Secure Face Authentication Framework in Open Networks

Yongjin Lee, Yongki Lee, Yunsu Chung, and Kiyoung Moon

In response to increased security concerns, biometrics is becoming more focused on overcoming or complementing conventional knowledge and possession-based authentication. However, biometric authentication requires special care since the loss of biometric data is irrecoverable. In this paper, we present a biometric authentication framework, where several novel techniques are applied to provide security and privacy. First, a biometric template is saved in a transformed form. This makes it possible for a template to be canceled upon its loss while the original biometric information is not revealed. Second, when a user is registered with a server, a biometric template is stored in a special form, named a 'soft vault'. This technique prevents impersonation attacks even if data in a server is disclosed to an attacker. Finally, a one-time template technique is applied in order to prevent replay attacks against templates transmitted over networks. In addition, the whole scheme keeps decision equivalence with conventional face authentication, and thus it does not decrease biometric recognition performance. As a result, the proposed techniques construct a secure face authentication framework in open networks.

Keywords: Face authentication, cancelable biometrics, one-time template.

Manuscript received Mar. 15, 2010; revised July 12, 2010; accepted Sept. 20, 2010.

Yongjin Lee (phone: +82 42 860 1699, email: solarone@etri.re.kr), Yunsu Chung (email: yoonsu@etri.re.kr), and Kiyoung Moon (email: kymoon@etri.re.kr) are with the Software Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Yongki Lee (corresponding author, email: yongki93.lee@samsung.com) is with the Digital IP Development Team, Samsung Electronics, Yongin, Rep. of Korea.

doi:10.4218/etrij.10.1510.0103

I. Introduction

As information technology advances, authentication systems are being increasingly employed in many areas in order to provide efficient and secure access control. However, this brings privacy concerns as well since personal information needs to be stored in authentication systems. Depending on the type of authentication, different approaches should be devised. Conceptually, authentication can be performed based on one of the following:

i) What you know (knowledge-based system),

ii) What you have (possession-based system),

iii) Who you are (biometrics-based system).

It is common to combine multiple methods to strengthen or supplement one another. For example, ATMs normally require a bank card and a PIN together for account access, which is a combination of the first two. These are relatively easy to implement compared to the third one, but there is a risk of losing a card or forgetting a PIN. This also means that impersonation is possible with acquired information and/or items. On the contrary, biometrics has no (or much less) such risk since it is intrinsically and permanently associated with a user, but there are other shortcomings to overcome.

The first shortcoming of biometric security is that biometric data is not fixed, but is slightly different each time it is scanned. Therefore, scanned data cannot be directly used as a key in conventional cryptographic systems, and it is not easy to protect such data using conventional cryptographic functions either. For example, since conventional encryption functions produce completely different outputs for similar inputs, comparing two encrypted templates produces an inconsistent matching score. Therefore, encrypted templates should be decoded, and the original templates are exposed for every

matching. For similar reasons, hashed templates cannot be used for user authentication.

The second problem for biometric security is that biometric data identifies a user as unique, but there are only a few substitutes. Therefore, once they lose their biometric data, they may permanently lose their identities. For example, a user has only one face and ten fingerprints, and it is impossible to change the biometric traits with new ones as if creating new passwords. Also, because biometric data is unique, it is much like the same password for multiple systems. Thus, if an attacker steals a user's biometric template from one system, he or she can login other systems using the stolen template.

The remainder of the paper is organized as follows. In section II we discuss background and related work. Section III summarizes the architecture and contribution of the proposed framework. Section IV is concerned with template randomization and section V with template hardening. In section VI, we discuss the one-time template (OTT). Section VII is a description of our experiment. Finally, we draw our conclusions in section VIII.

II. Background and Related Work

Biometric authentication requires enrollment and verification phases. In both phases, a user's biometric information, such as face, fingerprint, iris, or voice is scanned and processed to extract biometric features. If the scan is for enrollment, the features are registered as a template in a server's database for later verification. If it is for verification, the extracted features are compared with the registered features to make a decision to accept or reject.

Biometric templates should be replaced when they are compromised, just like changing a password or an ID card. This concept was introduced as private biometrics by Davida and others [1] and as cancelable biometrics by Ratha and others [2]. If the original forms of biometric features are used as templates, it may not be possible to cancel and replace them since biometric features are unique and intrinsically bound to a user. Accordingly, there has been intensive research on template protection, which can fall into a few categories:

i) Biometric hash (or robust hash) functions,

ii) Biometric sketches,

iii) One-way transformation that preserves biometric similarity metrics.

The biometric hash function is designed to generate the same hash output if the difference in inputs is small. Therefore, even though scanned data are different each time, it should produce a consistent hash output for a user. The biometric hash output must be difficult to invert, just like a cryptographic hash function. To make templates cancelable, pseudo-random data are combined with biometric data for hash input. Even if the same biometric features are used several times, different pseudo-random data produce independent hash output. Related work can be found in [3]-[6]. However, in this method, well-known feature matching algorithms may not be used since the matching is performed on hash output. Therefore, the performance will rely on biometric hash functions, and the performance of existing biometric recognition systems may not be guaranteed.

A biometric sketch is a method to extract a fixed key given a noisy biometric sample. The noise of a scanned sample is removed with the help of error correction codes, namely sketches. The derived key is used for user authentication or further cryptographic operation. If a template in a server is compromised, a new template can be generated with a different sketch. Some related work can be found in [7]-[11]. This kind of scheme has some privacy weaknesses, which are presented in [12]. An attacker can demonstrate linking and reversing protected templates when the same biometrics are deployed in multiple systems with different sketches.

The third method uses a transformation function that preserves the same biometric similarity metrics after the transformation as before the transformation. The biometric similarity metrics can be Euclidian distance, correlation, or any other between biometric samples depending on employed matching algorithms. The function is deemed as being one-way in that an inverse operation is infeasible if the transformation parameter is unknown. This is quite different from a cryptographic hash function, which requires no secret parameter. However, note that a cryptographic hash function does not keep the biometric similarity metrics. Related work can be found in [2], [13]-[15]. However, these have a shortcoming. The transformations are designed to lose some biometric information in order to keep the one-way property of the transformation. This may degrade biometric recognition performance. Our work can fall into this method while preserving biometric recognition performance. This makes it possible to construct OTT for open networks.

III. Overall Architecture and Contribution

Figure 1 shows user enrollment and verification processes based on the proposed protection scheme, which is the extension of our previous work [16]. The proposed method is a two-factor authentication scheme. A user should provide both of his or her biometric data and transform information for positive verification. At an enrollment stage, it randomizes an original template using a distance preserving transform and then hardens the template using a *soft vault*. The randomization allows a registered template cancelable, and the hardening is to

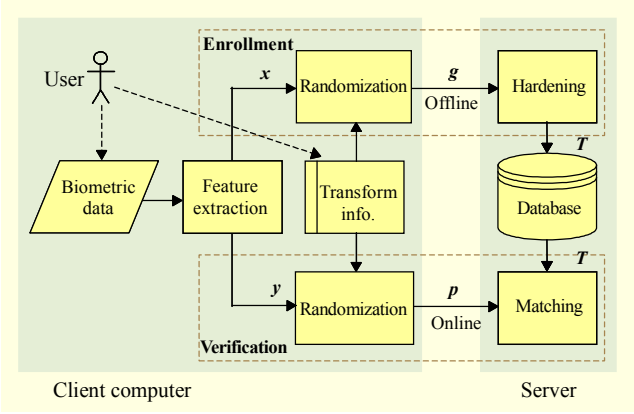


Fig. 1. User enrollment and verification in proposed framework.

prevent attackers from generating valid inputs and passing an authentication system even if the data in a server are disclosed. At the verification stage shown in Fig. 1, it randomizes an input template in the same way as it does at an enrollment stage. It compares the randomized template $\boldsymbol{p}$ with an enrolled hardened template $\boldsymbol{T}$. Note that $\boldsymbol{p}$ and $\boldsymbol{T}$ are not symmetric, and therefore a special procedure is required for template matching. Detailed explanations on template randomization and hardening will be given in sections IV and V, respectively. We designed the whole process such that biometric recognition performance does not deteriorate even if template matching is performed in an encoded state. It only assumes that a user's biometric template is in a form of real vectors, and template matching is performed using Euclidean distance, which are very minimal assumptions in most of the face authentication systems.

Finally, we expand the presented schemes for online communication by combining the concepts of OTT and hash-based randomized access control (H-RAC).

IV. Template Randomization

Even though biometrics has many desirable properties for ideal user authentication, it has some critical limitations: biometric templates are not revocable, and the distribution of biometric features is not uniform. To resolve such problems, we suggest a distance preserving transform. It generates a new template when an enrollment template is lost or compromised and conceals the range and distribution of biometric features so that an attacker cannot exploit the prior knowledge.

We assume a user's biometric feature or template is in a real vector form. We also assume matching is performed using Euclidean distance, as in eigenface [17] and fisherface [18]. We denote $\boldsymbol{x}$ as a *gallery* (enrolled template) and $\boldsymbol{y}$ as a *probe* (query template). They represent original templates, which are created directly from a user's biometric data.

At an enrollment stage, a user provides his or her biometric raw data, and an enrollment system creates an original gallery $\boldsymbol{x}$. Then, it generates a randomized template $\boldsymbol{g}$ as

$$\boldsymbol{g} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{A}$ is a random orthogonal matrix, and $\boldsymbol{b}$ is a random vector. An orthogonal matrix has the following properties:

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}=\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{I}, \tag{2}$$

where $\boldsymbol{A}^{\mathrm{T}}$ represents the transpose matrix of $\boldsymbol{A}$, and $\boldsymbol{I}$ is an identity matrix with the size of $\boldsymbol{A}$. The elements of $\boldsymbol{A}$ and $\boldsymbol{b}$ are generated using a pseudo-random number generator (PRNG), and the elements of $\boldsymbol{A}$ are orthonormalized using the Gram-Schmidt process. Instead of $\boldsymbol{x}$, $\boldsymbol{g}$ is used for a further process. The transformation data $\boldsymbol{A}$ and $\boldsymbol{b}$ are kept with the user on a personal token, such as a smartcard, or it can be directly generated from a PRNG using a user's password as a seed. Note that $\boldsymbol{A}$ and $\boldsymbol{b}$ are not stored with an enrollment template. If an enrolled template (or $\boldsymbol{A}$ and $\boldsymbol{b}$) is lost or compromised, we cancel it and create a new one using a new $\boldsymbol{A}$ and $\boldsymbol{b}$.

At the verification stage, a user provides his or her own biometric data along with $\boldsymbol{A}$ and $\boldsymbol{b}$. An authentication system creates an original probe $\boldsymbol{y}$ from the biometric data and then generates a transformed template $\boldsymbol{p}$ as

$$\boldsymbol{p} = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{b}. \tag{3}$$

Instead of $\boldsymbol{y}$, $\boldsymbol{p}$ is sent to a matching module over the network, as shown in Fig. 1. Matching can be performed directly using Euclidean distance on $\boldsymbol{g}$ and $\boldsymbol{p}$ if a user provides accurate $\boldsymbol{A}$ and $\boldsymbol{b}$ as in

$$\begin{aligned}\|\boldsymbol{g}-\boldsymbol{p}\|^2 &= (\boldsymbol{g}-\boldsymbol{p})^{\mathrm{T}}(\boldsymbol{g}-\boldsymbol{p})\\ &= (\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}-\boldsymbol{A}\boldsymbol{y}-\boldsymbol{b})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}-\boldsymbol{A}\boldsymbol{y}-\boldsymbol{b})\\ &= (\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{y})\\ &= (\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{y})\\ &= \|\boldsymbol{x}-\boldsymbol{y}\|^2.\end{aligned} \tag{4}$$

As shown in (4), matching itself does not change after transformation, and thus the decision between a genuine user and an impostor does not either. If a user provides a wrong $\boldsymbol{A}$ and $\boldsymbol{b}$, the distance between $\boldsymbol{g}$ and $\boldsymbol{p}$ is bigger than the distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ due to the mismatch of $\boldsymbol{A}$ and $\boldsymbol{b}$. This means that for positive verification, a user should provide correct transform data as well as his or her biometric data. Note that $\boldsymbol{A}$ and $\boldsymbol{b}$ are not directly involved in template matching, and the validity of $\boldsymbol{A}$ and $\boldsymbol{b}$ is checked implicitly through template matching.

The transform increases the randomness of biometric templates and allows us to create a new template. However, it is not preimage resistant: Given $\boldsymbol{g}$, it is easy to find an input template $\tilde{\boldsymbol{x}}$ and transformation information $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{b}}$ such that $f(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{A}}, \tilde{\boldsymbol{b}}) = \boldsymbol{g}$, where f denotes a distance preserving transform. It implies that once an attacker steals $\boldsymbol{g}$ from an

authentication system, he or she can recover its preimage $\tilde{\boldsymbol{x}}$, $\tilde{\boldsymbol{A}}$, and $\tilde{\boldsymbol{b}}$ without knowing $\boldsymbol{A}$ and $\boldsymbol{b}$ and log into the system by inputting the recovered preimage. To prevent such attacks, a transformed template is further encoded by a hardening technique. A related discussion on this kind of attack against cancelable biometrics can be found in [19].

V. Template Hardening

In this section, we propose a template hardening technique, named a soft vault. A soft vault further encodes a randomized template $\boldsymbol{g}$ so that an enrolled template cannot be used as an input to gain illegal access even if it is disclosed from an authentication system. Therefore, the finalized template stored on an authentication system is a hardened template. Before describing a soft vault, we derive a couple of decision equivalences. The decision equivalences provide a clue to designing the hardening scheme, and they are essential to prove that user authentication based on the protection scheme does not violate a conventional biometric decision rule.

1. Decision Equivalence

We start by reviewing genuine and impostor distributions, shown in Fig. 2. A genuine distribution characterizes a matching score between a gallery and a probe from the same person, and an impostor distribution characterizes a matching score between a gallery and a probe from different persons. When a dissimilarity measure, such as Euclidean distance, is used, a genuine distribution occurs below an impostor distribution. Considering the trade-off between a false acceptance rate (FAR) and false rejection rate (FRR), a threshold θ is determined from the distributions. Generally, the value of θ is chosen at the point where the two distributions intersect. At a verification stage, if a matching score is smaller than θ, a system accepts a user as genuine; otherwise, the user is rejected. The decision rule with θ can be restated in terms of the feature space as follows:

Statement 1. If a probe feature vector $\boldsymbol{p}$ falls inside a circle of radius θ centering at a gallery feature vector $\boldsymbol{g}$, then classify it as a genuine template.

Statement 2. If $\boldsymbol{p}$ falls outside a circle of radius θ centering at $\boldsymbol{g}$, then classify it as an impostor template.

If the condition of statement 1 is satisfied, the matching score (distance) between $\boldsymbol{p}$ and $\boldsymbol{g}$ is smaller than θ. This is the condition where a conventional biometric system accepts a user as genuine. Statement 2 is for the other case. Figure 3(a) shows examples in a two-dimensional feature space. In the figure, $\boldsymbol{p}_1$ lies inside a circle of radius θ centering at $\boldsymbol{g}$, so it is classified as a genuine template. For another example, $\boldsymbol{p}_2$ is classified as an impostor template since it lies outside the circle.

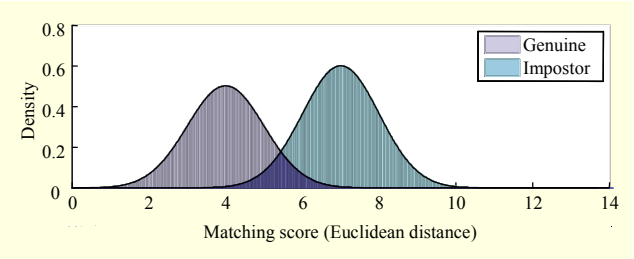


Fig. 2. Example of genuine/impostor distribution.

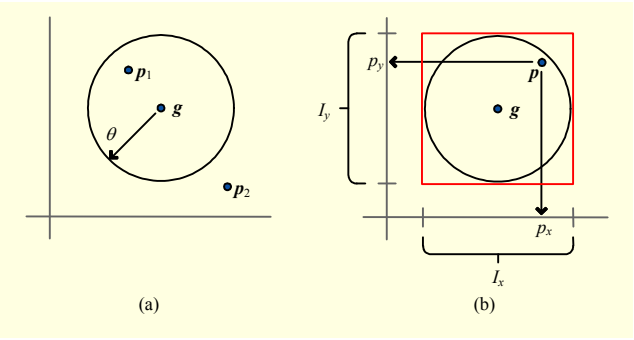


Fig. 3. (a) Threshold and decision boundary in a feature space and (b) circumscribed square around a decision boundary.

As shown in the figure, θ actually plays a role of a decision boundary in a feature space. We can easily verify that it is also true in a high-dimensional feature space.

We further revise statement 1 to derive another decision condition that is consistent with a conventional biometric decision rule. Its usefulness will be clear in the next subsection. As shown in Fig. 3(b), we draw a square circumscribed around a circle centering at $\boldsymbol{g}$. The square plays the role of a safeguard for the decision boundary. If the square is not invaded, the original decision boundary is not violated either. In the figure, we can easily see that the following statement is true, and its consequence is equivalent to statement 1. Therefore, a decision method based on the statement is equivalent to a conventional biometric decision rule.

Statement 3. If $\boldsymbol{p}$ falls inside a circumscribed square around a circle of radius θ centering at $\boldsymbol{g}$, and the distance between $\boldsymbol{p}$ and the center point of the square is less than θ, then $\boldsymbol{p}$ falls inside a circle of radius θ centering at $\boldsymbol{g}$.

Statement 3 checks the condition of statement 1 in two steps. First, it roughly tests whether a probe $\boldsymbol{p}$ belongs to an original decision boundary using a circumscribed square. Then, it examines whether $\boldsymbol{p}$ actually fits into the decision boundary. In Fig. 3(b), p_x and p_y represent the x-axis and y-axis components of $\boldsymbol{p}$, and I_x and I_y denote the x-axis and y-axis intervals of the square. Whether $\boldsymbol{p}$ lies inside the circumscribed square or not can be checked by examining whether p_x and p_y belong to I_x and I_y, respectively.

2. Hardening and Authentication

In this subsection, we describe our template hardening technique and user authentication method based on the previous discussion. To simplify the explanation, we define two terms, *real interval* and *chaff interval*. Figure 4 shows examples for real intervals and chaff intervals. In the figure, R and C denote the centers of real and chaff intervals, respectively. Real intervals correspond to the sides of the circumscribed square, that is, I_x and I_y. The centers of the real intervals are actually the same as gallery $\boldsymbol{g}$. Chaff intervals make it difficult for an attacker to pick out real intervals from mixed ones. Chaff intervals have the same length as real intervals, 2θ, and are set side by side from real intervals. According to the discussion on statement 3, chaff intervals should not be overlapped with real intervals in order to preserve an original decision boundary. Using these terms, a hardening (or enrollment) process consists of the following steps:

Step 1. Receive a gallery $\boldsymbol{g}$.
Step 2. Derive real intervals from $\boldsymbol{g}$.
Step 3. Set chaff intervals on the sides of the real intervals.
Step 4. Compute $H(\boldsymbol{g})$ and throw away $\boldsymbol{g}$.

In step 3, chaff intervals should be set on each side of real intervals such that the positions of real intervals are randomized. In step 4, $H(\boldsymbol{g})$ denotes the hashed value of $\boldsymbol{g}$. Note that it is infeasible to guess $\boldsymbol{g}$ from $H(\boldsymbol{g})$ since a hash function H is one-way and preimage resistant. A hardened template $\boldsymbol{T}$ consists of real intervals, chaff intervals, and the hashed value of $\boldsymbol{g}$. For an example of an n-dimensional feature vector and three chaff intervals on each dimension, a harden template can take the structure shown in Table 1. In the table, $H(R_1, R_2,\ldots, R_n)$ is equivalent to $H(\boldsymbol{g})$.

When m chaff intervals are set for an n-dimensional feature vector, the possible combinations are $(m+1)^n$, which grows exponentially with the dimension of a feature vector. Therefore, even if a hardened template is disclosed, it is computationally infeasible to find the exact combination for $\boldsymbol{g}$ when the dimension of a feature vector is high enough, for example, n=200 and m=1. In other words, the scheme is preimage resistant, and an attacker cannot gain access to a system using a stolen hardened template alone.

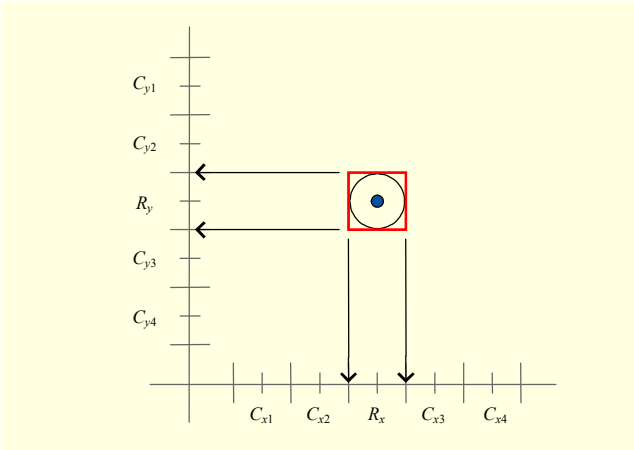


Fig. 4. Real and chaff intervals in two-dimensional feature space.

Table 1. Hardened template.

$H(R_1, R_2,\ldots, R_n)$
$C_{1,1}, R_1, C_{1,2}, C_{1,3}$
$C_{2,1}, C_{2,2}, R_2, C_{2,3}$
⋮
$R_n, C_{n,1}, C_{n,2}, C_{n,3}$

An authentication process with a hardened template consists of the following steps:

Step 1. Receive a probe $\boldsymbol{p}$ from a user.
Step 2. Load the hardened template $\boldsymbol{T}$ of the claimed identity by a user.
Step 3. Find the center $\tilde{\boldsymbol{g}}$ of the square that $\boldsymbol{p}$ falls inside using the intervals defined in $\boldsymbol{T}$.
Step 4. Compute $H(\tilde{\boldsymbol{g}})$ and read $H(\boldsymbol{g})$ from $\boldsymbol{T}$.
Step 5. Calculate the distance between $\boldsymbol{p}$ and $\tilde{\boldsymbol{g}}$, $D(\boldsymbol{p}, \tilde{\boldsymbol{g}})$.
Step 6. If $H(\tilde{\boldsymbol{g}}) = H(\boldsymbol{g})$ and $D(\boldsymbol{p}, \tilde{\boldsymbol{g}}) < \theta$, then accept a user.

In step 3, the square and its center point $\tilde{\boldsymbol{g}}$ can be identified by searching the intervals where the components of $\boldsymbol{p}$ belong. In step 6, the two hashed values are equal if and only if $\boldsymbol{p}$ falls within the circumscribed square whose center is $\boldsymbol{g}$. Therefore, step 6 actually checks the condition of statement 3, and our scheme accepts it if and only if a conventional biometric system does. Note that the decision between a genuine user and an impostor is made using $H(\boldsymbol{g})$ without revealing $\boldsymbol{g}$ explicitly. A server can recover $\boldsymbol{g}$ from $\boldsymbol{T}$ only when it receives valid $\boldsymbol{p}$ from a client, which is equivalent information to $\boldsymbol{g}$.

VI. One-Time Template

The work of the previous section is effective for concealing biometric templates. However, it is vulnerable to replay attacks. If an attacker eavesdrops and reuses a previous communication, he or she can easily obtain authentication from a server. For security against replay attacks, we use OTTs. A new template can be created in the sequel by changing $\boldsymbol{A}$ and $\boldsymbol{b}$. In this paper, we consider the case using Euclidean distance as a matching measure.

1. One-Time Template Generation and Update

Assume that just before the n-th authentication, a server has $\boldsymbol{T}_n$ and secret number K_n, and a user has $\boldsymbol{A}_n$, $\boldsymbol{b}_n$, and K_n. For easy explanation, suppose that a hardened template $\boldsymbol{T}_n$ is created from a randomized template $\boldsymbol{g}_n$. At the n-th authentication, a user provides his face image and a token storing $\boldsymbol{A}_n$ and $\boldsymbol{b}_n$. Then, a client creates the n-th probe $\boldsymbol{p}_n$ as shown in (5) and sends it to a server with the pseudo-ID $H(ID \| K_n)$:

$$\boldsymbol{p}_n = \boldsymbol{A}_n \boldsymbol{y} + \boldsymbol{b}_n. \tag{5}$$

A server searches template information with the pseudo-ID and performs template matching. If a matching result is positive, a server recovers $\boldsymbol{g}_n$ from $\boldsymbol{T}_n$. Note that a server does not store $\boldsymbol{g}_n$, but it temporally reconstructs $\boldsymbol{g}_n$. It can do that only when it obtains valid $\boldsymbol{p}_n$, which is equivalent to $\boldsymbol{g}_n$. Matching between $\boldsymbol{p}_n$ and the recovered $\boldsymbol{g}_n$ can be performed directly using Euclidean distance if a user provides accurate $\boldsymbol{A}_n$ and $\boldsymbol{b}_n$ as explained in the previous section. The detailed proof under the circumstance of template updating will be given in the next subsection.

After the n-th authentication is positively performed, a server creates the $(n + 1)$th temporary gallery $\boldsymbol{g}_{n+1}$ from $\boldsymbol{g}_n$ as shown in (6) and updates K_n into K_{n+1}.

$$\boldsymbol{g}_{n+1} = \boldsymbol{A}'_n \boldsymbol{g}_n + \boldsymbol{b}'_n, \tag{6}$$

where $\boldsymbol{A}'_n$ is a new random orthogonal matrix, and $\boldsymbol{b}'_n$ is a new random vector, which are generated from K_n by a PRNG. The initial template $\boldsymbol{g}_0$ for (6) is created according to (1), as described in the previous section, when a new user registers at an authentication server, and it should be created independently of initial K. Even if an authentication server kept $\boldsymbol{g}_n$ with K_n, an old gallery $\boldsymbol{g}_{n-1}$ and the original gallery $\boldsymbol{x}$ cannot be guessed from the information disclosed from a server. After updating $\boldsymbol{g}_n$ into $\boldsymbol{g}_{n+1}$, a server creates a new hardened template $\boldsymbol{T}_{n+1}$ from $\boldsymbol{g}_{n+1}$ and discards $\boldsymbol{T}_n$, $\boldsymbol{g}_n$, $\boldsymbol{g}_{n+1}$, $\boldsymbol{A}'_n$, and $\boldsymbol{b}'_n$.

After a client confirms the verification result from a server, it generates a new transform, $\boldsymbol{A}_{n+1}$ and $\boldsymbol{b}_{n+1}$, as shown in (7) and (8), and stores it on a user's personal token:

$$\boldsymbol{A}_{n+1} = \boldsymbol{A}'_n \boldsymbol{A}_n, \tag{7}$$

$$\boldsymbol{b}_{n+1} = \boldsymbol{A}'_n \boldsymbol{b}_n + \boldsymbol{b}'_n. \tag{8}$$

Using K_n kept with a user, a client creates $\boldsymbol{A}'_n$ and $\boldsymbol{b}'_n$ in the same way as, but independently of, a server. A client also updates K_n into K_{n+1} and stores it on a user's token.

Elements of $\boldsymbol{A}'_n$ and $\boldsymbol{b}'_n$ are generated based on K_n. For simple implementation, K_n can be used as a state number (or a seed number) for a PRNG. Since with the same state number, the same sets of random numbers are generated from a PRNG, a client and a server can create the same elements of $\boldsymbol{A}'_n$ and $\boldsymbol{b}'_n$, but independently. Then, they properly normalize the elements of $\boldsymbol{A}'_n$ and $\boldsymbol{b}'_n$.

Table 2. Protocol flow for OTTs.

1.	A client sends a probe $\boldsymbol{p}_n$ with a pseudo-ID $\hat{ID}_n = H(ID \| K_n)$.
2.	A server searches $\boldsymbol{T}_n$ in a database using the received pseudo-ID, and verifies $\boldsymbol{p}_n$. If a verification result is positive, a server sends R_n and $H(K_n \| R_n)$ to a client and updates $\boldsymbol{T}_n$ using K_n, K_n with $K_{n+1} = H(K_n \oplus R_n)$, and $\hat{ID}_n$ with $\hat{ID}_{n+1} = H(ID \| K_{n+1})$.
3.	A client updates $\boldsymbol{A}_n$ and $\boldsymbol{b}_n$ using K_n, and K_n with K_{n+1} in the same way after checking the validity of $H(K_n \| R_n)$.

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

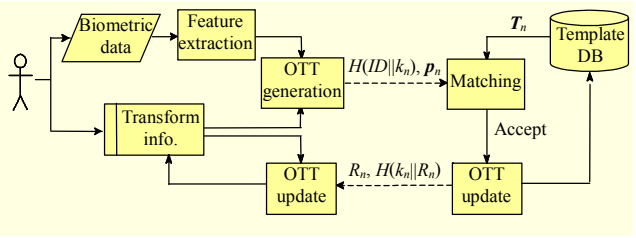


Fig. 5. Verification and template updating.

The protocol flow for OTTs is summarized in Table 2 and Fig. 5. We have modified semi-randomized access control (SRAC) [20] and applied it to OTTs. Since the values of $\hat{ID}_n$ and $\boldsymbol{p}_n$ change every time according to the protocol, replay attacks can be prevented. A random number R_n is used to update a secret number K_n. The reason that a server sends R_n with $H(K_n \| R_n)$ is to prevent a malicious change of R_n. A client can check the validity of R_n by comparing the value of $H(K_n \| R_n)$. We can also achieve forward secrecy by using a hash function. Since the chain of K_n is made by a hash function, guessing K_{n-1} by K_n is computationally infeasible.

By applying SRAC, we enhance the privacy of the proposed protocol. All the messages exchanged between a client and a server are used only once. Note that instead of a real ID, we use a pseudo-ID for the privacy protection of a user. Even if an attacker eavesdrops several instances of the protocol flow with the same user, he or she cannot figure out whether they are from the same user or not.

2. Matching Consistency

We can prove by mathematical induction that matching between randomized templates by Euclidean distance is performed consistently as a OTT is successively updated.

Let $\boldsymbol{g}_0$ be an initially enrolled template, and $\boldsymbol{p}_0$ be a first verification template after enrollment:

$$\boldsymbol{g}_0 = \boldsymbol{A}_0\boldsymbol{x} + \boldsymbol{b}_0, \tag{9}$$

$$\boldsymbol{p}_0 = \boldsymbol{A}_0\boldsymbol{y} + \boldsymbol{b}_0, \tag{10}$$

where $\boldsymbol{A}_0$ and $\boldsymbol{b}_0$ are transform data used at an enrollment stage. The following holds as proved previously:

$$\|\boldsymbol{g}_0 - \boldsymbol{p}_0\|^2 = \|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{11}$$

When $n = 1$, we have a gallery and a probe as follows:

$$\boldsymbol{g}_1 = \boldsymbol{A}_0'\boldsymbol{g}_0 + \boldsymbol{b}_0', \tag{12}$$

$$\boldsymbol{p}_1 = \boldsymbol{A}_1\boldsymbol{y} + \boldsymbol{b}_1, \tag{13}$$

where, $\boldsymbol{A}_1 = \boldsymbol{A}_0'\boldsymbol{A}_0$ and $\boldsymbol{b}_1 = \boldsymbol{A}_0'\boldsymbol{b}_0 + \boldsymbol{b}_0'$. With the updating rule, $\boldsymbol{p}_1$ can be written in terms of $\boldsymbol{p}_0$:

$$\begin{aligned}\boldsymbol{p}_1 &= \boldsymbol{A}_1\boldsymbol{y} + \boldsymbol{b}_1 \\ &= \boldsymbol{A}_0'\boldsymbol{A}_0\boldsymbol{y} + \boldsymbol{A}_0'\boldsymbol{b}_0 + \boldsymbol{b}_0' \\ &= \boldsymbol{A}_0'(\boldsymbol{A}_0\boldsymbol{y} + \boldsymbol{b}_0) + \boldsymbol{b}_0' \\ &= \boldsymbol{A}_0'\boldsymbol{p}_0 + \boldsymbol{b}_0'.\end{aligned} \tag{14}$$

Using the relation, the following can be derived:

$$\begin{aligned}\|\boldsymbol{g}_1 - \boldsymbol{p}_1\|^2 &= (\boldsymbol{g}_1 - \boldsymbol{p}_1)^{\mathrm{T}}(\boldsymbol{g}_1 - \boldsymbol{p}_1) \\ &= (\boldsymbol{g}_0 - \boldsymbol{p}_0)^{\mathrm{T}}\boldsymbol{A}_0'^{\mathrm{T}}\boldsymbol{A}_0'(\boldsymbol{g}_0 - \boldsymbol{p}_0) \\ &= \|\boldsymbol{g}_0 - \boldsymbol{p}_0\|^2 \\ &= \|\boldsymbol{x} - \boldsymbol{y}\|^2.\end{aligned} \tag{15}$$

At time n, a gallery and a probe are defined as follows:

$$\boldsymbol{g}_n = \boldsymbol{A}_{n-1}'\boldsymbol{g}_{n-1} + \boldsymbol{b}_{n-1}', \tag{16}$$

$$\boldsymbol{p}_n = \boldsymbol{A}_n\boldsymbol{y} + \boldsymbol{b}_n. \tag{17}$$

Assume that for arbitrary n, the following equation holds:

$$\|\boldsymbol{g}_n - \boldsymbol{p}_n\|^2 = \|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{18}$$

Based on the assumption, it can be proved that the following equation is also true:

$$\|\boldsymbol{g}_{n+1} - \boldsymbol{p}_{n+1}\|^2 = \|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{19}$$

According to the definition and updating rule, $\boldsymbol{g}_{n+1}$ and $\boldsymbol{p}_{n+1}$ can be written as follows:

$$\boldsymbol{g}_{n+1} = \boldsymbol{A}_n'\boldsymbol{g}_n + \boldsymbol{b}_n', \tag{20}$$

$$\begin{aligned}\boldsymbol{p}_{n+1} &= \boldsymbol{A}_{n+1}\boldsymbol{y} + \boldsymbol{b}_{n+1} \\ &= \boldsymbol{A}_n'\boldsymbol{A}_n\boldsymbol{y} + \boldsymbol{A}_n'\boldsymbol{b}_n + \boldsymbol{b}_n' \\ &= \boldsymbol{A}_n'(\boldsymbol{A}_n\boldsymbol{y} + \boldsymbol{b}_n) + \boldsymbol{b}_n' \\ &= \boldsymbol{A}_n'\boldsymbol{p}_n + \boldsymbol{b}_n'.\end{aligned} \tag{21}$$

By inserting the two equations into $\|\boldsymbol{g}_{n+1} - \boldsymbol{p}_{n+1}\|^2$, the following can be derived:

$$\begin{aligned}\|\boldsymbol{g}_{n+1} - \boldsymbol{p}_{n+1}\|^2 &= (\boldsymbol{g}_{n+1} - \boldsymbol{p}_{n+1})^{\mathrm{T}}(\boldsymbol{g}_{n+1} - \boldsymbol{p}_{n+1}) \\ &= (\boldsymbol{g}_n - \boldsymbol{p}_n)^{\mathrm{T}}\boldsymbol{A}_n'^{\mathrm{T}}\boldsymbol{A}_n'(\boldsymbol{g}_n - \boldsymbol{p}_n) \\ &= \|\boldsymbol{g}_n - \boldsymbol{p}_n\|^2 \\ &= \|\boldsymbol{x} - \boldsymbol{y}\|^2.\end{aligned} \tag{22}$$

It proves that matching is performed consistently from the beginning, and thus the updating rules are valid.

VII. Experiment

We perform face authentication experiments to confirm that our protection scheme does not degrade recognition performance. We use XM2VTS face data [21]. The data set contains 295 subjects with 8 frontal images for each subject. Four images per person are used for training (gallery), and the remaining four images are used for testing (probe). Face images are normalized and resized into 64×64 pixels, and face features are extracted using Fisher linear discriminant with principal component analysis [18]. The extracted features for each image are formed into a 49-dimension vector, and we consider it as an original template. One gallery per subject is created by calculating the mean of four feature vectors. Matching is performed based on Euclidean distance. Note that we are not trying to show the best performance of face recognition but to confirm empirically as well as theoretically that our scheme does not degrade the recognition performance of established biometric systems. Also, our scheme can be readily applied to any other data if they are in the form of a real vector.

1. Randomized Template

In the following experiments, randomized templates are used without applying a soft vault in order to show their own characteristics with the distance preserving transforms. Authentication experiments with a soft vault will be given in the next subsection. The other purpose of the following experiments is to confirm that both a private transform and a biometric template are required for positive verification. They also show that the distance preserving transformation can revoke a lost or compromised template by generating a new template. Once a new template is created and registered, a lost or stolen old template cannot be used anymore to gain access to a system. This also means that a user authentication scheme with OTTs can effectively prevent replay attacks against templates transmitted over networks.

Figure 6 shows histograms for matching scores for original and transformed templates. In the figures, *genuine* denotes

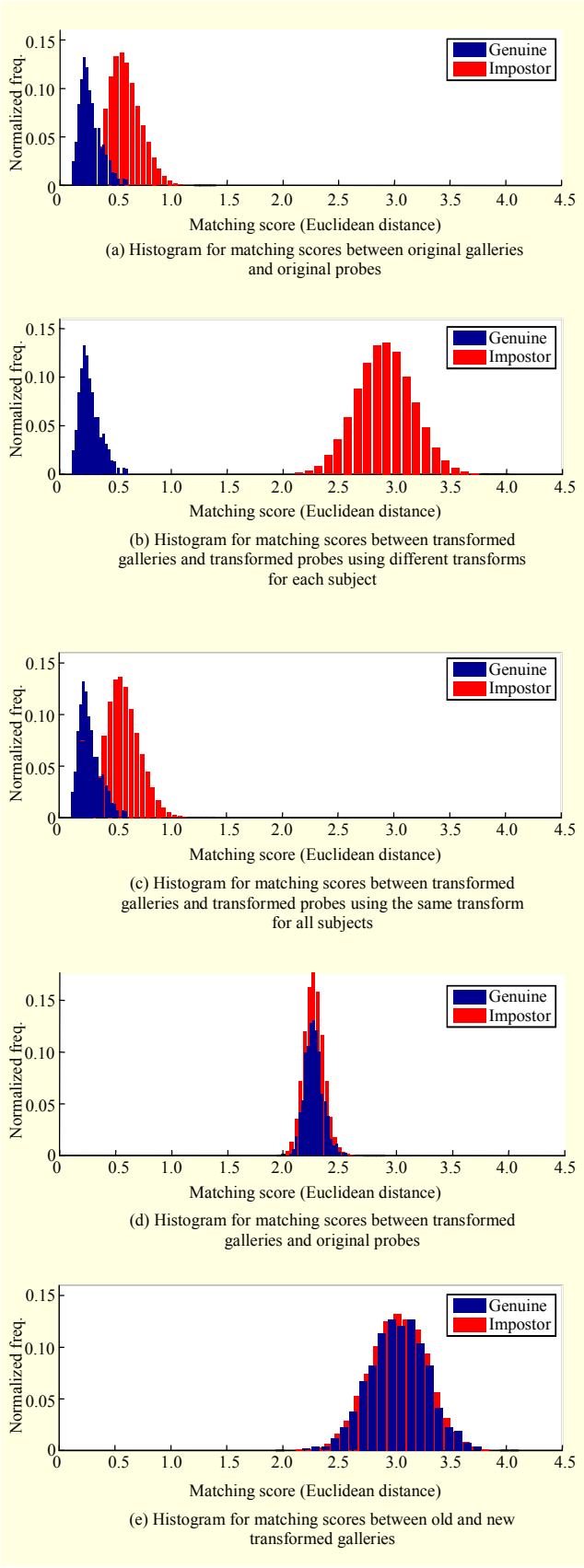


Fig. 6. Histograms for matching scores.

matching between templates from biometric features of the same subject, and *impostor* denotes matching between templates from biometric features of different subjects. The terms are used in association with the identities of templates only, not with a user's private transform.

Figure 6(a) is the histogram for matching scores between original galleries and original probes, which are not transformed. It is the baseline of the following experiments. The genuine and impostor histograms overlap each other, and the equal error rate is 9.75%.

Figure 6(b) is the histogram for matching scores between transformed galleries and transformed probes. A different transform is assigned to each subject, and for the same subject, a gallery and a probe are converted using the same transform. This experiment represents the ordinary operations where each user safely keeps his or her biometric and transformation data. As discussed in the previous section, there is no change in the genuine histogram, but rather the impostor histogram has moved away from the genuine histogram. They are completely separated from each other. However, one should not accept that perfect recognition is attainable by setting a threshold larger than the previous case where the protection scheme is not employed. For example, suppose that a system manager has changed a threshold from 0.5 to 1.0 in order to decrease FRR without increasing FAR. If an attacker steals a private transform, he or she can gain access with an arbitrary template, which a conventional system (the baseline experiment) does not accept as a genuine template. This is because when a private transform is stolen, an impostor histogram returns to its original form, as shown in Fig. 6(c), that is, it moves back closer to a genuine histogram. Although the new threshold does not allow any single impostor template under the separated genuine/impostor distribution, it is relatively lenient under the original genuine/impostor distribution. Therefore, a threshold must be determined based on the baseline experiment.

Figure 6(c) is the histogram for matching scores between transformed galleries and transformed probes using the same transform for all subjects. This experiment simulates the situation where an attacker obtains a user's transformation data but not biometric data. In the figure, the impostor matching represents an attacker trying to gain an illegal access to a system with stolen transformation data and his or her own biometric data. Note that the genuine and impostor histograms are the same as the histograms of the baseline. So, even if the attacker obtains a user's private transform, he or she still needs a valid biometric template, that is, one which a conventional biometric system accepts as a genuine template, in order to gain access to a system.

Figure 6(d) is the histogram for matching scores between transformed galleries and original probes. This experiment

assumes a situation where an attacker obtains a user's biometric data but not transformation data. In the figure, the genuine matching represents an attacker trying to gain illegal access to a system using stolen biometric data with a dummy transform, which consists of an identity matrix and a zero vector. For attacks with arbitrary transforms, refer to Fig. 6(e). The genuine histogram is separated from the genuine histogram of Fig. 6(a). This means that it is impossible to gain access to a system using biometric data alone without valid transformation data, that is, data which are associated with the corresponding galleries.

Figure 6(e) is the histogram for matching scores between old and new transformed galleries. They are converted from the same original template using different transforms. This experiment shows that a new template can cancel and replace a lost old template. It also demonstrates that the OTT scheme can effectively prevent replay attacks over networks. In the figure, the genuine matching represents that an attacker tries to gain illegal access to a system using an old transformed gallery. The genuine histogram is separated from the genuine histogram of Fig. 6(a). This means that once a new template is registered, an old template cannot be used anymore to gain access to a system.

2. Hardened Template

Figure 7 shows FAR/FRR diagrams for original and hardened templates. We apply a soft vault scheme to an original template (without distance preserving transformation) to make it clear that the soft vault scheme does not degrade biometric recognition performance. We insert three chaff intervals for each feature dimension. It can be easily guessed from the previous discussion that the recognition performance does not depend on the number of chaff intervals. The two diagrams are equivalent, and thus it empirically confirms as well that the proposed hardening technique does not violate the conventional biometric decision rule.

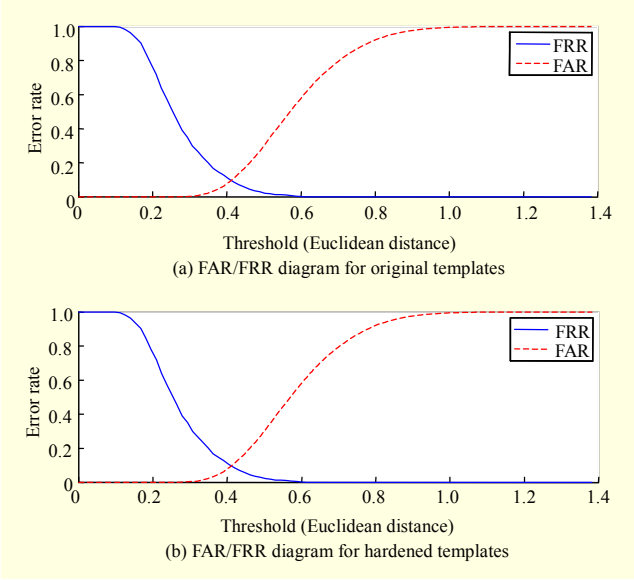


Fig. 7. FAR/FRR diagram.

VIII. Conclusion

Our method encodes a user's biometric feature into a secure form. This can be compared with BioHashing [22], [23], which is one of the most representative implementations of cancelable biometrics. BioHashing generates a BioCode from a user's biometric features by projecting them onto user-specific random vectors and then discritizes the projection coefficients into zero or one. Our scheme is similar to BioHashing in that it is a two-factor authentication scheme, meaning that both transform information and biometric data are required for positive verification, and consists of two processes, randomization and hardening (discritization for BioHashing). However, BioHashing has a couple of security flaws, which ours does not.

The first problem with BioHashing is that an attacker can fool an authentication system using a lost private transform alone [24]. Thus, it is difficult to say that BioHashing is truly a two-factor authentication scheme. In our method, however, an attacker still needs to provide a valid biometric template which an original biometric system accepts as a genuine template, even if he or she has a user's private transform. This is because a matching result does not change after randomizing and hardening templates.

The second problem with BioHashing is that it is not preimage resistant. This makes it easy to find an input from a BioCode, so an attacker can deceive an authentication system using a lost BioCode alone even without a user's private transform [19]. In our scheme, however, it is computationally impossible to recover an input from a hardened template, and thus an attacker cannot gain illegal access using a hardened template alone.

The proposed scheme can be considered as a concrete instance of cancelable biometrics. It can create a new template if an enrolled template is compromised and performs template matching in an encoded state without revealing an original template. In addition, the proposed scheme does preserve matching, and thus it can be directly applied to a conventional biometric system without any degradation of recognition performance.

References

[1] G. Davida, Y. Frankel, and B.J. Matt, "On Enabling Secure

Applications through Off-Line Biometric Identification," *IEEE Symp. Security Privacy*, 1998, pp. 148-157.

[2] N.K. Ratha, J.H. Connell, and R.M. Bolle, "Enhancing Security and Privacy in Biometrics-Based Authentication Systems," *IBM Syst. J.*, vol. 40, no. 3, Jan. 2001, pp. 614-634.

[3] T. Connie et al., "PalmHashing: A Novel Approach for Cancelable Biometrics," *Inf. Process. Lett.*, vol. 93, no. 1, Jan. 2005, pp. 1-5.

[4] T. Kevenaar et al., "Face Recognition with Renewable and Privacy Preserving Binary Templates," *IEEE Workshop Automatic Identification Advanced Technol.*, 2005, pp. 21-26.

[5] Y. Sutcu, T. Sencar, and N. Memon, "A Secure Biometric Authentication Scheme Based on Robust Hashing," ACM Workshop on Multimedia and Security, 2005, pp. 111-116.

[6] J.H. Ton and T. Kalker, "Robust Audio Hashing for Content Identification," Content-Based Multimedia Indexing, 2001.

[7] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," *Adv. Cryptology - EUROCRYPT*, LNCS 3027, 2004, pp. 523-540.

[8] A. Juels and M. Sudan, "A Fuzzy Vault Scheme," *IEEE Int. Symp. Inf. Theory*, 2002, p. 408.

[9] A. Juels and M. Wattenberg, "A Fuzzy Commitment Scheme," *ACM Conf. Computer Commun. Security*, 1999, pp. 28-36.

[10] Q. Li, Y. Sutcu, and N. Memon, "Secure Sketch for Biometric Templates," *Adv. Cryptology - ASIACRYPT*, LNCS 4284, 2006, pp. 99-113.

[11] J.P. Linnartz and P. Tuyls, "New Shielding Functions to Enhance Privacy and Prevent Misuse of Biometric Templates," *Audio- and Video-Based Biometric Person Authentication*, LNCS 2688, 2003, pp. 393-402.

[12] K. Simoens, P. Tuyls, and B. Preneel, "Privacy Weaknesses in Biometric Sketches," *IEEE Symp. Security Privacy*, 2009, pp. 188-203.

[13] R. Ang, R. Safavi-Naini, and L. McAven, "Cancelable Key-Based Fingerprint Templates," *Australasian Conf. Inf. Security Privacy*, LNCS 3574, 2005, pp. 242-252.

[14] N. Ratha et al., "Cancelable Biometrics: A Case Study in Fingerprints," *Int. Conf. Pattern Recog.*, vol. 4, 2006, pp. 370-373.

[15] N.K. Ratha et al., "Generating Cancelable Fingerprint Templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, 2007, pp. 561-572.

[16] Y.J. Lee et al., "One-Time Templates for Face Authentication," *Int. Conf. Convergence Inf. Technol.*, Nov. 2007, pp. 1818-1823.

[17] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, Jan. 1991, pp. 71-86.

[18] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 19, no. 7, Oct. 1997, pp. 711-720.

[19] Y.J. Lee, Y.S. Chung, and K.Y. Moon, "Inverse Operation and Preimage Attack on Biohashing," *IEEE Workshop Computational Intell. Biometrics: Theory, Algorithms, Appl.*, Mar. 2009, pp. 92-97.

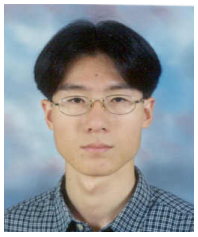
[20] Y.K. Lee and I. Verbauwhede, "Secure and Low-Cost RFID Authentication Protocols," *IEEE Int. Workshop Adaptive Wireless Networks*, Nov. 2005.

[21] K. Messer et al., "XM2VTSDB: The Extended M2VTS Database," *Audio- and Video-Based Biometric Person Authentication*, Mar. 1999, pp. 72-77.

[22] A.T.B. Jin, D.N.C. Ling, and A. Goh, "Biohashing: Two Factor Authentication Featuring Fingerprint Data and Tokenised Random Number," *Patt. Recog.*, vol. 37, no. 11, Nov. 2004, pp. 2245-2255.

[23] 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," *Int. Conf. Biometric Authentication*, LNCS 3072, 2004, pp. 117-123.

[24] A. Kong et al., "An Analysis of BioHashing and Its Variants," *Pattern Recog.*, vol. 39, no. 7, July 2006, pp. 1359-1368.



Yongjin Lee received his BS at the Department of Electronic, Computer, Electrical and Control Engineering from Hanyang University, Korea, in 2002, and his MS at the Department of Computer Science and Engineering from Pohang University of Science and Technology (POSTECH), Korea, in 2004. Since 2004, he has been a researcher at ETRI, Korea. His research interests are in the area of machine learning and pattern recognition.

Yongki Lee received his BS and MS in computer science and engineering from Hanyang University, Korea, in 1997 and 1999, respectively, and his MS and PhD in electrical engineering from the University of California Los Angeles (UCLA), USA, in 2006 and 2009, respectively. From 2007 to 2009, he was a visiting scholar in Computer Security and Industrial Cryptography (COSIC), K.U. Leuven, Belgium. Since 2010, he has been a senior engineer in the Digital IP Development Team, Samsung Electronics. His research interests are in the area of security protocol design/analysis and security hardware implementation especially for authentication protocol design for radio frequency identification and efficient security processor design.

Yunsu Chung received his MS and PhD in electronics from Kyungpook National University, Daegu, Korea, in 1995 and 1998, respectively. Since 1999, He has been with ETRI. He has researched in the field of image processing and computer vision. His major research interests include biometrics, video surveillance, human robot interface and human computer interface.

Kiyoung Moon received his BS and MS degrees in electronics engineering in 1986 and 1989, respectively, from Kyungpook National University, Korea. He received the PhD degree in computer science from Chungnam National University, Korea, in 2006. Since 1994, he has been a principal member of technical staff at ETRI, Korea. He is currently working as the project manager of the Biometric Technology Research Team. His research interests include biometrics, XML security, distributed system, and application security.