Two Factor Face Authentication Scheme with Cancelable Feature

Jeonil Kang[1], DaeHun Nyang[1], and KyungHee Lee[2]

[1] Information Security Research Laboratory, INHA University*
dreamx@seclab.inha.ac.kr, nyang@inha.ac.kr
http://seclab.inha.ac.k
[2] Department of Electrical Engineering, University of Suwon
khlee@suwon.ac.kr

Abstract. Though authentication using biometric techniques has conveniences for people, security problems like the leakage of personal bio-information would be serious. Even if cancelable biometric is a good solution for the problems, only a few biometric authentication scheme with cancelable feature has been published. In this paper, we suggest a face authentication scheme with two security factors: password and face image. Using matching algorithm in the permuted domain, our scheme is designed to be cancelable in the sense that templates that is composed of permutation and weight vector can be changed freely.

1 Introduction

Many biometirc technologies have been studied to protect bio-information from stolen. Generally "matching" is process that it compares a template with acquired image, and thus, bio-information may be recoverable from templates. However, this property works as week point in security side.

Therefore, N.K. Ratha's "Cancelable Biometric"[1] concept which can cancel encoded information from database is needed. Database has information that is encoded as a template, and acquired information become encoded by the same function as database. These original and acquired information are compared in encoded domain, and the result is the same as the result of comparison in non-encoded domain. It is good for the encoding function to have a feature of non-invertible(or one-wayness). Since, however, generally this encoding processing is also based on image processing, it is too hard to apply fully random permutation in bit. Hence, it is still too hard to use cryptographic method. Mario Savvides's scheme is an example of cancelable biometric[8].

M. Braithwaite proposed an idea that matching is performed in transformed domain by some function, where they did not define what the functions are[2]. In this paper, we suggest an instance of their concept. We propose a cancelable biometric authentication scheme by combining PCA(Principal Component Analysis)[3]-[6] and PBKDF(Password-Based Key Derivation Function)

* This work was supported by INHA UNIVERSITY Research Grant

S.Z. Li et al. (Eds.): IWBRS 2005, LNCS 3781, pp. 67–76, 2005.

method[7]. Basically, this scheme works on local system, but we expect that it can be applied to authentication through remote system.

2 Face Recognition Using Eigenfaces

In 1991, Mattew A. Turk published face recognition scheme using eigenface which is based on PCA[3][4]. PCA is a way of identifying patterns in data, and expressing the data in such a way as to highlight its similarities and differences[6]. Eigenface is a face space that can be obtained by analyzing sample face images with PCA method, and it contains features for identifying whether an acquired image is a face or whos face is if the image is a face.

Let the training set of face images be $\Gamma_1, \Gamma_2, \cdots, \Gamma_M$. The average face of the set is defined by $\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$ and different faces from the average face are defined as $\Phi_i = \Gamma_i - \Psi$. And then we can get covariance matrix C about face vector.

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T \tag{1}$$

where the matrix $A = [\Phi_1\Phi_2\cdots\Phi_M]$. If size of face image is $N \times N$, the matrix C is $N^2 \times N^2$. Since the N^2 eigenvectors and eigenvalues are computationally infeasible and hard to handle, it has to modify the matrix C to $M \times M$ matric using mathematical trick. M is much smaller than N^2. (You can see more detail processes in [4].) For now, u_i denotes the eigenvectors for $i = 1, \cdots, M'$, where M' is a value that is gotten for the face detection heuristically ($M' \leq M$), and $U = [u_1u_2\cdots u_{M'}]$.

Each different face image Φ_k can be represented as a linear combination of the eigenvectors u_i.

$$\Omega_k = U^T\Phi_k = U^T(\Gamma_k - \Psi) \tag{2}$$

where $k = 1, 2, \cdots, N_c$ and N_c is the number of face classes. Each normalized training face Φ_k is represented by a vector $\Omega_k = [w_1^k, w_2^k, \cdots, w_{M'}^k]^T$.

Given an unknown face image Γ is normalized and projected on the eigenspace.

$$\Omega = U^T(\Gamma - \Psi) \tag{3}$$

A vector $\Omega = [w_1w_2\cdots w_{M'}]^T$ means the contribution of each eigenface in representing the input face image. The simplest method for determining which face class provides the best description of an input face image is to find the face class k that minimizes the Euclidean distance which is below some chosen threshold θ_ϵ.

$$\epsilon_k = \| \Omega - \Omega_k \| \tag{4}$$

If $\epsilon_k > \theta_\epsilon$, the face is classified as "unknown."

3 Principles of Macthing in Permuted Domain

In this section, we describe our face authentication scheme with cancelable feature. The cancelable feature is implemented by performing matching procedure in permuted domain. Biometrics information is stored at authentication system in permuted form and also the permutation itself is not stored anywhere. Thus, even though the user template is exposed to an attacker, it cannot be used for the attacker to be authenticated in other systems. Also, he cannot be authenticated even to the very system where the template has been stolen if the password that is used for generating the permutation is not exposed. Thus, we can say that our system is two factor authentication scheme using password and face image.

If the user want to cancel the biometrics information, he can do it by simply changing the password. More precisely, by changing password, user cannot change the biometrics information but he can chang the authentication information stored at the system. Even though the attacker gets the original face image, he cannot succeed in being authenticated in some other place unless he also can find out the password at each site.

3.1 Face Recognition in Permuted Domain

In the scheme using eigenface, average face Ψ, eigenvectors U and each weighted factor Ω_k can be represented in matrix form. They can be pre-computed and restored in a database for face recognition system. If the scheme using eigenface is used in an authentication system and Ω_k are exposed, someone might be able to be authenticated as specific authorized person by making ϵ to zero using exposed Ω_k. To do this, he must offer Γ_k to the system because the system computes Ω_k from Γ_k received.

However, if Ψ, U, and Ω_k are exposed in one scoop, an attacker can obtain face images from equation (2).

$$\Gamma_k = (UU^T)^{-1}U\Omega_k + \Psi \tag{5}$$

Thus, we need to store them in database only after they are securely encoded with some method such as cryptographic hash functions. However, cryptographic operation prevents face authentication algorithm from matching face exactly. To solve this problem, we need some transformation that does not effect matching procedure even in the transformed domain. In the authentication scheme using eigenface, permutation is a good choice.

Definition 1. *Lets $f_{c,k}(A)$ and $f_{r,k}(A)$ be some functions that permute some matrix A in the same sequence by column and by row respectively. Here, k means a user index.*

Let $P_{c,k}$ and $P_{r,k}$ be some $N \times N$ permutation matrices and A be $N \times 1$ matrix. And then, we can represent these functions by

$$f_{c,k}(A) = P_{c,k}A \tag{6}$$

$$f_{r,k}(A) = P_{r,k}A \tag{7}$$

Also, $P_{c,k}$ and $P_{r,k}$ have a relation of

$$P_{c,k}^T = P_{r,k} \tag{8}$$

where T denotes the transpose of a matrix.

Lemma 1. *If there is relationship between given face image Γ, average face image Ψ and eigenvectors U as (3), it satisfies*

$$U^T(\Gamma - \Psi) = f_{c,k}(U^T)\{f_{r,k}(\Gamma) - f_{r,k}(\Psi)\} \tag{9}$$

Proof. Let A and B be $N \times 1$ matrices, D be $M \times N$ matrix. And then, because of the characteristic of matrix arithmetic, we can say

$$f_{r,k}(A - B) = P_{r,k}(A - B) = P_{r,k}A - P_{r,k}B = f_{r,k}(A) - f_{r,k}(B) \tag{10}$$

Also, when we multiply D and A, D_{ij}(an element of D of i-th row and j-th column) is always multiplied by A_i(i-th element of A) because they have the same permutation sequence, even though D and A are permuted by $f_{c,k}()$ or $f_{r,k}()$. Thus,

$$DA = f_{c,k}(D)f_{r,k}(A) \tag{11}$$

From equations (10) and (11), we can obtain

$$D(A - B) = f_{c,k}(D)f_{r,k}(A - B) = f_{c,k}(D)\{f_{r,k}(A) - f_{r,k}(B)\} \tag{12}$$

□

According to **Lemma 1**, now, the system can accept permuted Γ_k instead of non-permuted one to match face images. Note that because Ω_k has different dimension from other matrices (e.g. U has $N^2 \times M'$ and Ψ and Γ_k have $N^2 \times 1$, but Ω_k has $M' \times 1$), it cannot be permuted by $f_{c,k}()$ or $f_{r,k}()$.

If the system has been hacked and Ω_k has been exposed, someone may worry about security problems caused by his or her carelessness that Ω_k is not permuted and the possibility that different databases have the same Ω_k values for each person. However, since Ω_k is decided along face image Γ_k, eigenvectors U, and average face Ψ, if different systems have different U and Ψ, Ω_k of one system is different from that of other systems.

Mattew A. Turk said in his paper[3] "In many of our test cases, based on $M = 16$ face images. $M' = 7$ eigenfaces were used. The number of eigenfaces to be used is chosen heuristically based on the eigenvalues." If $M' = 7$ and Ω_k are permuted, all of the possible permutation of Ω_k are $5040(= 7!)$ cases, and there is no value for security. To increase security strength, thus, it has to use more face images for training and more eigenvalues about minimum 34 or 35. ($34! < 2^{128} < 35!$) From that reason, Ω_k is not permuted in this paper.

3.2 Derivation of Permutation Matrix from User Password

Because the system does not have any information about the permutation itself, that is, face image acquiring device must send permuted face image to the system. To make permutation on the face image, acquiring device gets a password from user and provides the permutation matrix.

To obtain secure permutation matrix, user password must be changed to a string with high entropy by cryptographic method like PBKDF(Password-Based Key Derivation Function) in PKCS#5 [7].

Definition 2. *$D_K()$ denotes password-based key derivation function, and this function returns a random key string.*

$$\textit{random key string} = D_K(\textit{user password})$$

$D_K()$ is fully based on PBKDF in PKCS#5. Basically functions of PKCS#5 consist of hash functions or pseudo random generator functions, and it is infeasible to compute the inverse.

Definition 3. *$D_{M_x}()$ denotes permutation matrix derivation function from random key string, and this function returns a column-permuted matrix or a row-permuted matrix according to x. x can be r or c.*

$$\textit{permutation matrix} = D_{M_x}(\textit{random key string})$$

A permutation matrix is generated the unit matrix I by permuting by column or row using to the random key string RKS. Let $RKS_{<i..j>}$ (described as `RKS<i..j>`) be a bit sequence (can be recognized as integer) from the i-th bit to the j-th bit of random key string. Then, permutation matrix derivation function $D_{M_x}()$ works as follows.

```
// RKS : random key string, input
// n : image height * image width
// m : larger integer then upper bound lg n

loop i := 0 to n
    j := RKS<i..i+m-1>
    if j >= n
        j := j % n
    if x = r
        swap(the i-th row of I, the j-th row of I)
    else if x = c
        swap(the i-th column of I, the j-th column of I)

return I
```

With $D_{M_x}()$, we can obtain a permutation matrix $P_{c,k}$ and $P_{r,k}$, and we can compute permutation function $f_{c,k}()$ or $f_{r,k}()$. We can rewrite (6) and (7) by

$$f_{x,k}(A) = D_{M_x}(D_K(\textit{user password}))A \tag{13}$$

Also, we can extract the *inverse permutation matrix* which can be computed from the original matrix using the same key, RKS easily.

4 Face Authentication Scheme with Cancelable Biometric

4.1 System Requirement

If the system stores permuted U^T and permuted Ψ only, eigenface-based matching algorithm works well with permuted acquired face images. In this case, however, memory required to store user specific information is quite large. So, the system stores only one global U^T_{RR} and Ψ_{RR} that are permuted with some *master random key*, and then only user specific information required to store is just one permutation per user. By doing this, we can save lots of memory. Global variables shown in below are secured by being permuted with permutation matrix from MRK(*Master Random Key*) that is never stored in the system and never to be obtained by any method.

Table 1. Global Variables and the explanation

Var	Explanation
U^T_{RR}	is a set of eigenvectors which is permuted by permutation matrix derived from MRK using $D_{P_c}()$.
Ψ_{RR}	is average face image which is permuted by permutation matrix derived from MRK using $D_{P_r}()$.

Each user account shown in below is consisted of user specific weight factor and permutation matrix.

Table 2. Account Variables and the explanation

Var	Explanation
Ω_k	is a user specific weight factor. It indicates how to reconstruct user's original face image using eigenvector U and average face Ψ.
$S_{x,k}$	means *User Specific Permutation* which can be used in face recognition process. It makes global variables to user specific variables when it is multiplied by global variables. Optionally, it can be compressed to small size because it has many zero elements. Also, $S^T_{c,k} = S_{r,k}$.

Consequently, we show some variation of previous equations here.

$$U^T_{RR} = D_{M_c}(MRK)U^T \tag{14}$$

$$\Psi_{RR} = D_{M_r}(MRK)\Psi \tag{15}$$

If matrix $f_{c,k}(U^T)$ and $f_{r,k}(\Psi)$ are user specific matrices permuted with permutation matrix from user passwords, then

$$f_{c,k}(U^T) = D_{M_c}(D_K(user\ password))U^T = S_{c,k}U^T_{RR} \tag{16}$$

$$f_{r,k}(\Psi) = D_{M_r}(D_K(user\ password))\Psi = S_{r,k}\Psi_{RR} \tag{17}$$

Thus, user specific data required to store are $S_{x,k}$ and Ω_k.

4.2 Face Recognition

When a user wants to be authenticated at the system that adopts our scheme, he should offer his face image and his password to "acquiring part." Let that face image be Γ, and then "face image permutation part" should compute permuted face image Γ_P using his password.

$$\Gamma_P = D_{M_r}(D_K(user\ password))\Gamma \tag{18}$$

After that, it is sent to "main process part" which tries to find user's weight factor Ω by searching in the user accounts. For each user account, main process part should compute candidates of Ω, Ω'_k.

$$\Omega'_k = f_{c,k}(U^T)\{\Gamma_P - f_{r,k}(\Psi)\} = S_{c,k}U^T_{RR}\{\Gamma_P - S_{r,k}\Psi_{RR}\} \tag{19}$$

And, it might be able to find a legal user account by comparing these Ω'_k vectors with user specific weight factor Ω_k.

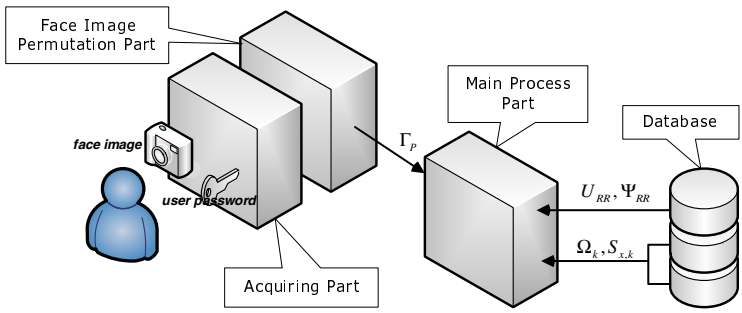

Fig. 1. Face Recognition Process. This diagram has been illustrated with the assumption that our scheme works in local system. At remote system, it will be separated acquiring and face image permutation part from other parts as client device

Since each account has different permutaion sequence, Euclidean distance between Ω'_k and Ω_k might have a large gap between the right account and wrong accounts.

4.3 Canceling Template

A template in our scheme is defined to be ($S_{x,k}$, Ω_k). Canceling template, thus, is to change $S_{x,k}$. The process to change password requires some security for preventing leakage of user face image during the the transmission, if the process is performed remotely. When user wants to change his password by security

risk, he must offer his password to the system to recompute $S_{x,k}$. However, it is too dangerous to offer in form of raw *user password*, $D_K(user\ password)$ or $D_{M_r}(D_K(user\ password))$ all. Because, $\Gamma = P_r^{-1} \cdot \Gamma_P$ and Γ_P can be obtained by observing authentication process. Thus, we assume that the password information is encrypted by session key safely.

Moreover, we must consider the case that a new password is adapted. If U^T and Ψ were recovered during this process, it is dangerous because U^T and Ψ might be able to be exposed to someone. Thus, the following pre-computed matrix should be offered for preventing the recovery.

$$P_{new}(P_{old})^{-1} = D_{M_r}(D_K(new\ password))D_{M_r}^{-1}(D_K(old\ password)) \quad (20)$$

$D_{M_x}^{-1}()$ denotes the inverse permutation matrix function that computes the inverse matrix from *random key string*. $P_{new}(P_{old})^{-1}$ cannot be separated and it changes $U_{P_{old}}^T$ to $U_{P_{new}}^T$ and $\Psi_{P_{old}}$ to $\Psi_{P_{new}}$ directly.

And then, the system must compute USP_k newly using U_{RR}^T and $U_{P_{new}}^T$, or Ψ_{RR} and $\Psi_{P_{new}}$. Because of (17),

$$S_{c,k} = U_{P_k}^T(U_{RR}^T)^{-1} = (\Psi_{P_k}(\Psi_{RR})^{-1})^T = S_{r,k}^T \quad (21)$$

5 Security Analysis

An attacker that does not know the right faces of the user who participates in a system might try to obtain user face image, or would like to obtain permitted authority to deal with the system.

Attack 1. *User face recovery from Γ_P:* An attacker might try to recover an user face image from a permuted face image Γ_P. However, this is infeasible because size of image is too large about 400(=20×20) or 900(=30×30). Trials complexity in such like this case can be about 400! or 900!. This make it too hard to get a good result in some months (or years).

Attack 2. *User face recovery from $U_{RR}^T \cdots$:* A hacker might be able to obtain U_{RR}^T, Ψ_{RR}, Ω_k, and $S_{x,k}$ by intrusion to the system. Using (5), he might try to recover user's face. However, $(U_{RR}U_{RR}^T)^{-1}U_{RR}\Omega_k + \Psi_{RR}$ is the same as permuted face image Γ_P.

Attack 3. *Raw U or Ψ recovery from $U_{RR}^T \cdots$:* If an attacker can recover a raw U or Ψ, he might be able to recover the user face images. Surely, they are not good images to see, because U does not contain whole information about training image set, but it has enough information to be authenticated. However, this is also infeasible because the size of the matrix is too large as like face image Γ_P.

Even though our scheme does not allow the leakage of face images, an attacker might be able to obtain face image of user who participates in system from other sources. Then, the attacker might try to do these attacks.

Attack 4. *Dictionary attack against user password:* Our scheme seems to be able to get user password by trying dictionary attack. However, it is too hard to attack in a local system: an attacker must enter the dictionary words one by one. Additionally, the system manufacturer might use some methods that can delay dictionary attack.

Attack 5. *Discovery of the password creates similar permutation matrix:* If an attacker knows similar permutation matrix in which a few rows or columns are different from original one, he might enter the system legally because of the characteristic of Euclidean distance. However, this is infeasible, even though it reduces the size of the searching space greatly, but it is almost 300-398! or 800-898!. Thus, he may expect to find out the password which creates similar permutation matrix in his dictionary. If he can find it, he can save much time to do dictionary attack. Unfortunately, he can find it in a probability of $|Dic| \times (N^2 - ds)!^{-1}$, where $|Dic|$ denotes the size of the dictionary and ds denotes the limit value of decision about "the similar". This probability is due to the difference of search spaces: dictionary space ($= |Dic|$), random key string space ($= 2^{N^2+log_2 N^2}$), and permutation space ($= N^2!$).

Attack 6. *Discovery of other users' face images:* If an attacker succeeds in stealing the whole database from the authentication system, and by some means, attacker finds out a user's password, he can recover U and Ψ after computing inverse permutation from the password. Once he obtains U and Ψ, it is very easy task to compute all other users' original face images by evaluating equation (5). However, even if we assume this system cracking situation, attacker cannot use the original face image to be authenticated to other systems and even to the very system cracked unless he does know exactly the corresponding password at each system.

6 Conclusion

In this paper, we proposed a cancelable face authentication scheme based on eigenface methodology. Our scheme consists of two factors for strengthening the security weakness: comparison in the permuted domain and permutation sequence generated from user password.

After that, we thought and illustrated some attack methods that might be able to break our scheme, and we could not find any reasonable weak points from our scheme under the assumption that user face image can not be exposed by hacking the system. Even if a user face image was exposed from other sources, an attacker cannot attempt a successful attack that might collapse the system. Moreover, we suggested the method by which we can change user password safely.

In order to make the application with a good performance and strong security, more and deeper researches and verifications should be required, especially about U and Ψ as the result of training face images because U and Ψ have different

pattern in different training group(at the ages, countries, races, etc.) and it should have flexibility in the training steps.

References

1. N.K. Ratha, J.K. Connell, and R.M. Bolle: Enhancing security and privacy in biometrics-based authentication systems, IBM Systems Journal, vol. 40, no. 3, pp. 614-634, 2001
2. Michael Braithwaite, Ulf Cahn von Seelen, James Cambier, John Daugman, Randy Glass, Russ moore, and Ian Scott: Application-Specific Biometric Templates, Iridian Technologies Inc., Proceedings of AutoID, pp. 167 - 171, 2002
3. Mattew A. Turk and Alex P. Pentland: Face Recognition Using Eigenfaces, Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on 3-6 pp. 586 - 591, June 1991
4. Mattew A. Turk and Alex P. Pentland: Eigenfaces for Recognition, Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71-86, 1991.
5. Zhuji and Y.L. Yu: Face recognition with eigenfaces, Industrial Technology, 1994. Proceedings of the IEEE International Conference on 5-9, pp. 434 - 438, Dec. 1994
6. Lindsay I Smith: A tutorial on Principal Components Analysis, February 26, 2002
7. PKCS#5 v2.0: Password-Based Cryptography Standard, RSA Laboratories, March 25, 1999
8. Mario Savvides, B.V.K. Vijaya Kumar, and P.K. Khosla: Cancelable biometric filters for face recognition, Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on Vol.3, pp. 922- 925, Aug. 2004
9. Anil K. Jain, Umut Uludag, and Rein-Lien Hsu: Hiding a Face in a Fingerprint Image, Proc. of ICPR, Aug. 2002
10. http://www.biometrics.org, Biometric Consortium