One-Time Templates for Face Authentication

Yongjin Lee[†], Yongki Lee[‡], Yunsu Chung[†] and Kiyoun Moon[†]

[†]Biometrics Technology Research Team

Electronics and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, Korea.
{solarone, yoonsu, kymoon}@etri.re.kr

[‡]Electrical Engineering Department
University of California at Los Angeles
Los Angeles, CA 90095-1594, USA.
jfirst@ee.ucla.edu

Abstract

This paper proposes a novel scheme for one-time templates for face authentication. The suggested method changes templates after each completion of authentication like one-time passwords, thus the snapped templates cannot be reused. Matching between a gallery and a probe is performed by Euclidean distance and cosine function as in their primal form without recovering their original templates even though the templates are transformed. The suggested method does not require much modification of conventional face recognition methods and can be built on established systems as an add-on.

1. Introduction

In response to increased security concerns, public interest in biometric systems is increasing. Biometrics utilizes a user's physiological or behavioral characteristics, which are unique, immutable and not sharable. However, like two sides of the same coin, its best forte is its worst foible. The loss or compromise of a user's biometric templates causes serious problems. Since these templates have few substitutes, a user's identity can be permanently stolen.

For the reason, it is recommended that a user's biometric information should be coded by some cryptographic techniques. However, it cannot guarantee that the matching between a gallery and a probe in the encrypted state is consistent because of the randomness and variation of biometric data and the general nature of cryptographic functions, which produces completely different outputs for similar inputs. Therefore, a user's template should be decrypted at a

matching stage and the risk of losing a user's original template still remains.

Ratha et al. have introduced 'cancelable biometrics' as a remedy for the problem of compromised templates [8]. It distorts or transforms a user's template by some non-invertible functions to obscure the user's raw physical characteristics and performs matching in a transformed domain. When a template is compromised, a new biometric template is issued (like a new enrollment of a new user) by distorting the biometric traits in a different way using a new instance of the non-invertible function. However, Ratha et al. have not provided any specific and practical functions in their work even though they have discussed randomization as a general way of producing cancelable templates.

Based on the concept of cancelable biometrics, Ueshige et al. have proposed one-time authentication to protect biometric templates transmitted over networks [11]. Their scheme generates a series of new templates by applying a new instance of transforms repeatedly for each authentication as in one-time passwords based on a hash-chain. They suggested using invertible transforms instead of non-invertible transforms since repeated non-invertible transformations may ruin biometric information in templates such that the transformed templates are not valid for biometric pattern recognition. Even though they described some properties for one-time transforms and a general architecture for the systems, they did not provide any specific methods.

Inspired from the works of Ratha et al. and Ueshige et al., we propose one-time templates for face authentication. A new template can be generated if it is somehow compromised or lost and matching between a gallery and a probe is carried out in a transformed state without revealing their originalities. By applying such transformation repeatedly,

we can generate a different template for each authentication so that a snapped old template cannot be reused. Even though the suggested method generates a new template by transforming an old one, matching between a gallery and a probe is performed by Euclidean distance or cosine function as in their primal form. Because the matching is performed as in their primal form before the transformation, our method can be efficiently implemented on established biometric systems.

We have applied our method to face authentication systems, but it is not restricted to face templates, in fact it can be used for any other biometric template in a real vector form without calibrations.

The rest of the paper is organized as follows. A conventional face authentication method is briefly reviewed in Sec. 2. The proposed method is described in Sec. 3. and 4. Experimental results are given in Sec. 5. Finally, a conclusion is drawn.

2. Face Authentication

In face recognition, feature extraction is one of the most important steps and it converts high dimensional image data into low dimensional feature vectors [1]. Among the methods, 'Eigenface' [10], which uses Principal Component Analysis (PCA), and 'Fisherface' [2], which uses Fisher Linear Discriminant (FLD), are the two most representative ones. Both methods define bases from training data and extract facial features by projecting facial images onto the bases. These extracted features are usually in a real vector form and considered as a user's biometric template. While FLD derives its bases such that the ratio between *between-scatter* and *within-scatter* of the features is maximized, PCA constructs bases such that the variance of extracted features is maximized, implying that original data can be reconstructed from the features with minimum mean square error [3]. Therefore, an authentication system employing eigenface involves some risk of revealing original face images from lost templates and bases.

In general, a user of an authentication system involves two steps for the service, an enrollment and an authentication stage. At an enrollment stage, a gallery, x , is created from a user's face image and registered with an authentication system. At an authentication stage, a user requests to verify his claimed identity and provides his live image. Then, a probe, y , is generated from the acquired image and compared with the registered gallery in order to verify his claimed identity. The two most popular matching methods for the templates are Euclidean distance (Eq. 1) and cosine function (Eq. 2) due to their simplicity.

$$\|x - y\|^2 = (x - y)^T (x - y) \quad (1)$$

$$\cos(x, y) = \frac{x \cdot y}{\|x\| \|y\|} \quad (2)$$

While Euclidean distance gives comparison values as dissimilarity scores, cosine function gives similarity scores based on the angle between two vectors. Thus, a user should receive a smaller matching score than a threshold in order to be accepted as a valid one in an authentication system using Euclidean distance, on the contrary a user should receive a larger matching score in a system employing cosine function.

3. Transformation and Matching

In this section, we propose two similar transforms for biometric template protection, and compare with previously proposed transforms. One transform is used for matching using Euclidean distance and the other one is for matching using cosine function. We describe a normal authentication process with transformed templates and show that matching between transformed templates is performed as in their primal form before the transformation. Based on this section, we will describe a one-time template scheme in the next section.

3.1. Matching using Euclidean Distance

For convenience, we use x as a gallery and y as a probe before transformation and g and p as their corresponding transformed templates.

A transformed templet is created using an orthogonal matrix and a vector which are generated randomly and independently. An orthogonal matrix has properties as shown in Eq. 3.

$$A^T A = A A^T = I \quad (3)$$

where A^T represents the transpose matrix of A and I is an identity matrix with the same size of A .

At an enrollment stage, a transformed template for a gallery, x , is created as written in Eq. 4.

$$g = Ax + b \quad (4)$$

where A is a random orthogonal matrix and b is a random vector.

After storing g , we discard x . Since A and b are random, it is impossible to infer the original template x from g without A and b . If g (or A and b) is lost or compromised, we cancel it and create a new template using a new A and b . A and b are kept with the user on a personal token such as a smartcard. For high security, different A and b should be used for different users and different systems. In the latter case, it prevents templates from being interchanged between different systems. A similar method has

been suggested for face authentication by Kang et al. which shuffles facial features using a permutation matrix only [5]. In contrast, our method is more general and secure since it not only changes the order of features but also garbles their values. Additional pleasing effect using $\mathbf{b}$ will be explained in the next section.

At an authentication step, a user provides his face image and a token storing $\mathbf{A}$ and $\mathbf{b}$. Then, a probe, $\mathbf{p}$, is created as written in Eq. 5.

$$\mathbf{p} = \mathbf{A}\mathbf{y} + \mathbf{b} \quad (5)$$

Matching can be performed directly using Euclidean distance on $\mathbf{g}$ and $\mathbf{p}$ if a user provides accurate $\mathbf{A}$ and $\mathbf{b}$ as shown in Eq. 6. Using this method, the user's template can be kept in a secure state without any risk of revealing its primal template even at the matching stage.

$$\begin{aligned} \|\mathbf{g} - \mathbf{p}\|^2 &= (\mathbf{g} - \mathbf{p})^T (\mathbf{g} - \mathbf{p}) \\ &= (\mathbf{A}\mathbf{x} + \mathbf{b} - \mathbf{A}\mathbf{y} - \mathbf{b})^T (\mathbf{A}\mathbf{x} + \mathbf{b} - \mathbf{A}\mathbf{y} - \mathbf{b}) \\ &= (\mathbf{x} - \mathbf{y})^T \mathbf{A}^T \mathbf{A} (\mathbf{x} - \mathbf{y}) \\ &= (\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y}) \\ &= \|\mathbf{x} - \mathbf{y}\|^2 \end{aligned} \quad (6)$$

For an impostor who provides the wrong $\mathbf{A}$ and $\mathbf{b}$, the difference between $\mathbf{g}$ and $\mathbf{p}$ is larger than between $\mathbf{x}$ and $\mathbf{y}$ due to the mismatch of $\mathbf{A}$ and $\mathbf{b}$. This difference can be adjusted freely by changing the variance of $\mathbf{b}$. To be verified as a valid user, one should provide his face and the correct $\mathbf{A}$ and $\mathbf{b}$.

Mathematically, a distance preserving map implies existence of an inverse map [12]. Therefore, our method can be considered to employ the matching preserving at the cost of the non-invertibility for a practical purpose. As explained, our method does not change genuine matching but matching score values between different persons increase. Thus, biometric systems employing the suggested method do not need to change a threshold. Because of this, our method can be efficiently implemented on established biometric systems without much modification.

3.2. Matching using Cosine Function

For cosine function as a matching measure, we can generate a gallery and a probe using Eq. 7 and Eq. 8 instead of Eq. 4 and Eq. 5

$$\mathbf{g} = \mathbf{A}\mathbf{x} \quad (7)$$

$$\mathbf{p} = \mathbf{A}\mathbf{y} \quad (8)$$

As in case of Euclidean distance, cosine between $\mathbf{g}$ and $\mathbf{p}$ is equivalent to the cosine between $\mathbf{x}$ and $\mathbf{y}$ for a valid

user (Eq. 9).

$$\begin{aligned} \cos(\mathbf{g}, \mathbf{p}) &= \frac{\mathbf{g} \cdot \mathbf{p}}{\|\mathbf{g}\| \|\mathbf{p}\|} \\ &= \frac{\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{y}}{\sqrt{\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}} \sqrt{\mathbf{y}^T \mathbf{A}^T \mathbf{A} \mathbf{y}}} \\ &= \frac{\mathbf{x}^T \mathbf{y}}{\sqrt{\mathbf{x}^T \mathbf{x}} \sqrt{\mathbf{y}^T \mathbf{y}}} \\ &= \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} \\ &= \cos(\mathbf{x}, \mathbf{y}) \end{aligned} \quad (9)$$

3.3 Comparison with Related Works

As mentioned in the previous subsection, our method is more general than the method by proposed by Kang et al., which shuffles facial features using a permutation matrix only [5]. In addition, their method has been suggested only for Euclidean distance as a matching measure. Moreover, even though they have insisted that their authentication scheme is two factor, meaning that a user should provide his face image and private transform in order to be verified as a genuine, a user can pass the system using biometric information only. The reason is that differences between two distinct orthogonal matrices including permutation are small and the added variation is not significant compared to the variance of facial features since the norm of row and column vectors of an orthogonal matrix is 1. In other words, the differences between two distinct orthogonal matrices are suppressed by the variation of biometric data. This discussion will be more evident in the experiment section.

The other similar method was used in BioHashing proposed by Teoh et al. [9]. They have suggested using a rectangular matrix with orthonormal vectors to randomize a user's original template. A rectangular matrix with orthonormal vectors is non-invertible, thus it can protect an original template secure but does not preserve matching. Even though they have showed that matching scores before and after the transformation with the rectangular matrix are statistically identical [4], their method cannot be used for one-time templates. This is because repeated non-invertible distortion ruins biometric information in templates gradually such that the transformed templates are not valid for biometric pattern recognition. This means that an impostor can pass the authentication system using a lost private transform even without valid biometric data. Such similar flaw in BioHashing has been already reported in [6]: Just one applying BioHashing destroys almost all biometric information in an original template and an impostor can pass the system using a lost transform only.

Table 1. Protocol Flow for One-Time Templates.

1.	A user sends a probe p_t .
2.	A server verifies p_t . If it is a valid user, the server sends R_t and $H(K_t \ R_t)$ to the user, and updates g_t using K_t , and K_t with $K_{t+1} = H(K_t \oplus R_t)$. ¹
3.	A user also updates A_t and b_t using K_t , and K_t with K_{t+1} after checking the validity of $H(K_t \ R_t)$.

4. One-Time Templates

Based on the work described in the previous section, we propose one-time templates for the security against to the replay attack. A new template can be created in sequel by changing A and b . In the section, we consider the case using Euclidean distance as a matching measure, but the case of cosine function is not much different.

Just before t th authentication, an authentication sever has g_t and a secret number, K_t , and a user has A_t , b_t and K_t . At t th authentication, a user provides his face image and a token storing A_t and b_t , then t th probe, p_t , is created as shown in Eq. 10 and sends it to the authentication sever.

$$p_t = A_t y + b_t \quad (10)$$

Then, the authentication sever performs a matching process. Matching can be performed directly using Euclidean distance on g_t and p_t if a user provides accurate A_t and b_t as explained in the previous section. After t th authentication is performed successfully, the authentication sever creates $t + 1$ th gallery from the previous one as shown in Eq. 11 and updates K_t .

$$g_{t+1} = A'_t g_t + b'_t \quad (11)$$

where A'_t is a new random orthogonal matrix and b'_t is a new random vector, which are generated from K_t . Details on generating A'_t , b'_t and updating K_t will be described shortly later. The initial template for Eq. 11 is created according to Eq. 4 as described in the previous section when a new user registers at the authentication sever and it should be created independently of initial K . Even though the authentication sever keeps g_t and K_t , the previous gallery, g_{t-1} , and the original gallery, x , cannot be guessed from the lost information from the sever.

After the user confirms the authentication, he generates a new transform, A_{t+1} and b_{t+1} , as shown Eq. 12 and store it on his personal token.

$$\begin{aligned} A_{t+1} &= A'_t A_t \\ b_{t+1} &= A'_t b_t + b'_t \end{aligned} \quad (12)$$

¹ R_t is a random number, H is a hash function, $\|$ is a concatenation function of two strings and $\oplus$ denotes exclusive OR.

A matrix generated from multiplication of two orthogonal matrices is again orthogonal, thus the matching preserving property discussed in the previous section is still conserved. Using K_t kept with a user, A'_t and b'_t are created in the same way with generating the new gallery but independently of the server. After that, a user also updates K_t .

Elements of A'_t and b'_t are generated based on K_t . To simplify implementation, using K_t as a state number (or a seed number) for a random generator, we generate a set of random numbers in correspondence with each element of A'_t and b'_t . Since for the same state number, the same set of random numbers is generated from a random number generator, a user and an authentication sever create the same elements of A'_t and b'_t based on K_t . Then, the numbers corresponding to A'_t are orthogonalized and the numbers corresponding to b'_t are normalized properly.

The protocol flow for one-time templates is summarized in Table. 1. We have modified SRAC (Semi-Randomized Access Control) [7] and applied it to one-time templates. Since the values of p_t and g_t change every time according to the protocols, the replay attack can be prevented. R_t is used to update K_t . The reason that the server sends R_t with $H(K_t \| R_t)$ is to prevent malicious change of R_t . A user can check the validity of R_t by comparing the value of $H(K_t \| R_t)$. Another security property we can achieve by using a hash function is the forward secrecy. Since the chain of K_t is made by a hash function, predicting K_{t-1} by K_t is computationally infeasible.

5. Experiments

We performed face verification experiments to confirm our method. We used our own face database which consists of 55 people and 20 images per person. 10 images for each person were used for training (gallery), and the rest were used for testing (probe). One gallery per person was created by taking the mean of 10 templates. Face features were extracted using eigenface [10] and fisherface [2] without any preprocessing. Matching was performed using Euclidean distance.

Fig. 1 and 2 show distributions of matching scores. In the figures, 'genuine' and 'impostor' denote the matching

between the same person and between different persons, respectively. 'Temporal' denotes the matching between g_{t-1} and g_t of the same person. In each figure, (a) is authentication results for normal templates without employing a private transform, (b) and (c) are authentication results with one-time templates employing a permutation matrix only, (d) and (e) are authentication results with one-time templates employing an orthogonal matrix only, and (f) and (g) are authentication results with one-time templates employing the proposed transform.

Fig. (b) and (d) show that the two distributions are overlapped and a user may pass an authentication system by providing an arbitrary transform. It means that it is difficult to say that the method proposed by Kang et al. [5] is a two factor scheme. Similarly, (c) and (e) show that an old template can be reused to be verified as a valid user. This is because differences between two distinct orthogonal matrices are small and the added variation was not significant compared to the variance of facial features.

In Fig. (f) and (g), the two score distributions are completely separated and the proposed method could achieve 0% EER (Equal Error Rate). Fig. (f) shows that a user cannot pass an authentication system without an accurate transform since the differences between two distinct transforms from two different users always exceed the range of genuine matching scores. Similarly, Fig.(g) shows that an old template cannot be reused to be accepted as a genuine. In fact, these experimental results prove that a random vector, b , plays a more important role than an orthogonal matrix in a two factor scheme.

Note that in the proposed method, even if an impostor obtains a genuine's private transform, to be authenticated as a genuine, he should provide a biometric template which an original biometric system accepts as a genuine template. This is because a genuine distribution does not change after transformation and when a private transform is disclosed, an impostor score distribution returns to the original one before transformation, Fig. (a). This is significant distinction from BioHashing [9] where an impostor can pass an authentication system using a lost transform only [6]. The reason is that when a private transform is revealed, a genuine and an impostor distribution do not revert to the original ones and the overlapped area between the two distributions becomes larger than before applying BioHashing.

6. Conclusions

In the paper, we proposed a novel scheme for one-time templates for face authentication. Our work has been inspired partly by Ueshige et al., however they have only proposed a general idea and have not provided any specific transforms for cancelable biometrics nor for one-time templates [11]. Unlike the original concept of cancelable

biometrics, we employed invertible transforms instead of non-invertible transforms as discussed by Ueshige et al. In the case of one-time templates, invertible (one-to-one) transforms are more desirable since repeated non-invertible transformations destroy biometric information in templates such that the transformed templates are not valid for biometric authentication any more. Also, it means that if the information in biometric templates are ruined by some non-invertible transform, an impostor can pass the authentication using a lost private transform only. However, in our scheme an impostor should provide valid biometric information even though he obtains a valid private transform.

Acknowledgment

This work was supported by the IT R&D program of MIC/IITA. [2007-S-020-01, Development of Privacy Enhanced Biometric System]

References

- [1] M. Bartlett. *Face Image Analysis by Unsupervised Learning*. Kluwer Academic Publisher, 2001.
- [2] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711–720, Oct. 1997.
- [3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.
- [4] A. T. B. Jin. Cancellable biometrics and multispace random projections. In *Computer Vision and Pattern Recognition Workshop*, June 2006.
- [5] J. Kang, D. Nyang, and K. Lee. Two factor face authentication scheme with cancelable feature. In *IWBRS, LNCS 3781*, pages 67–76, 2005.
- [6] A. Konga, K.-H. Cheunga, D. Zhanga, M. Kamelb, and JaneYoua. An analysis of biohashing and its variants. *Pattern Recognition*, 39(7):1359–1368, July 2006.
- [7] Y. Lee and I. Verbauwhede. Secure and low-cost rfid authentication protocols. In *2nd IEEE International Workshop on Adaptive Wireless Networks (AWiN)*, November 2005.
- [8] N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(1):614–634, January 2004.
- [9] A. B. J. Teoh, D. C. L. Ngo, and A. Goh. An integrated dual factor authenticator based on the face data and tokenised random number. In *ICBA 2004, LNCS 3072*, pages 117–123, 2004.
- [10] M. A. Turk and A. P. Petland. Face recognition using eigenface. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Maui, Hawaii, July 1991.
- [11] Y. Ueshige and K. Sakurai. A proposal of one-time biometric authentication. In *Proc. of SAM 2006*, Las Vegas, Nevada, USA, June 2006.
- [12] Wikipedia. <http://en.wikipedia.org/wiki/isometry>.

(a) Normal template, EER = 18.18%

(b) Permutation matrix, EER = 14.70

(c) Permutation matrix, EER = 24.55%

(d) Orthogonal matrix, EER = 14.69%

(e) Orthogonal matrix, EER = 24.73%

(f) Proposed method, EER = 0%

(g) Proposed method, EER = 0%

Figure 1. The distributions of matching scores with Eigenface

(a) Normal template, EER = 6.18%

(b) Permutation matrix, EER = 5.67%

(c) Permutation matrix, EER = 7.09%

(d) Orthogonal matrix, EER = 5.64%

(e) Orthogonal matrix, EER = 7.27%

(f) Proposed method, EER = 0%

(g) Proposed method, EER = 0%

Figure 2. The distributions of matching scores with Fisherface