Fingerprint Template Protection Using One-Time Fuzzy Vault

Woo Yong Choi[1], Yongwha Chung[2], Jin-Won Park[3] and Dowon Hong[1]
[1] Software Research Laboratory, ETRI
161 Gajeong-dong, Yuseong-gu, Daejeon 305-700 - Korea
[e-mail: {wychoi4, dwhong}@etri.re.kr]
[2] Department of Computer and Information Science, Korea University
Jochiwon-eup, Yeongi-gun, Chungnam 339-700 - Korea
[e-mail: ychungy@korea.ac.kr]
[3] School of Games, Hongik University
Jochiwon-eup, Yeongi-gun, Chungnam 339-701 - Korea
[e-mail: jinon@hongik.ac.kr]
*Corresponding author: Yongwha Chung

Received July 8, 2011; revised October 7, 2011; accepted October 31, 2011; published November 29, 2011

Abstract

The fuzzy vault scheme has emerged as a promising solution to user privacy and fingerprint template security problems. Recently, however, the fuzzy vault scheme has been shown to be susceptible to a correlation attack. This paper proposes a novel scheme for one-time templates for fingerprint authentication based on the fuzzy vault scheme. As in one-time passwords, the suggested method changes templates after each completion of authentication, and thus the compromised templates cannot be reused. Furthermore, a huge number of chaff minutiae can be added by expanding the size of the fingerprint image. Therefore, the proposed method can protect a user's fingerprint minutiae against the correlation attack. In our experiments, the proposed approach can improve the security level of a typical approach against brute-force attack by the factor of 10^{34}.

Keywords: Fuzzy vault, one-time template, fingerprint authentication, cancelable biometrics, correlation attack

This work was supported by the IT R&D program of MKE/KEIT [10035157, Development of digital forensic technologies for real-time analysis] and the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. 2009-0086 148).

DOI: 10.3837/tiis.2011.11.020

1. Introduction

With a growing concern regarding security, interest in biometrics is increasing. Because biometrics utilizes a user's physiological or behavioral characteristics, which are unique and immutable, the compromise of biometric templates is a serious problem.

Ratha et al. have introduced cancelable biometrics as a remedy for the problem of compromised templates [1][2]. Cancelable biometrics distorts or transforms a user's template using some non-invertible functions to obscure the user's raw physical characteristics, and its matching is performed 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 or practical functions in their work, even though they have discussed randomization as a general way of producing cancelable templates.

The fuzzy vault scheme [3], which is based on binding a fingerprint template with a private key and scrambling it with a large amount of redundant data, has emerged as a promising solution to user privacy and fingerprint template security problems. Some researchers have implemented the fuzzy vault for fingerprints, and have protected the fingerprint minutiae by adding chaff points into the vault [4][5][6]. Generally, more chaff points guarantee higher security. However, the maximum number of chaff points for hiding the real minutiae is limited by the size of the fingerprint image. Therefore, the most previous researches of the fuzzy fingerprint vault reported their verification accuracy with less than or equal to 400 chaff points. For higher security, we need an approach which allows more number of chaff points. Furthermore, this scheme has been shown to be susceptible to a correlation attack that correlates two vaults created by the same finger in order to reveal the fingerprint minutiae hidden in the vaults [7][8].

In this paper, we propose a one-time fuzzy vault for fingerprint authentication. A new template can be generated if it is somehow compromised or lost, and matching is carried out in a transformed state without revealing the users' original minutiae. Also, by applying such transformation repeatedly, we can generate a different template for each authentication so that an old compromised template cannot be reused. The proposed algorithm not only transforms real minutiae but also expands the size of a fingerprint image, which enables hundreds of thousands of chaff minutiae to be added. Therefore, it can protect a user's fingerprint minutiae against a brute-force attack as well as a correlation or replay attack.

This paper is organized as follows. The backgrounds to the one-time fuzzy vault are described in Section 2. Section 3 explains the proposed one-time fuzzy vault algorithm, and the experimental results are explained in Section 4. Section 5 summarizes the conclusion.

2. Backgrounds

2.1 Security Model

In general, biometric systems have 4 modules – sensor, feature extraction/template creation, matching, and decision modules [9]. **Fig. 1** shows the possible attack points of a biometric system.

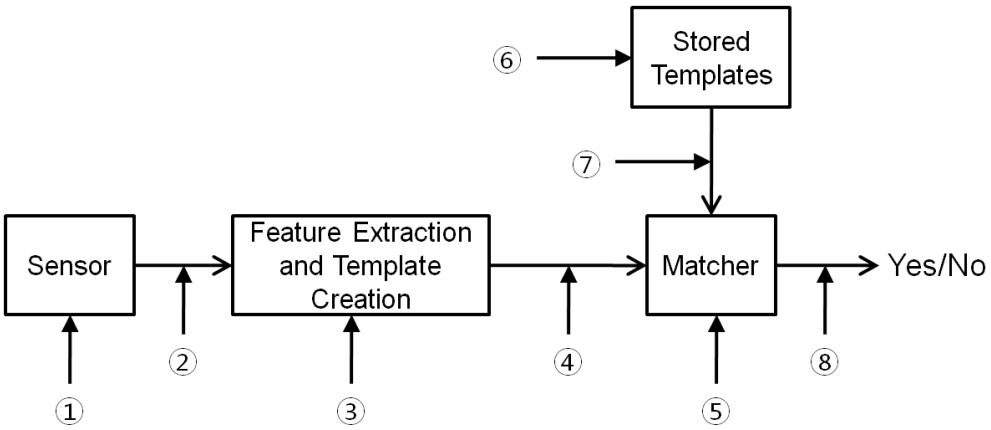

Fig. 1. Possible attack points of a biometric system [9]

Ratha et al. [9] identified 8 attack points in this scheme as described below.

① Attack at the sensor. (e.g., fake gelatin fingers)
② Attack on the channel between the sensor and the feature extractor (known as Replay Attack). This is not always possible since biometric sensor and feature extractor are sometimes combined.
③ Attack on the feature extraction and template creation module.
④ Attack on the channel between template creation and the matching modules (known as Replay Attack). The attacker can intercept a valid user's template for later use. There is large threat when templates are compared on a remote system.
⑤ Attack on the matcher. (e.g., change the threshold for a successful match)
⑥ Attack on the system database. The stolen template can be used to create a malicious template.
⑦ Attack on the channel between the system database and the matcher.
⑧ Attack on the final decision.

All of these attack points will make the biometric system vulnerable, but no method can defend all these attacks simultaneously. For example, anti-spoofing is used to defend the system against attack point 1, and challenge-response [2] is used to defend against attack point 2. In this paper, we focus on the attack points 4 and 6.

In many applications, fingerprint templates are stored in a central database (e.g., on-line banking system) or in a device (e.g., door lock system), so there exists a threat of so-called "big brother" problem. In addition, the fingerprint templates can also be compromised while they are transmitted from the client to the server over networks. The cancelable biometrics and fuzzy vault are well designed for eliminating the attack on the system database (attack point 6). However, they are still vulnerable to replay attack between template creation and the matching module (attack point 4). The compromise of a user's biometric templates causes serious problems. Since these templates have few substitutes, a user's identity can be permanently stolen.

2.2 Fingerprint Authentication

Fingerprint recognition is the most common biometric method for authentication. Since everyone's fingerprint is unique and invariable during life, fingerprint has been used as an evidence of forensic science and a personal authentication method.

A fingerprint authentication system has two phases: *enrollment* and *verification*. In the

offline *enrollment* phase, an enrolled fingerprint image is preprocessed, and the minutiae are extracted and stored. Image preprocessing refers to the refinement of the fingerprint image against the image distortion (poor contrast, flaw, smudge, etc.) obtained from a fingerprint sensor. Then, minutiae are extracted from the preprocessed image, which contain ridge endings, bifurcations, and singular points (cores and deltas). The singular points are very useful information to align two fingerprints. However, it is difficult to detect the singular points accurately due to the image distortion [10]. In addition, even the presence of cores and/or deltas is not guaranteed. Wang et al. [11] proposed the singular point detection method robust to noisy environment based on 2D Fourier expansion. On the other hand, Ahmad et al. [12] used local features to avoid using the inaccurate singular points.

In the on-line *verification* phase, the input fingerprint minutiae are compared with the enrolled fingerprint minutiae. Actually, matching is composed of the *alignment* stage and the *matching* stage. In order to match two fingerprints, the differences of the direction and the position between two fingerprints should be detected. Therefore, in the alignment stage, transformations such as translation and rotation between two fingerprints are estimated, and two minutiae are aligned according to the estimated parameters. If alignment is performed accurately, the matching stage is simply referred to point matching. In the matching stage, the enrolled and the input minutiae are compared based on their positions, orientations, and types, and finally, matching score is computed.

2.3 Cancelable Biometrics

To overcome the "big brother" problem, the concept of cancelable biometrics has been introduced by Ratha et al. [2]. It transforms biometric images or features into new ones by some non-invertible functions for an attacker not to restore the original one. Also, the matching between a probe and a gallery should be performed in a transformed domain.

Some fingerprint-related algorithms inspired from cancelable biometrics have been proposed [10], [12]-[16]. Ratha et al. [10] proposed the surface folding scheme for cancelable fingerprint templates. However, it was shown that the original template is recovered by a dictionary attack [17]. Ahmad et al. [12] proposed a new cancelable scheme which uses a relative position of a minutia in the polar coordinate space. They generated new feature vectors from the relations with neighboring minutiae, which result in cancelable template. Lee et al. [14] proposed a new method for making cancelable fingerprint templates using local minutiae transform. It provided a well-defined method for producing changing functions, which make fingerprint templates cancelable. Boult et al. [16] introduced the concept of revocable fingerprint-based biotoken. They used a reflective modulus operator as a distance measure to preserve the nearness property, and constructed intra- and inter-fingerprint minutia pair comparison tables. They divided the minutiae into two components: one of which is encrypted while the other is stored unsecured. The encrypted part is for security, and the unsecured part is for recognition accuracy.

2.4 Fuzzy Fingerprint Vault (FFV)

In 2002, Juels and Sudan [3] proposed the fuzzy vault, a new architecture with applications similar to Juels and Wattenberg's fuzzy commitment scheme [18], but a fuzzy vault is more compatible with partial and reordered data. In the fuzzy vault, Alice can place a secret value κ (e.g., a private encryption key) in a vault and lock (secure) it using an unordered set A. Bob, using an unordered set B, can unlock the vault (access κ) only if B overlaps with A to a great extent. The procedure for constructing the fuzzy vault is as follows: First, Alice selects a polynomial p in the variable u that encodes κ (e.g., by fixing the coefficients of p according to

κ). She computes the polynomial projections, $p(A)$, for the elements of A. She adds some randomly generated "chaff" points that do not lie on p, to arrive at the final point set R. When Bob tries to learn κ (i.e., finding p), he uses his own unordered set B. If B overlaps with A substantially, he will be able to locate many points in R that lie on p. Using error correcting codes, it is assumed that he can reconstruct p (and hence κ). The security of the scheme is based on the infeasibility of the polynomial reconstruction problem (i.e., if Bob does not know many points that lie on p, he cannot feasibly find the parameters of p, and hence he cannot access κ). Note that, since this fuzzy vault can work with unordered sets (common in biometric templates, including fingerprint minutiae data), it is a promising candidate for crypto-biometric systems.

Clancy et al. [4] and Uludag et al. [5] suggested the method for applying the fuzzy vault to fingerprint authentication, which is named as *fuzzy fingerprint vault*. It generates a lot of chaff minutiae and mixes them up with the real minutiae. Then, the real minutiae are projected on a randomly generated polynomial, and the chaff minutiae are projected off the polynomial. The polynomial is successfully reconstructed using either brute-force search or Reed-Solomon code if a sufficient number of real minutiae are chosen. The genuine user can choose a sufficient number of real minutiae by presenting his or her fingerprint while the impostors cannot. Some researchers have implemented the fuzzy vault for fingerprints, and have protected the fingerprint minutiae by adding chaff points into the vault [4][5][6][13][15][19][20]. However, this scheme has been shown to be susceptible to a correlation attack [8] that finds the real minutiae using multiple vaults enrolled for different applications. Nandakumar et al. [15] proposed a scheme for strengthening the security of fuzzy vault using password. It transformed the minutiae to prevent the compromised minutiae from being used by an attacker for wrong purposes.

2.5 One-Time Templates

Suppose that an attacker intercepted a template (or a cancelable template) while it was transmitted from the client to the server over networks. Then, the system is vulnerable to replay attack until the user detect the exposure and change the template. The cancelable biometrics and the fuzzy vault cannot protect the template against replay attack, even though they can fix the "big brother" problem. One-time template introduced by Ueshige et al. [21] is a good solution to this problem. They suggested the method of generating a series of new templates repeatedly as in one-time passwords, but did not provide any specific methods. This paper shows how one-time template is applied to fingerprint recognition systems.

3. Proposed Algorithm

Note that the alignment of two fingerprints is performed by overlapping two minutiae of different fingerprints, whose angles have the same direction. The main idea of the proposed algorithm is to transform the coordinates of the minutiae but not the angles in order to prevent the updated template from being aligned with the original template. In addition, by expanding the size of the fingerprint image, hundreds of thousands of chaff minutiae can be added. Thus, the proposed algorithm can bring considerable enhancement in security to the fuzzy fingerprint vault system.

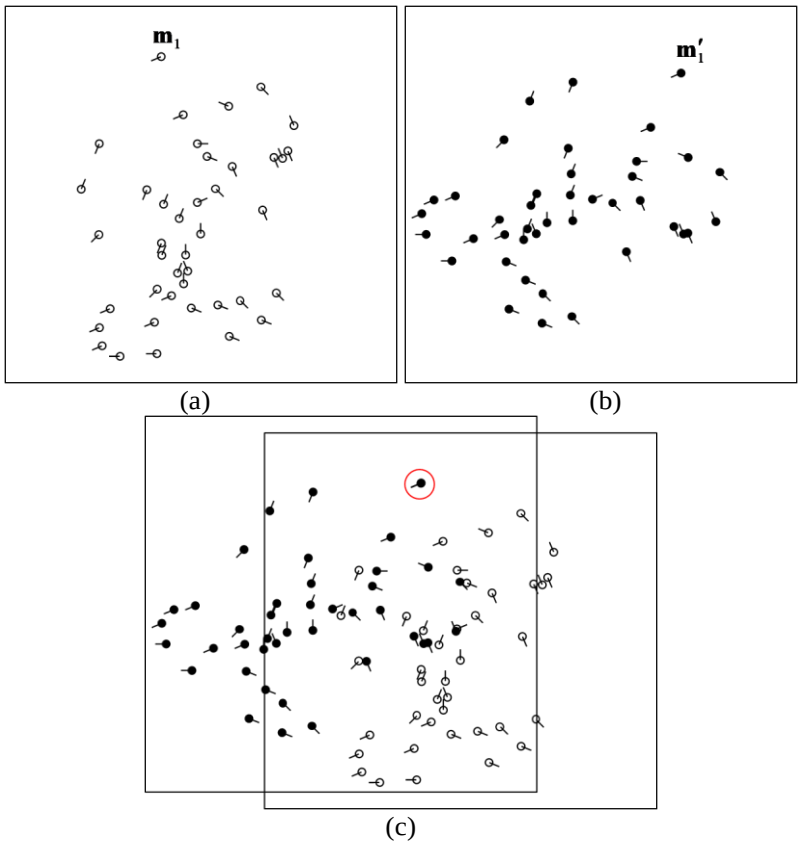

Fig. 2. Example of alignment results between original and transformed minutiae: (a) original minutiae, (b) transformed minutiae ($\alpha = 315°$), and (c) alignment results when $\mathbf{m}_1$ and $\mathbf{m}_1'$ are overlapped.

3.1 One-Time Transform

The proposed algorithm transforms the fingerprint minutiae using a rotation matrix and vector. It transforms the location of the minutiae, but keeps the angle of the minutiae unchanged.

To begin with, let $\mathbf{m}_i$ be the i-th minutia obtained from a fingerprint image, which is denoted by:

$$\mathbf{m}_i = (x_i, y_i, \theta_i, t_i), \quad i = 1, \cdots, M \tag{1}$$

$$\mathbf{w}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \tag{2}$$

where M is the number of the fingerprint minutiae, (x_i, y_i) are the x and y coordinates of the i-th minutia, and θ_i and t_i are the angle and type of the i-th minutia, respectively.

Let us consider a random rotation matrix $\mathbf{A}$ that rotates vectors in the xy-plane counterclockwise by a random angle of α, and a random vector $\mathbf{b}$ as follows:

$$\mathbf{A} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{3}$$

In fact, α is chosen uniformly from the range [30°, 330°] to guarantee minimum rotation. Then, the transformed minutia $\mathbf{m}_i'$ is created by (4) and (5):

$$\mathbf{m}_i' = (x_i', y_i', \theta_i, t_i), \quad i = 1, \cdots, M_1 \tag{4}$$

$$\mathbf{w}_i' = \begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \mathbf{A}\mathbf{w}_i \tag{5}$$

Since the angles of the minutiae are not changed, this transform has the effect of misaligning the transformed template with the original. **Fig. 2** shows how it works. Although $\mathbf{m}_1$ is transformed to $\mathbf{m}_1'$, the alignment would fail when $\mathbf{m}_1$ and $\mathbf{m}_1'$ are overlapped.

3.2 One-Time Fuzzy Vault (OTFV)

One-time fuzzy vault updates the fingerprint templates by using a one-time transform and by adding a lot of chaff minutiae. 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 original values. Also, by applying such transformation repeatedly, we can generate a different template for each authentication so that a compromised old template cannot be reused.

3.2.1 Enrollment Procedure

The enrollment procedure of OTFV is as follows.

① Given the fingerprint image to be enrolled, the minutiae are extracted from the image:

$$\mathbf{m}_i = (x_i, y_i, \theta_i, t_i), \quad i = 1, \cdots, M_E \tag{6}$$

$$\mathbf{w}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \tag{7}$$

where M_E is the number of enrollment minutiae.

② A random rotation matrix $\mathbf{A}_1$ and a random vector $\mathbf{b}_1$ are generated:

$$\mathbf{A}_1 = \begin{pmatrix} \cos\alpha_1 & -\sin\alpha_1 \\ \sin\alpha_1 & \cos\alpha_1 \end{pmatrix} \tag{8}$$

$$\mathbf{b}_1 = \begin{pmatrix} b_{11} \\ b_{12} \end{pmatrix} \tag{9}$$

This transformation information can be stored in a smartcard or a security token. In this paper, all arithmetic operations are conducted in a finite field of order 2^{26}, namely GF(2^{26}). Each coordinate is scaled to the range [0, 2^{13}] for the purpose of the arithmetic in GF(2^{26}). Thus, b_{ij} values are chosen uniformly in GF(2^{13}).

③ The enrollment minutiae are transformed by (8) and (9), and let $\mathbf{m}_i^{(1)}$ be the transformed minutiae:

$$\mathbf{m}_i^{(1)} = (x_i^{(1)}, y_i^{(1)}, \theta_i, t_i), \quad i = 1, \cdots, M_E \tag{10}$$

$$\mathbf{w}_i^{(1)} = \begin{pmatrix} x_i^{(1)} \\ y_i^{(1)} \end{pmatrix} = \mathbf{A}_1 \mathbf{w}_i + \mathbf{b}_1 \tag{11}$$

Once the original minutiae are transformed, they should be deleted for the sake of security.

④ A number of chaff minutiae are generated and a minutiae set is constituted along with the transformed enrollment minutiae. After adding the chaff minutiae, the total number of minutiae is M_R.

⑤ The (k+1) elements a_i ($0 \le i \le k$) are chosen uniformly at random in GF(2^{26}) and the polynomial $p(u)$ is defined as follows:

$$p(u) = a_0 + a_1 u + a_2 u^2 + \cdots + a_k u^k \tag{12}$$

⑥ The real and chaff minutiae are projected on and off the polynomial, respectively. That is:

$$v_i = \begin{cases} p(u_i) & \text{if } u_i \text{ is real} \\ p(u_i) + \delta_i & \text{if } u_i \text{ is chaff} \end{cases} \tag{13}$$

where δ_i is a non-zero element of GF(2^{26}).

⑦ The vault is constituted by the real and chaff points, and is stored in the database.

3.2.2 Verification Procedure

The verification procedure of OTFV consists of the following steps.

① Given the fingerprint image to be verified, the minutiae are extracted from the image:

$$\tilde{\mathbf{m}}_i = (\tilde{x}_i, \tilde{y}_i, \tilde{\theta}_i, \tilde{t}_i), \quad i = 1, \cdots, M_V \tag{14}$$

$$\tilde{\mathbf{w}}_i = \begin{pmatrix} \tilde{x}_i \\ \tilde{y}_i \end{pmatrix} \tag{15}$$

where M_V is the number of the verification minutiae.

② The verification minutiae are transformed using $\mathbf{A}_1$ and $\mathbf{b}_1$ stored in the smartcard or the security token:

$$\tilde{\mathbf{m}}_i' = (\tilde{x}_i', \tilde{y}_i', \tilde{\theta}_i, \tilde{t}_i), \quad i = 1, \cdots, M_V \tag{16}$$

$$\tilde{\mathbf{w}}_i' = \begin{pmatrix} \tilde{x}_i' \\ \tilde{y}_i' \end{pmatrix} = \mathbf{A}_1 \tilde{\mathbf{w}}_i + \mathbf{b}_1 \tag{17}$$

③ The transformed verification minutiae are compared with the vault stored in the database, and the matched minutiae are finally selected:

$$\mathbf{m}_i^{(1)} = (x_i^{(1)}, y_i^{(1)}, \theta_i, t_i), \quad i = 1, \cdots, M_M \tag{18}$$

where M_M is the number of the matched minutiae. The vault can be successfully unlocked only if the matched minutia set contains a sufficient number of the real minutiae, that is:

$$M_M - 2n_c \geq k + 1 \tag{19}$$

where n_c is the number of the chaff minutiae contained in the matched minutiae set.

④ The matched minutiae set may contain some chaff minutiae as well as the real minutiae even if the verification fingerprint is from a valid user. To remove the chaff points and reconstruct the polynomial, a Reed-Solomon decoder [22] is used. Reed-Solomon code is an error correcting code that can remove a chaff point at the expense of a real point. For example, if the matched point set contains 2 chaff points, at least 10 real points are required to reconstruct a 7-degree polynomial. The reconstructed polynomial is compared with the true polynomial stored in the vault. A decision of whether to accept or reject the user depends on the result of this comparison.

3.2.3 Update Procedure

After each completion of authentication, the enrolled template is to be updated by the one-time transform. This has the effect of making the compromised template or minutiae useless.

Let $\mathbf{m}_i^{(n)}$ be the enrollment minutiae that is transformed n times, and let $\mathbf{A}^{(n)}$ and $\mathbf{b}^{(n)}$ be the transforms to be stored in the smartcard in order to transform the user's verification minutiae. Note that $\mathbf{A}_n$ and $\mathbf{b}_n$ are a random rotation matrix and a random vector for the n-th transform, respectively.

Then, the xy-coordinate of the updated minutiae is as follows:

$$\begin{aligned}
\mathbf{w}_i^{(1)} &= \mathbf{A}_1\mathbf{w}_i + \mathbf{b}_1 \\
&= \mathbf{A}^{(1)}\mathbf{w}_i + \mathbf{b}^{(1)} \\
\mathbf{w}_i^{(2)} &= \mathbf{A}_2\mathbf{w}_i^{(1)} + \mathbf{b}_2 = \mathbf{A}_2(\mathbf{A}^{(1)}\mathbf{w}_i + \mathbf{b}^{(1)}) + \mathbf{b}_2 \\
&= \mathbf{A}_2\mathbf{A}^{(1)}\mathbf{w}_i + \mathbf{A}_2\mathbf{b}^{(1)} + \mathbf{b}_2 \\
&= \mathbf{A}^{(2)}\mathbf{w}_i + \mathbf{b}^{(2)} \\
&\vdots \\
\mathbf{w}_i^{(n+1)} &= \mathbf{A}_{n+1}\mathbf{A}^{(n)}\mathbf{w}_i + \mathbf{A}_{n+1}\mathbf{b}^{(n)} + \mathbf{b}_{n+1} \\
&= \mathbf{A}^{(n+1)}\mathbf{w}_i + \mathbf{b}^{(n+1)}
\end{aligned} \tag{20}$$

Since $\mathbf{A}_i$ is a rotation matrix, $\mathbf{A}^{(n)}$ is also a rotation matrix. Thus, we can obtain the following recursive formula:

$$\begin{aligned}
\mathbf{A}^{(n+1)} &= \mathbf{A}_{n+1}\mathbf{A}^{(n)} \\
\mathbf{b}^{(n+1)} &= \mathbf{A}_{n+1}\mathbf{b}^{(n)} + \mathbf{b}_{n+1}
\end{aligned} \tag{21}$$

After the n-th authentication, the enrollment minutiae are updated by $\mathbf{A}_{n+1}$ and $\mathbf{b}_{n+1}$, and $\mathbf{A}^{(n+1)}$ and $\mathbf{b}^{(n+1)}$ are stored in the user's smartcard. Next, steps ④-⑦ of the enrollment procedure are repeated. The verification procedure is simply replacing $\mathbf{A}_1$ and $\mathbf{b}_1$ with $\mathbf{A}^{(n)}$ and $\mathbf{b}^{(n)}$ in (17), respectively.

4. Experimental Results

For the purpose of evaluating the OTFV algorithm, we used DB1 of FVC2002 [23] fingerprint database (8 impressions for each of the 100 distinct fingers). Each sample is matched against the remaining samples of the same finger to compute the genuine acceptance rate (GAR). The first sample of each finger is matched against the first sample of the remaining fingers to compute the false acceptance rate (FAR). If matching g against h is performed, the symmetric one (i.e., h against g) is not executed to avoid correlation. The total numbers of genuine and impostor tests are $((8 \times 7) / 2) \times 100 = 2{,}800$ and $(100 \times 99) / 2 = 4{,}950$, respectively.

The recognition accuracy, the security level, and the execution time of the OTFV algorithm are compared with those of the FFV algorithm. For the OTFV algorithm, the method to determine the number of the chaff minutiae to be inserted in OTFV is shown in **Table 1**. Density is the number of minutiae per 1000 pixels. The size of OTFV is determined as 8000 × 8000 for the purpose of arithmetic in GF(2^{26}), and hence, the numbers of chaff points for OTFV are determined by multiplying the density by 64,000.

Table 1. The decision of the number of chaff minutiae to be inserted into OTFV

Size	388 × 374		
No. Minutiae	30		
No. FFV Chaff	200	300	400
Density	1.58	2.27	2.96
OTFV size	8000 × 8000		
No. OTFV Chaff	101,408	145,512	189,616

Table 2 shows the recognition accuracy and the security levels of FFV and OTFV. The security level is obtained by calculating the average number of evaluations for an attacker to crack the vault using brute force attack [5]. For example, for the case of 7-degree polynomial and 101,408 chaff points, an attacker needs at least 8 minutiae to reconstruct the correct polynomial. The average number of the enrollment minutiae (M_E) is 30; hence there are a total of $C(101438,8) \approx 3 \times 10^{35}$ combinations. Only $C(30,8) \approx 6 \times 10^{6}$ of these combinations will reconstruct the correct polynomial. Thus, it will need an average of 5×10^{28} evaluations for an attacker to crack the vault.

For the impostor tests, we randomly generated the transformation information because the impostors do not know the user's transformation information. The GAR of OTFV was almost the same as that of the conventional fuzzy vault, but FAR of OTFV was 0. If an impostor knows the user's transformation, the proposed OTFV becomes exactly the same as the conventional FFV, and the FAR will be almost the same as that of FFV. As the random vector **b** performs parallel translation, the fingerprint image becomes large, and hence, hundreds of thousands of chaff minutiae can be added. Therefore, the proposed approach could insert up to 190,000 chaff points without degrading the verification accuracy, and the security level is improved by the factor of 10^{34}.

Table 2. Recognition accuracy and security levels of the FFV and OTFV. The recognition accuracies are compared according to Failure To Enrollment Rate (FTER), Genuine Acceptance Rate (GAR), False Acceptance Rate (FAR), and Half Total Error Rate (HTER).

Polynomial degree	FTER (%)	FFV					OTFV (Proposed)				
		No. Chaff	GAR (%)	FAR (%)	HTER (%)	Security Level	No. Chaff	GAR (%)	FAR (%)	HTER (%)	Security Level
7	0.3	200	92.2	4.8	6.3	2×10^{7}	101,408	91.4	0.0	4.3	5×10^{28}
		300	91.3	3.1	5.9	4×10^{8}	145,512	90.2	0.0	4.9	9×10^{29}
		400	90.2	2.3	6.1	4×10^{9}	189,616	89.2	0.0	5.4	7×10^{30}
8	0.6	200	89.6	2.2	6.3	2×10^{8}	101,408	88.3	0.0	5.8	2×10^{32}
		300	89.0	1.5	6.2	6×10^{9}	145,512	87.6	0.0	6.2	6×10^{33}
		400	88.0	0.9	6.5	6×10^{10}	189,616	86.8	0.0	6.6	6×10^{34}
9	1.3	200	87.0	1.2	7.1	2×10^{9}	101,408	86.3	0.0	6.9	1×10^{36}
		300	86.6	0.5	7.0	9×10^{10}	145,512	85.3	0.0	7.4	4×10^{37}
		400	85.4	0.5	7.6	1×10^{12}	189,616	84.8	0.0	7.6	6×10^{38}
10	1.9	200	84.0	0.4	8.2	2×10^{10}	101,408	83.2	0.0	8.4	5×10^{39}
		300	83.5	0.2	8.3	1×10^{12}	145,512	82.7	0.0	8.6	3×10^{41}
		400	81.3	0.1	9.4	2×10^{13}	189,616	82.0	0.0	9.0	5×10^{42}
11	2.6	200	80.4	0.1	9.9	3×10^{11}	101,408	79.8	0.0	10.1	3×10^{43}
		300	79.2	0.1	10.5	2×10^{13}	145,512	79.5	0.0	10.3	2×10^{45}
		400	78.5	0.1	10.8	5×10^{14}	189,616	78.5	0.0	10.7	5×10^{46}
12	3.1	200	76.9	0.0	11.6	3×10^{12}	101,408	76.2	0.0	11.9	2×10^{47}
		300	75.9	0.0	12.1	4×10^{14}	145,512	75.5	0.0	12.2	2×10^{49}
		400	74.6	0.0	12.7	1×10^{16}	189,616	75.0	0.0	12.5	6×10^{50}

Table 3. The comparison of the execution times and template sizes of FFV and OTFV

Algorithm	No. Chaff Minutiae	Test	Time (sec)		Template Size (Kbyte)
			Enrollment	Verification	
FFV	200	Genuine	0.005	0.8	1.6
		Impostor	0.004	0.9	1.6
	300	Genuine	0.007	1.5	2.3
		Impostor	0.007	1.7	2.3
	400	Genuine	0.009	2.5	2.9
		Impostor	0.011	2.9	3.0
OTFV (Proposed)	1,000	Genuine	0.011	1.0	7.0
		Impostor	0.012	1.1	7.1
	10,000	Genuine	0.130	0.9	68.6
		Impostor	0.135	1.0	68.6
	100,000	Genuine	1.443	0.9	683.8
		Impostor	1.446	0.9	683.8

To investigate the changes in execution times and template sizes over the number of chaff minutiae, we inserted 1K, 10K, and 100K chaff minutiae, and the experimental results are shown in **Table 3**. Since the enrollment times and the template sizes are almost the same according to the degree of polynomial, we report the results of 9-degree polynomial as a representative. The sizes of OTFV were determined to maintain the densities as the 200 chaff minutiae case. All of the experiments were performed on a PC with Pentium IV 3.2 GHz CPU and 4GB RAM. As the more chaff minutiae are inserted, the more time is required to enroll the fingerprint, which is performed in off-line. The enrollment time is directly proportional to the

number of chaff minutiae while the on-line verification times are almost the same. Also, the update procedure of OTFV can be run in the background after the user has been authorized. On the other hand, the template sizes are also directly proportional to the number of chaff minutiae. As 7 bytes should be allocated to a minutia, about 700 Kbytes are required for 100,000 chaff minutiae. Although the template size is quite large, the size is acceptable with the current storage cost level. Note that, large template is stored in a server, whereas small transformation information is stored in a smartcard. Further, the number of chaff points is controllable. For less secure applications, we can set the number of chaff points as 1,000 or 10,000 in order to save the storage cost. As we mentioned, the previous researches could not insert a large number of chaff points, not because of the storage cost but because of the low verification accuracy.

5. Conclusions

In this paper, we proposed the OTFV algorithm, which shows significant enhancement of the security of FFV. The proposed algorithm changes the templates after each completion of authentication, as in one-time passwords, and thus the old templates cannot be reused. Therefore, the proposed OTFV algorithm can be a fundamental solution to the security vulnerability of FFV caused by correlation attack.

The security level of FFV is determined by the number of chaff minutiae. However, a typical approach can insert a limited number of chaff points (*i.e.*, 200~400) because of the poor verification accuracy problem. On the other hand, OTFV not only transforms the real minutiae but also expands the size of a fingerprint image, which enables hundreds of thousands of chaff minutiae to be added without performance degradation. In the experiments, we added chaff minutiae to the OTFV templates 500 times more than to the FFV templates, and hence, the security level is increased by 10^{34} times. Although the off-line enrollment of OTFV takes more time and it requires more hard disk space to save the fingerprint templates, the on-line verification time of the OTFV is almost the same as that of FFV, and the template size is also acceptable with the current cost of storage.

References

[1] R. Bolle, J. Connell, N. Ratha, "Biometrics Perils and Patches," *Pattern Recognition*, vol. 35, no. 12, pp. 2727-2738, Dec. 2002. Article (CrossRef Link)

[2] N. Ratha, J. Connell, R. Bolle, "Enhancing Security and Privacy in Biometrics-based Authentication Systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614-634, 2001. Article (CrossRef Link)

[3] A. Juels, M. Sudan, "A Fuzzy Vault Scheme," in *Proc. of IEEE Int. Symp. on Information Theory*, pp. 408, 2002. Article (CrossRef Link).

[4] T. Clancy, N. Kiyavash, D. Lin, "Secure Smartcard-based Fingerprint Authentication," in *Proc. of ACM SIGMM Workshop on Biometrics Methods and Applications*, pp. 45-52, 2003. Article (CrossRef Link)

[5] U. Uludag, S. Pankanti, A. Jain, "Fuzzy Vault for Fingerprints," in *Proc. of 5th Int. Conf. on Audio- and Video-Based Biometric Person Authentication (LNCS 3546)*, pp. 310-319, July 2005. Article (CrossRef Link)

[6] Y. Chung, D. Moon, S. Lee, S. Jung, T. Kim, D. Ahn, "Automatic Alignment of Fingerprint Features for Fuzzy Fingerprint Vault," in *Proc. of 1st SKLOIS Conf. on Information Security and Cryptology (LNCS 3822)*, pp. 358-369, Mar. 2006. Article (CrossRef Link)

[7] W. Scheirer, T. Boult, "Cracking Fuzzy Vaults and Biometric Encryption," in *Proc. of*

Biometrics Symposium, pp. 1-6, Sep. 2007. Article (CrossRef Link)
[8] A. Kholmatov, B. Yanikoglu, "Realization of Correlation Attack against the Fuzzy Vault Scheme," in *Proc. of SPIE Symp. on Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, vol. 6819, pp. 1-7, 2008. Article (CrossRef Link)
[9] N. Ratha, J. Connell, R. Bolle, "An Analysis of Minutiae Matching Strength," in *Proc. of 3rd Int. Conf. on Audio- and Video-Based Biometric Person Authentication*, pp. 223-228, June 2001. Article (CrossRef Link)
[10] N. Ratha, S. Chikkerur, J. Connell, R. Bolle, "Generating Cancelable Fingerprint Templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 561-572, Apr. 2007. Article (CrossRef Link)
[11] Y. Wang, J. Hu, D. Philip, "A Fingerprint Orientation Model Based on 2D Fourier Expansion (FOMFE) and Its Application to Singular-Point Detection and Fingerprint Indexing," *Special Issue on Biometrics: Progress and Directions, IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 573-585, Apr. 2007. Article (CrossRef Link)
[12] T. Ahmad, J. Hu, S. Wang, "Pair-Polar Coordinate-based Cancelable Fingerprint Templates," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2555-2564, Oct.-Nov. 2011. Article (CrossRef Link)
[13] J. Hu, "Mobile Fingerprint Template Protection: Progress and Open Issues," in *Proc. of 3rd IEEE Conf. on Industrial Electronics and Applications*, pp. 2133-2138, June 2008. Article (CrossRef Link)
[14] C. Lee, J. Choi, K. Toh, S. Lee, J. Kim, "Alignment-Free Cancelable Fingerprint Templates Based on Local Minutiae Information," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 4, pp. 980-992, Aug. 2007. Article (CrossRef Link)
[15] K. Nandakumar, A. Nagar, A. Jain, "Hardening Fingerprint-based Fuzzy Vault Using Password," in *Proc. of 2nd Int. Conf. on Biometrics (ICB)*, pp. 927-937, Aug. 2007. Article (CrossRef Link)
[16] T. Boult, W. Scheirer, R. Woodworth, "Revocable Fingerprint Biotokens: Accuracy and Security Analysis," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1-8, June 2007. Article (CrossRef Link)
[17] S. Shin, M. Lee, D. Moon, K. Moon, "Dictionary Attack on Functional Transform-based Cancelable Fingerprint Templates," *ETRI Journal*, vol. 31, no. 5, pp. 628-630, Oct. 2009. Article (CrossRef Link)
[18] A. Juels, M. Wattenberg, "A Fuzzy Commitment Scheme," in *Proc. of 6th ACM Conf. Computer and Communications Security*, G. Tsudik, ed., pp.28-36, 1999. Article (CrossRef Link)
[19] Y. Dodis, R. Ostrovsky, L. Reyzin, A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," in *Proc. of Eurocrypt (LNCS 3027)*, pp. 523-540, 2004. Article (CrossRef Link)
[20] A. Kanak, I. Sogukpinar, "Fingerprint Hardening with Randomly Selected Chaff minutiae," in *Proc. of 12th Int. Conf. on Computer Analysis of Images and Patterns*, pp. 383-390, 2007. Article (CrossRef Link)
[21] Y. Ueshige, K. Sakurai, "A Proposal of One-time Biometric Authentication," in *Proc. of Int. Conf. on Security and Management*, pp. 78-83, June 2006. Article (CrossRef Link)
[22] S. Gao, "A New Algorithm for Decoding Reed-Solomon Codes," *Communications, Information and Network Security*, Kluwer Academic Publishers, pp. 55-68, 2003. Article (CrossRef Link)
[23] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, A. Jain, "FVC2002: Second Fingerprint Verification Competition," in *Proc. of Int. Conf. on Pattern Recognition*, vol. 16, pp. 811-814, 2002. Article (CrossRef Link)

Woo Yong Choi received the BS and MS degrees from Pusan National University, Korea, in 1998 and 2000. He received the PhD degree from the Korea University, Korea in 2011. He worked for L&H Korea from 2000 to 2001 as a researcher. Currently, he is a senior researcher in the Software Research Laboratory, ETRI. His research interests include biometrics, security, and digital forensics.

Yongwha Chung received the BS and MS degrees from Hanyang University, Korea, in 1984 and 1986. He received the PhD degree from the University of Southern California, USA in 1997. He worked for ETRI from 1986 to 2003 as a Team Leader. Currently, he is a Professor in the Department of Computer Information, Korea University. His research interests include biometrics, security, and performance optimization.

Jin-Won Park received the BS degree from Seoul National University, Korea and the MS and the PhD degrees from The Ohio State University, U.S.A. in 1975, 1982 and 1987 respectively. He had worked for KDI from 1977 to 1980 as a researcher, and ETRI from 1988 to 1999 as a Team Leader. Currently, he is a Professor in the School of Games, Hongik University in Korea. His research interests include Computer Simulation, engineering education and performance optimization.

Dowon Hong received his B.S., M.S. and Ph.D. degrees in mathematics from Korea University, Seoul, Korea on 1994, 1996, and 2000. He is currently a principal member of engineering staff and the team leader of Cryptography Research team at the Electronics and Telecommunication Research Institute, Korea where his research interests are broadly in the area of applied cryptography, networks security, and digital forensics.