biometrics. However, if these devices are lost or stolen, the owners will confront a highly impacted threat: their stored biometric templates, either in raw or transformed forms, can be extracted and used illegally by others. In this paper, we propose some concrete constructions of face-based authentication systems in which the stored templates are protected by applying a cryptographic technique called secure sketch. We also suggest a simple fusion method for combining these authentication techniques to improve the overall accuracy. Finally, we evaluate accuracy rates among these constructions and the fusion method with some existing datasets.

Keywords: Secure sketch, Face Authentication, Biometric template protection.

1 Introduction

Current mobile devices (laptops, mobile phones, etc.) are used not only for simple tasks like communicating or web browsing, but they can also be used to do more complex tasks such as learning and working. As a result, some sensitive information is stored on mobile devices for such tasks. To ensure the confidentiality of the personal information, usually an authentication process is implemented. Besides password-based authentication, modern devices equipped with input sensors open a new way of doing authentication: biometric. Using biometric for user authentication actually has some advantages [1]. However, while passwords can be easily protected by storing only their one-way hash values, it is not easy to do so with biometric data. The problem lies in the noisy nature of biometric data, i.e. the biometric templates captured from the same person in different times will certainly be different. Hence, if we apply the one-way hash function to biometric data, we will unable to compare the distance between the stored data and the authentication data.

In this paper, we propose one construction that offers face-based authentication and provides protection for stored biometric templates at the same time. We follow the concept of "secure sketch" proposed by Dodis et al. [2]. One property of secure sketch is that it allows reconstructing the original biometric template exactly when

provided another template that is closed enough to the first one. Because of that property, we can protect the stored templates by using a one-way hash function on them.

The remains of this paper are structured as follow: in section 2, we review some related works; in section 3, we present our construction technique of secure sketch on 3 different face recognition algorithms; in section 4, our experiment results are reported and base on them we introduce a simple fusion method to improve the performance of our system; finally, in section 5, we conclude the paper with findings and directions for future researches.

2 Related Works

Using biometric for authentication is not new [3]. One perspective of this problem is how to reliably and efficiently recognize and verify the biometric features of people. This is an interesting topic for pattern recognition researchers. Another perspective of this problem is how to protect the stored templates and this is the focus of security researchers as well as this paper.

There are many approaches to the problem of template protection for biometric data. One of which is the "secure sketch" proposed by Dodis et al. [2]. In its simplest form, the working of the secure sketch is described in Fig.1.

Fig. 1. The working of the secure sketch

There are 2 components of the secure sketch: the sketch (*SS*) and the recover (*Rec*). Given a secret template *w*, the *SS* component generates public information *s* and discards *w*. When given another template *w'* that is closed enough to *w*, the *Rec* component can recover the *w* exactly with the help of *s*. There are 3 properties of the secure sketch as described in [2]:

1. *s* is a binary string {0, 1}*.
2. *w* can be recovered if and only if $|w - w'| \leq \delta$ (δ is a predefined threshold).
3. The information about *s* does not disclose much about *w*.

In our construction, only property 2 is guaranteed. Fortunately, we can easily transform our sketch presentation into binary string to make it compatible with property 1. And although we do not prove property 3 in this paper, we do give a method to measure the reduction of the search space used to brute-force attack this system using public information s. For these reasons, we still call our construction secure sketch.

Our construction is implemented with biometric templates extracted from 3 face recognition methods: the Eigenfaces method proposed by Turk and Pentland [4], the 2DPCA method proposed by Yang et al. [5], and the Local Binary Patterns Histograms (LBPH) proposed by Ahonen et al. [6]. We select more than one face recognition method to experiment how generic our construction is when applied to different

template formats. Another reason is we want to combine the results of these individual authentications to improve the overall resulted performance.

To recover *w* exactly when given another *w'* closed to it, some error correction techniques are needed. In fact, the public information *s* is used to correct *w'*, but it should not disclose too much about *w*. We follow the idea presented in [7] paper to design this error correction technique. Our technique can be applied on discrete domains and it gives reasonable results when experimenting with the Eigenfaces, 2DPCA, and LBPH face recognition methods.

Individual biometric recognition systems can be fused together to improve the recognition performance. The fusion can be implemented at feature extraction level, score level, or decision level [8]. In this paper, based on the specific results obtained from the experiments with individual features (i.e. Eigenfaces, 2DPCA, and LBPH), we propose a simple fusion technique at the decision level, and in fact, it improves the overall performance significantly.

3 Construction Methods

3.1 Processing Stages

The stages of our construction are summarized in Fig. 2. Firstly, the face feature is extracted. The formats of the extracted features depend on the face recognition methods used. Secondly, a quantization process is applied to the features' values in continuous domain to convert them into values in discrete domain. This stage is needed because discrete values allow exact recovery more easily than continuous values do. The quantized values play the role of *w* and *w'* as described in previous section. The sketch generation stage produces *s* given *w*. And finally, the feature recovery stage tries to recover the original *w* given another input *w'* and *s*. To validate whether the recovered feature matches with the original feature *w*, a one-way hash function can be applied to *w* and the result is stored. Then, the same hash function will be applied to the recovered feature and its result is compared with the stored value.

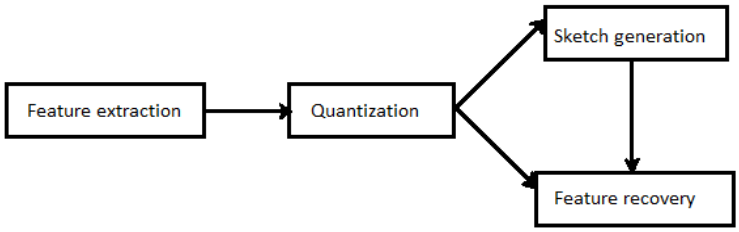

Fig. 2. The stages of our construction

3.2 Feature Extraction

Eigenfaces

Given a set of training face images, the Eigenfaces method finds a subspace that best represents them. This subspace's coordinates are called eigenfaces because they

are eigenvectors of the covariance matrix of the original face images and they have the same size as the face images'. The detail of the calculating these eigenfaces was reported in [4].

Once the eigenfaces are calculated, each image can be projected onto its space. If the number of eigenfaces is N, then each image in this space is presented by an N-dimensional vector.

2DPCA

The 2DPCA [5] works similarly to the Eigenfaces. However, while the Eigenfaces treats each image as a vector, the 2DPCA treats them as matrixes. Then, the 2DPCA tries to find some projection unit vectors X_i such that the best results are obtained when the face matrixes are projected on them.

In 2DPCA, the projection of a face matrix M on a vector X_i resulted in a transformed vector Y_i that has the number of elements equals to the rows of M. If a face matrix has R rows, and the number of projection vectors is P, then the transformed face matrix has a size of RP. In other words, each face image in the 2DPCA method is presented by an N-dimensional vector, in this case $N = RP$.

Local Binary Patterns Histogram

Local Binary Patterns (LBP), which was first introduced by Ojala et al. [9], is used for texture description of images. The LBP operator summarizes the local texture in an image by comparing each pixel with its neighbors. Later, Ahonen et al. proposed LBPH method for face recognition based on the LBP operator [6].

In this method, at first, a face image is converted to LBP image. Each pixel value is computed by its neighbor values. If the center pixel is greater or equal its neighbor value, then denote it with 1 and 0 otherwise. The surrounding pixels yield a binary number for a center pixel (Fig. 3). After that, the image is divided into small areas and histograms are calculated for each area. The feature vector is obtained by concatenating the local histograms. In this case, if an image is divided into A areas, then it is presented by an N-dimensional vector, in this case $N = 256A$ (256 grayscale values).

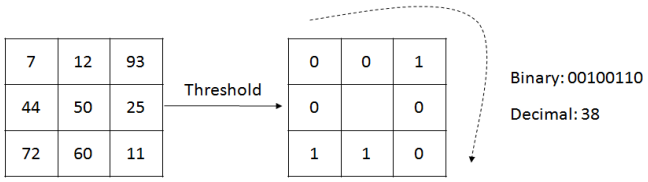

Fig. 3. LBP operator

3.3 Quantization

The purpose of the quantization stage is to convert feature values from continuous domain to discrete domain. At a first glance, this stage may reduces the security of the authentication process significantly by reducing the continuous search space with

infinite elements to a discrete search space with finite elements. However, an informed attacker will understand that biometric authentication is not an exact-matching process, and therefore no need to try every possible values, but only values separated by a threshold. In other words, only finite values are needed to brute-force attacks the continuous template values. Furthermore, we can control the size of the quantized domain by changing the range of the continuous values that mapped to the same quantized value. For these reasons, this stage actually does not affect the security of the authentication system.

Our quantization process works as follow: after normalization, the value of each element of feature vectors is a floating point number in [0, 1]. Then, the quantization process will transform this value to an integer in [0, N], with $N > 0$. Let x be the value before quantization, and x' be the value after quantization, then the quantization formula can be written as (1). *Round* function returns a nearest integer to a parameter.

$$x' = round(xN) \tag{1}$$

3.4 Sketch Generation

The sketch generation stage produces public information s that can be used later to recover quantized template w. Our construction, based on the idea presented in [7] paper, is described below.

The domain of w is [0, N]. We create a codebook where the codewords spread along the range [0, N] and the distance between any pair of neighbor codeword is the same. In particular, the distance between the codeword c_i and c_{i+1} is 2δ where δ is a positive integer. Then, for any value of w in the range $[c_i - \delta, c_i + \delta]$, the mapping function M returns the nearest codeword of w, or $M(w) = c_i$. The sketch generation use the mapping function to return the difference between a value w and its nearest codeword, or

$$SS(w) = w - M(w) \tag{2}$$

The values of the sketch generation $SS(w)$ is in the range $[-\delta, \delta]$ irrespectively of the particular value of w. So, given $SS(w)$, an attacker only knows that the correct value w is in the form $SS(w) + M(w)$. To brute-force attack the system, the attacker needs to try all possible values of $M(w)$, or every codeword. The larger δ is, the smaller the codeword space is. Note that the codeword space is always smaller than the quantized space [0, N]. When $\delta = 1$, the codeword space is three times smaller than the quantized space, and when $\delta = 2$, this number is five times.

3.5 Feature Recovery

Given the authentication input w', the feature recovery stage uses s and w' to reproduce w if the difference between w and w' is smaller than or equal to δ. Call the recovered value w'', it is calculated as

$$w'' = M(w' - SS(w)) + SS(w) \tag{3}$$

To prove the correct w is reproduced when $|w - w'| \leq \delta$, replace $SS(w)$ by the right-hand-side in (2), we have

$$w'' = M(w' - w + M(w)) + w - M(w) \quad (4)$$

If $|w - w'| \leq \delta$, then $w' - w + M(w)$ is in $[M(w) - \delta, M(w) + \delta]$

According to the codebook construction, applying the mapping function on any value in this range will return its nearest codeword, which is also $M(w)$. Substituting the function $M(w' - w + M(w))$ as $M(w)$ in formula (4), we have $w'' = w$.

3.6 Security Analysis

In this section, we consider the security of our proposed construction against the most basic attack: brute-force. Each feature template w can be considered as a vector of N elements. To get the correct w, every element in the vector must be corrected successfully. In previous section, we demonstrated that every codeword must be tried to brute-force attack an element. In current construction, we use the same quantization and codebook for every element regardless of its value distribution. So, assume there are S codewords in a codebook, and then in average, an attacker must try $\frac{S^N}{2}$ cases to get the correct w using a brute-force attack. Of course, the attacker may use the distribution information of the values to reduce the number of searches needed, but it is out of the scope of this paper.

Here are some specific numbers regarding the security of our experiments

- The quantized range is [0, 1000]
- Eigenfaces:
  - We tested with N = 10, 12, 14, 16, 18
  - Codeword space range from S = 20 (with δ = 25) to 200 (with δ = 2)
  - The minimum and maximum security offered are 20^{10} and 200^{18}
- 2DPCA
  - Image height is 200 and the number of projection axes chosen is 1, 2, 3, 4, 5, and 6. So, we have N = 200, 400, 600, 800, 1000, and 1200 respectively
  - Codeword space range from S = 2 (with δ = 250) to 10 (with δ = 50)
  - The minimum and maximum security offered are 2^{200} and 10^{1200}
- LBPH
  - We divide the images into a 5 x 5 grid. So, we have N = 5 x 5 x 256 = 6400
  - Codeword space range from S = 5 (with δ = 100) to 10 (with δ = 50)
  - The minimum and maximum security offered are 5^{6400} and 10^{6400}

4 Experiments

Our proposed constructions are then tested with the Faces94 database [10]. The experiments measures true accept rates (the percentage of times a system correctly accepts a true claim of identity) and true reject rates (the percentage of times a system correct-

ly rejects a false claim of identity) of different recognition algorithms and with different codeword space. The purpose of the experiments is to verify an ability to apply these constructions in real applications with reasonable threshold. We choose images of 43 people, randomly in the Faces94 database. We have 2 sets of images per a person, one for creating feature vectors and one for recovering feature vectors. For each algorithm, we will conduct 43 x 43 tests, in which 43 of them (testing a person with himself/herself) should recover and 1806 (testing a person with others) should not.

4.1 Individual Tests

Eigenfaces

Firstly, we choose mages of 37 people (2 images from each person, 17 are female) are selected randomly from the Faces94 database to create the eigenfaces. They need not to be the same as 43 people in the training set. Next, we use the first set of 43 people to create feature vectors. Then, the other set is used to recover the original feature vectors. For every pair of image, x and y, the scheme try to recover x from y. So, if x and y are the same, the system should recover correctly and if x is different from y, the system should not be able to recover x. The eigenfaces numbers chosen are 10, 12, 14, 16 and 18. And the codeword are chosen with space equally and δ range from 2 to 25. Our true reject rate is always 100%, so we only show the performance in term of true accept rate. The true accept rate is depicted in Fig. 4.

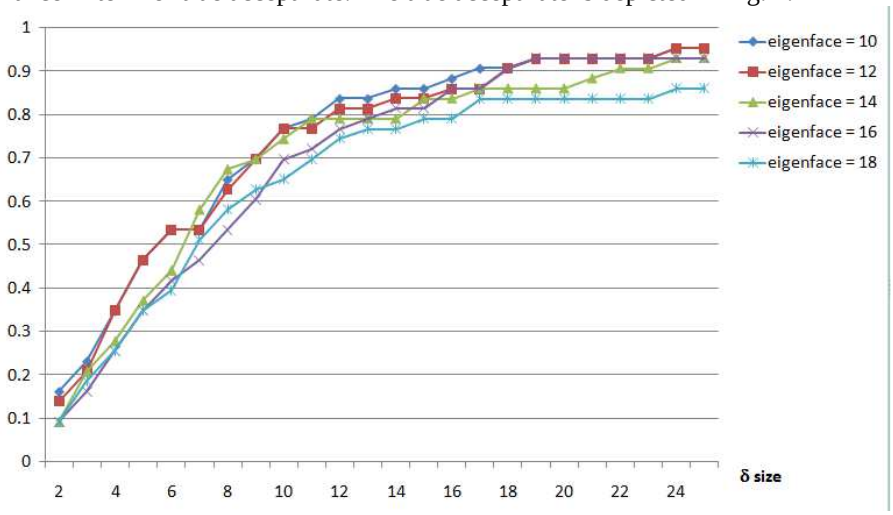

Fig. 4. True accept rate for Eigenfaces algorithm

2DPCA

The settings to measure the true accept rate and true reject rate of this experiment is similar to the settings in Eigenfaces. The number of projection axis chosen is just only 2 to 6, because just the image height is 200 pixels, a large enough dimension

value. And the codeword are chosen with space equally and δ range from 50 to 250. As in the experiment with Eigenfaces algorithm, our true reject rate is always 100%, so we only show the performance in term of true accept rate. The true accept rate of this system is depicted in Fig. 5.

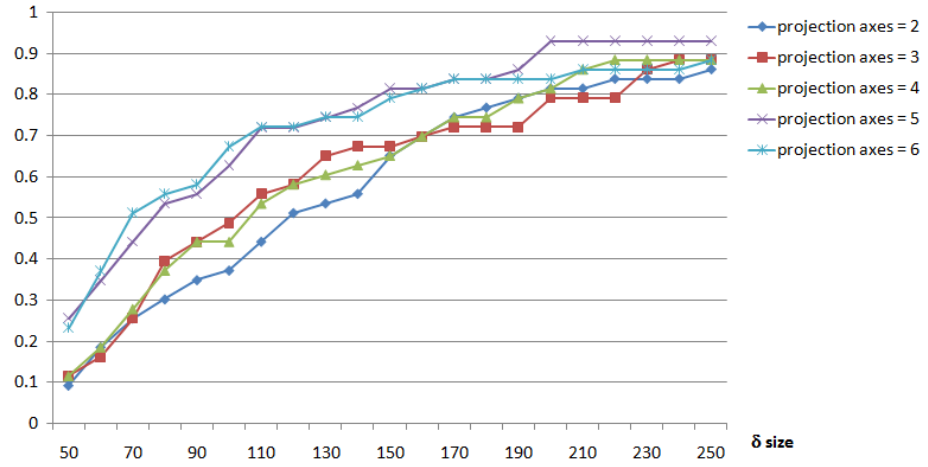

Fig. 5. True accept rate for 2DPCA algorithm

Local Binary Patterns Histograms

For the local histogram algorithm, we divide the face images into a 5 x 5 grid, and for each cell of the grid, there are 256 values for 256 grayscale levels. Hence, the dimension of the feature vector in this case is 5x5x256, which is 6400. The codeword are chosen with space equally and δ range from 50 to 100. The reason we stop at 100 is beyond this value, the true reject rate decrease significantly. Unlike the 2 previous algorithms, the local histogram returns true reject rate less than 100% when the size of δ is more than some threshold. The true accept rate and true reject rate for this algorithm is depicted in Fig. 6

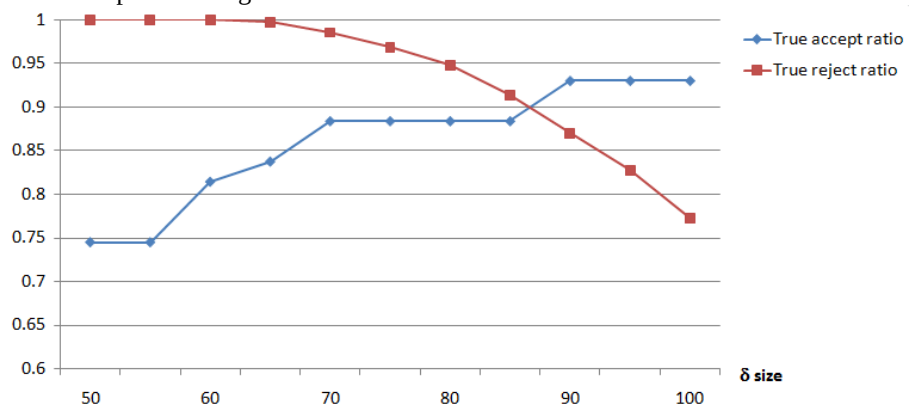

Fig. 6. True accept rate and true reject rate for LBPH algorithm

4.2 Fusion Tests

The working of our face-based secure sketch for Eigenfaces and 2DPCA algorithms both achieve the true reject rate of 100%, but none of them achieve the 100% true accept rate with different experiment parameters. To further experiment if the fusion of these results could improve the overall performance of our construction, we implement a simple fusion method of these 2 algorithms at the decision making level. In this case, we want to improve the true accept rate, so our fusion is the Boolean function OR that will return true when either the Eigenfaces or 2DPCA result matches. Because the Eigenfaces algorithm produces best result when the eigenfaces number is 12, and because the 2DPCA algorithm produces best result when the projection axes are 5, we use these setting in the fusion construction. The δ values for eigenfaces are chosen at 5, 10, 15, 20, and 25; the δ values for 2DPCA are chosen at 50, 75, 100, 125, 150, 175, 200, 225, and 250. The true reject rate of our fusion also return 100%, but there is a significant improvement in the true accept rate of the construction. In fact, when the δ value of the 2DPCA reach 100, the fusion always return 100% true accept rate when selecting the δ value for the Eigenfaces algorithm at 5, 10, 15, 20 and 25.

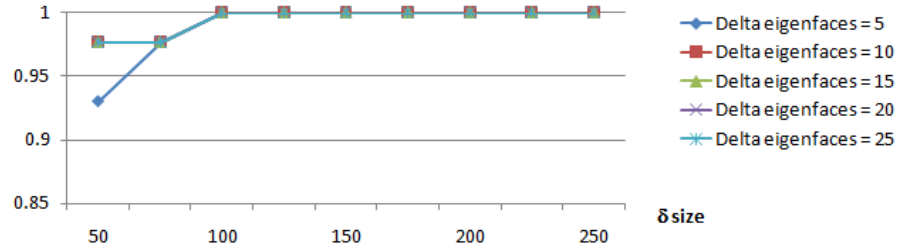

Fig. 7. True accept rate for Fusion test

5 Conclusions and Future Works

In this paper, we present a practical construction of face-based authentication technique with template protection using secure sketch. Although not exactly as the secure sketch proposed in the [2] paper, we demonstrate some security measure of our method in which brute-force is the only attacking technique used. Experiment results show the potential of our construction for using in security application, which the true reject rate is always 100%. The true accept rate of our construction is also increased when a simple fusion technique is applied.

However, more theoretical works is needed to prove, especially the security bound when attackers know about the distribution of the extracted feature values. Furthermore, the construction is also need to be tested on more complex human face database to see how it works. Not only improvement on individual feature authentication, there is a need to improve the fusion method. The fusion now is just a simple Boolean function at the decision level. When the feature is recovered correctly, it can be used to calculate the distance between the original feature and the feature used for authentica-

tion. Using this distance in the fusion may give more choices in designing the final result. And finally, as the development of modern devices, more sensors input are equipped to capture other features, therefore fusion between different biometric features is also a possible way to enhance the system.

Acknowledgements: The authors would like to give special thanks to POSCO, South Korea, for their financial support.

6 References

1. O'Gorman, L.: Comparing Passwords, Tokens, and Biometrics for User Authentication. Proceedings of the IEEE , 91(12), 2021-2040 (2003)
2. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. SIAM J. Comp. 38(1), 97-139 (2008)
3. Wayman, J., Jain, A., Maltoni, D., Maio, D. (eds.): Biometric Systems: Technology, Design and Performance Evaluation. Springer-Verlag, London (2005)
4. Turk, M., Pentland, A.: Eigenfaces for Recognition. J. Cognitive Neuroscience 3(1), 71-86 (1991)
5. Yang, J., Zhang, D., Frangi, A.F., Yang, J-y.: Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 26(1), 131-137 (2004)
6. Ahonen, T., Hadid, A., Pietikainen, M.: Face Recognition with Local Binary Patterns. In: Pajdla, T., Matas, J. (eds.): ECCV 2004. LNCS, vol. 3021, pp. 469-481. Springer-Verlag, Berlin Heidelberg (2004)
7. Juels, A., Wattenberg, M.: A Fuzzy Commitment Scheme. In: 6th ACM Conference on Computer and Communications Security, pp. 28 - 36. ACM, New York (1999)
8. Ross, A., Jain, A.: Information Fusion in Biometrics. Pattern Recogn. Lett. 24(13), 2115-2125 (2003)
9. Ojala, T., Pietikäinen, M., Harwood, D.: A Comparative Study of Texture Measures with Classification based on Feature Distributions. Pattern Recogn. 29, 51-59 (1996)
10. Libor Spacek's Faces94 database, http://cswww.essex.ac.uk/mv/allfaces/faces94.html