5

Protection of the Fingerprint Minutiae

Woo Yong Choi[1], Yongwha Chung[2] and Jin-Won Park[3]
[1]*Electronics and Telecommunications Research Institute (ETRI),*
[2]*Korea University,*
[3]*Hongik University*
Republic of Korea

1. Introduction

With a growing concern regarding security, interest in biometrics is increasing. Since biometrics utilizes a user's physiological or behavioral characteristic, which is unique and immutable, the compromise of biometric templates is a serious problem. Fingerprint authentication system is one of the most widely used biometric authentication systems. In general, in the enrollment procedure, the features are extracted from the enrollment image and are stored as a template. The template is compared to the features extracted from the verification image. Unlike passwords, however, biometrics has no or little substitutions. For example, if one's fingerprint template is compromised, he or she cannot use that fingerprint for any other fingerprint authentication system from then on.

Ratha et al. have introduced cancelable biometrics as a remedy for the problem of compromised templates (Bolle et al., 2002; Ratha et al., 2001). 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. Ratha et al. proposed the surface folding scheme for cancelable fingerprint templates (Ratha et al., 2007). They proposed a one-way transformation which moves minutia positions using two-dimensional Gaussian functions defined over the feature domain. However, if an attacker obtains two transformed templates and transformation parameters, the original template is recovered by a dictionary attack (Shin et al., 2009).

Fuzzy vault is a crypto-biometric algorithm proposed by Juels et al. (Juels & Sudan, 2002). It gives a promising solution to personal privacy and fingerprint template security problems. Clancy et al. and Uludag et al. suggested the method for applying the fuzzy vault to fingerprint authentication, which is named as *fuzzy fingerprint vault* (Clancy et al, 2003; Uludag et al., 2005). 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 (Chung et al., 2006; Clancy et al., 2003; Dodis et al.,

2004; Kanak & Sogukpinar, 2007; Nandakumar et al., 2007; Uludag et al., 2005). Lagrange interpolation (Hildebrand, 1987) is the most widely used polynomial interpolation method. However, it requires a little much time especially when the degree of polynomial is large. Brute-force search is employed for polynomial reconstruction attempts until the true polynomial is reconstructed, and Lagrange interpolation is used to interpolate the polynomial. Therefore, even if the real minutiae are chosen more than the degree of the polynomial, the brute-force search cannot reconstruct the polynomial in real-time when several chaff minutiae are chosen along with the real minutiae. All the previous results adopted the brute-force search to reconstruct the polynomial or skipped the procedure for polynomial reconstruction because of its difficulty (Li et al., 2006). In this work we propose a fast algorithm for polynomial reconstruction. To reduce the execution time, it determines the candidate sets with chaff points by using the Gaussian elimination and excludes them from the reconstruction trial. Since the Gaussian elimination is a time-consuming process, we have found a recursive formula to perform the Gaussian elimination effectively by using the Consistency Theorem (Anton, 1994). We confirmed that the proposed algorithm can be performed in real time even at the worst case.

There are a few attack methods on the fuzzy vault. Scheirer et al. suggested the methods of attacks against fuzzy vault including the attack via record multiplicity, which is known as the *correlation attack* (Scheirer & Boult, 2007). The correlation attack gives very powerful attack method when two fuzzy vaults obtained from the same fingerprint are available. On the other hand, when only one fuzzy vault is compromised and no additional information is available, brute-force attack can be used. Brute-force attack is employed for polynomial reconstruction attempts using the random combination of points until the true polynomial is reconstructed. In this work we propose a new attack algorithm which applies a fast polynomial reconstruction algorithm based on the Consistency Theorem. Also, we evaluate the proposed attack method, and compare it with the known attack methods such as the brute-force attack and the correlation attack.

This chapter is organized as follows. The conventional fingerprint authentication methods are described in Section 2. Section 3 presents the fuzzy fingerprint vault system and the proposed polynomial reconstruction algorithm followed by experimental results. In Section 4, various attack methods for fuzzy fingerprint vault (brute-force attack, correlation attack, and the fast polynomial reconstruction attack) and the experimental results are explained. Section 5 summarizes the conclusion.

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 the evidence of forensic science and the personal authentication method. Modern fingerprint recognition began in 1684, when Nehemiah Grew studied and described ridges, furrows and pores on hand and foot surfaces. In the late 1960s, the Live-Scan system which records fingerprints electronically was developed, which was a turning point of fingerprint recognition. Fingerprint was centralized into database, and the automatic fingerprint recognition system has been developed consistently.

A fingerprint authentication system consists of fingerprint sensor, pre-processing, feature extraction, storage, and matching as shown in Fig. 1. The fingerprint pattern is captured by a fingerprint sensor. A fingerprint sensor takes a snapshot of a fingerprint and saves it into an

image file. From the image, unique features of each fingerprint are extracted and saved in the storage. The storage may be either a central database or a smartcard. Before feature extraction, pre-processing is performed to extract the reliable features from the image. For fingerprint matching, features of an input fingerprint are compared to the features of the enrolled fingerprint data. By comparing similarity between two fingerprint feature sets, it is decided whether the two fingerprints are from the same person or not.

The fingerprint recognition algorithms can be classified into two categories: image-based and minutiae-based. Image-based methods are based on optical correlation and transform based features. The image-based methods lack the ability to track with variations in position, scale, and orientation angle, and hence, they cannot give reliable recognition accuracy. Nevertheless, the image-based methods have been studied continuously because of the following properties (Nanni & Lumini, 2009; Yang & Park, 2008). First, the image-based methods can be combined with the minutiae-based methods to improve the accuracies of the fingerprint authentication systems. Second, fingerprint features can be represented by a fixed length vector, which is suitable for various learning systems.

Minutiae-based methods are more popular matching techniques, which are included in almost all contemporary fingerprint identification and verification systems. They are based on the minutiae, such as ending, bifurcation, and singular points in the fingerprint, which have been known to be effective for fingerprint recognition. The minutiae form a pattern of points, and hence several well-known point pattern matching algorithms have been proposed in the late 80's.

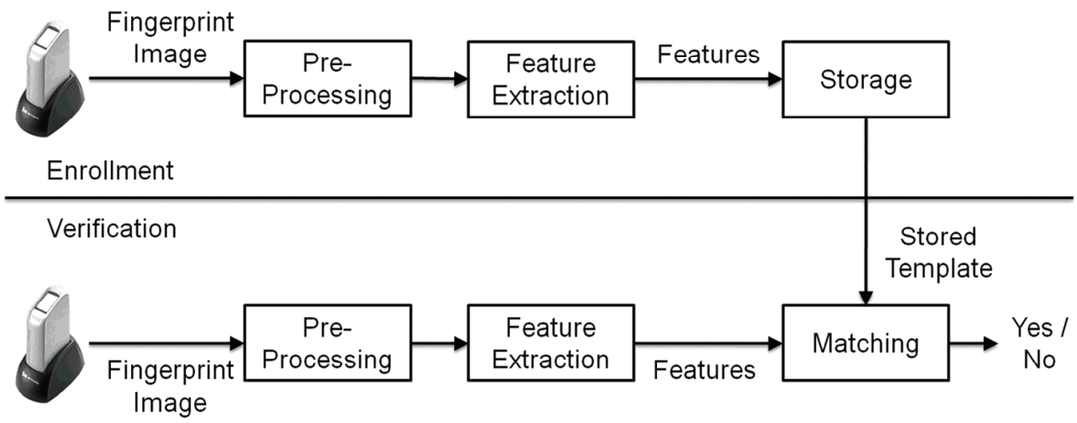


Fig. 1. Block diagram of the fingerprint authentication system (Pan et al., 2003)

A fingerprint authentication system has two phases: *enrollment* and *verification*. In the off-line enrollment phase, an enrolled fingerprint image is preprocessed, and the minutiae are extracted and stored. In the on-line *verification* phase, the similarity between the enrolled minutiae and the input minutiae is examined.

Image preprocessing refers to the refinement of the fingerprint image against the image distortion (poor contrast, flaw, smudge, etc.) obtained from a fingerprint sensor. Minutiae Extraction refers to the extraction of features from the fingerprint image. After this step, some of the minutiae are detected and stored into a pattern file, which includes the position, the orientation, and the type (ridge ending or bifurcation) of the minutiae.

The input fingerprint minutiae are compared with the enrolled fingerprint minutiae. Actually, Minutiae Matching is composed of the *alignment* stage and the *matching* stage. In

order to match two fingerprints captured with unknown direction and position, the differences of the direction and the position between two fingerprints should be detected, and alignment between them needs to be executed. 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 referred to point matching simply. In the matching stage, two minutiae are compared based on their position, orientation, and type. Then, a matching score is computed.

3. Fuzzy fingerprint vault

3.1 Enrollment procedure

Fig. 2 shows the block diagram of the enrollment procedure of the Fuzzy Fingerprint Vault (FFV) system. Given the fingerprint image to be enrolled, we first extract minutiae from the image to form a locking set of the form.

$$L = \{\mathbf{m}_i \mid 1 \le i \le n_e\} \tag{1}$$

where $\mathbf{m}_i = (x_i, y_i, \theta_i, t_i)$ is the i-th enrollment minutia, and n_e is the number of the enrollment minutiae. Then, a number of chaff minutiae are generated and constitute a minutia set along with the real minutiae. After adding the chaff minutiae, the total number of minutiae is n_r. All arithmetic operations are conducted in a finite field of order 2^{20}, namely GF(2^{20}). Thus, each coordinate is scaled to the range [0, 1024] for the purpose of the arithmetic in GF(2^{20}). A finite field (Stallings, 2005) is a field with a finite number of elements, also called a Galois field. All operations performed in the finite field result in an element within that field. For polynomial representation, we define the finite field over the irreducible polynomial $x^{20} + x^3 + 1$. Then, we randomly select (k + 1) elements from GF(2^{20}) and generate a k-degree polynomial as follows.

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

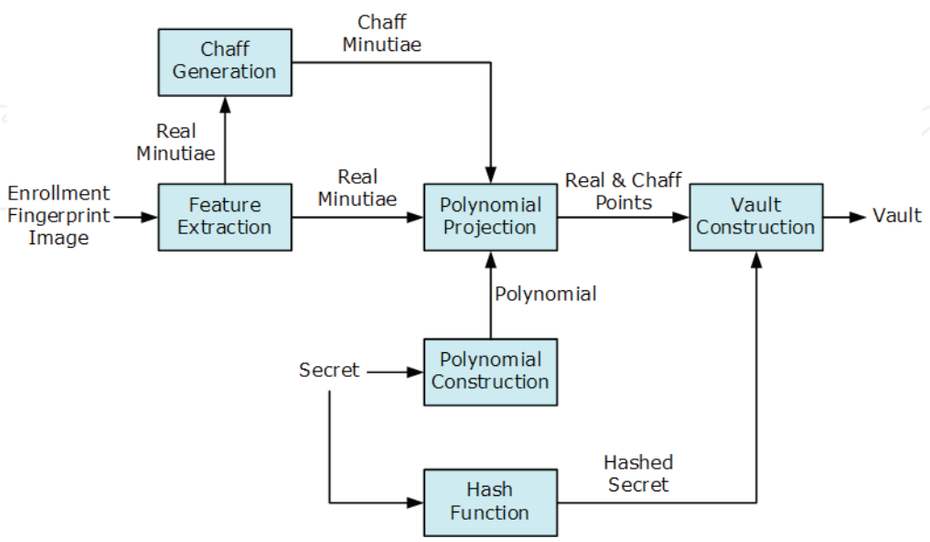


Fig. 2. Block diagram of the enrollment procedure of the FFV system (Choi et al., 2009)

This polynomial becomes the secret to be protected. As in the work of Uludag et al., we concatenate x and y coordinates of a minutia to arrive at the locking/unlocking data unit u. Then, we project the real and the chaff points (i.e., minutiae) on and off the polynomial, respectively. That is,

$$v_i = \begin{cases} p(u_i) & \text{if } u_i \text{ is real} \\ p(u_i) + \delta_i & \text{if } u_i \text{ is chaff} \end{cases} \tag{3}$$

where δ_i is a non-zero element of GF(2^{20}). Finally, the vault is constituted by the real and the chaff points, and the secret. The secret should be stored in a hashed form, instead of in the clear form.

3.2 Verification procedure

Fig. 3 shows the block diagram of the verification procedure of the FFV system. Given the fingerprint image to be verified, the minutiae are first extracted from the image and the verification minutiae set V is denoted by

$$V = \{\tilde{\mathbf{m}}_i \mid 1 \le i \le n_v\} \tag{4}$$

where $\tilde{\mathbf{m}}_i = (\tilde{x}_i, \tilde{y}_i, \tilde{\theta}_i, \tilde{t}_i)$ is the i-th verification minutia, and n_v is the number of the verification minutiae. Then, the verification minutiae are compared with the enrolled minutiae with real and chaff minutiae mixed, and an unlocking set U is finally selected.

$$U = \{\mathbf{m}_i \mid 1 \le i \le n_m\} \tag{5}$$

where n_m is the number of the matched minutiae. The vault can be successfully unlocked only if U overlaps with L to a great extent.

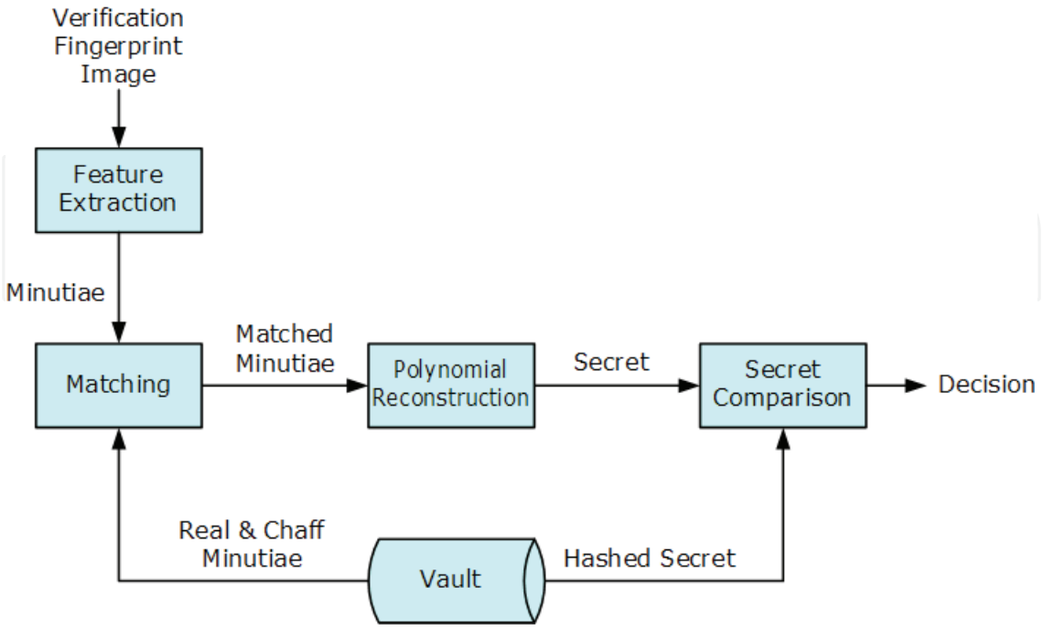


Fig. 3. Block diagram of the verification procedure of the FFV system (Choi et al., 2009)

These n_m points may contain some chaff points as well as the real points even if the user is genuine. Hence, in order to interpolate the k-degree polynomial, we have to select $(k + 1)$ real points from among the n_m points. After the polynomial is interpolated, it is compared with the true polynomial stored in the vault. A decision to accept/reject the user depends on the result of this comparison. If $|U \cap L| \geq (k + 1)$, the k-degree polynomial can be successfully reconstructed by using the brute-force search. The most widely used algorithm for polynomial interpolation is the Lagrange interpolation. The number of cases that select $(k + 1)$ minutiae from n_m minutiae is $C(n_m, k + 1)$. Let n_{real} be the number of real minutiae in set U, then the number of cases that correctly reconstruct the polynomial is $C(n_{real}, k + 1)$. Therefore, the average number of polynomial interpolation is

$$\frac{C(n_m, k+1)}{C(n_{real}, k+1)} \tag{6}$$

Furthermore, when a higher degree of polynomial is used, the Lagrange interpolation needs much more time to reconstruct the polynomial. More precisely, it can be done in $O(k \log^2(k))$ operations (Gathen & Gerhardt, 2003). Hence, it becomes impracticable as n_m and/or k increases. So, Uludag used only 18 minutiae to prevent n_m from being too large (Uludag et al., 2005).

Juels et al. suggested that the chaff points can be removed by means of the Reed-Solomon decoding algorithm (Juels & Sudan, 2002). Among various realizations of the Reed-Solomon code, we used the Gao's algorithm (Gao, 2003), which is one of the fastest Reed-Solomon decoding algorithms. It compensates one chaff point at the cost of one real point, so if there are many chaff points are matched along with real points, it is quite probable that the right user is rejected (i.e., false negative). The situation becomes much more serious when the degree of polynomial increases.

3.3 Polynomial reconstruction

If the matched point set of equation (5) contains more than $(k + 1)$ real point, the true polynomial can be reconstructed by using the brute-force search. Brute-force search chooses $(k + 1)$ points from among n_m points and tries to reconstruct the k-degree polynomial using the Lagrange interpolation, which requires a relatively large number of computations. So, when the matched point set contains many chaff minutiae, the number of the Lagrange interpolation to be performed increases exponentially, and hence, the polynomial reconstruction cannot be performed in real time.

In this section, we introduce a fast algorithm for the polynomial reconstruction (Choi et al., 2008). To begin with, let us consider the following theorem which provides the conditions under which a linear system of m equations in n unknowns is guaranteed to be consistent.

> **Consistency Theorem.** If $\mathbf{Ax} = \mathbf{b}$ is a linear system of m equations with n unknowns, then the followings are equivalent.
> (a) $\mathbf{Ax} = \mathbf{b}$ is consistent.
> (b) $\mathbf{b}$ is in the column space of $\mathbf{A}$.
> (c) The coefficient matrix $\mathbf{A}$ and the augmented matrix $[\mathbf{A} \mid \mathbf{b}]$ have the same rank.

Let us consider a linear system of $(k + 2)$ equations with $(k + 1)$ unknowns.

$$\mathbf{U} \cdot \mathbf{a} = \mathbf{v} \tag{7}$$

where

$$\mathbf{U} = \begin{bmatrix} 1 & u_1 & u_1^2 & \cdots & u_1^k \\ 1 & u_2 & u_2^2 & \cdots & u_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u_{k+2} & u_{k+2}^2 & \cdots & u_{k+2}^k \end{bmatrix}, \ \mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix}, \ \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{k+2} \end{bmatrix} \tag{8}$$

Then, the corresponding augmented matrix **W** is of the form.

$$\mathbf{W} = \left[\begin{array}{ccccc|c} 1 & u_1 & u_1^2 & \cdots & u_1^k & v_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & u_{k+2} & u_{k+2}^2 & \cdots & u_{k+2}^k & v_{k+2} \end{array}\right] \tag{9}$$

It is straightforward that the rank of matrix **U** is (k + 1). According to the Consistency Theorem, the rank of the augmented matrix **W** must be equal to (k + 1) to guarantee that the linear system has a solution. The Gaussian elimination was used to check whether the augmented matrix had rank (k + 1) or not. The elementary row operations, provided we do not perform the operation of interchanging two rows, were used to reduce the augmented matrix **W** into the row-echelon form.

$$\left[\begin{array}{ccccc|c} 1 & u_1 & u_1^2 & \cdots & u_1^k & v_1 \\ 0 & 1 & u_2^{2(2)} & \cdots & u_2^{k(2)} & v_2^{(2)} \\ 0 & 0 & 1 & \cdots & u_3^{k(3)} & v_3^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & v_{k+1}^{(k+1)} \\ 0 & 0 & 0 & \cdots & 0 & v_{k+2}^{(k+2)} \end{array}\right] \tag{10}$$

where $u_j^{l(i)}$ and $v_j^{(i)}$ are the values of u_j^l and v_j when the j-th row has "leading 1" at the i-th element, respectively. Note that the diagonal elements of equation (10) cannot be zero because the rank of matrix **U** is (k + 1). From the parts (a) and (c) of the Consistency Theorem, it follows that if $v_{k+2}^{(k+2)} \neq 0$, the linear system of equation (7) does not have a solution. Hence, there exists at least one chaff point in the set $\{(u_i, v_i) \mid 1 \leq i \leq k + 2\}$, and we need not perform the polynomial reconstruction process. On the contrary, if $v_{k+2}^{(k+2)} = 0$, then all the points are probably the real points. Thus, we try to reconstruct the polynomial with (k + 1) points and compare it with the true polynomial.

Up to this point, we have explained how to reconstruct a k-degree polynomial from (k + 2) matched minutiae. In general, the unlocking set has n_m minutiae, so let us consider a linear system of n_m equations with (k + 1) unknowns as follows.

$$\begin{bmatrix} 1 & u_1 & u_1^2 & \cdots & u_1^k \\ 1 & u_2 & u_2^2 & \cdots & u_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u_{n_m} & u_{n_m}^2 & \cdots & u_{n_m}^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n_m} \end{bmatrix} \tag{11}$$

where $n_m > (k + 1)$. Clearly, if $n_m < (k + 1)$, then the polynomial cannot be reconstructed. Also, if $n_m = (k + 1)$, we can reconstruct the polynomial with the $(k + 1)$ points. Suppose that we select $(k + 2)$ real points from among n_m points, we can reconstruct the true polynomial. However, if at least one chaff point exists in the $(k + 2)$ selected points the true polynomial cannot be reconstructed. The procedure for the proposed polynomial reconstruction algorithm is as follows.

1. $(k + 1)$ points are selected from among n_m points with real and chaff points mixed, and these points are placed to the top of equation (11).
2. The augmented matrix of equation (11) is obtained, and is reduced into the following row-echelon form.

$$\left[\begin{array}{ccccc|c} 1 & u_1 & u_1^2 & \cdots & u_1^k & v_1 \\ 0 & 1 & u_2^{2(2)} & \cdots & u_2^{k(2)} & v_2^{(2)} \\ 0 & 0 & 1 & \cdots & u_3^{k(3)} & v_3^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & v_{k+1}^{(k+1)} \\ 0 & 0 & 0 & \cdots & 0 & v_{k+2}^{(k+2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & v_{n_m}^{(k+2)} \end{array}\right] \tag{12}$$

3. To determine whether the $(k + 1)$ points $(u_1, v_1),\cdots,(u_{k+1}, v_{k+1})$ are valid candidates or not, we will check the values of $v_{k+2}^{(k+2)},\cdots,v_{n_m}^{(k+2)}$. Therefore, the proposed algorithm needs one more real point than the brute-force search to reconstruct the polynomial. If there is at least one zero among $v_{k+2}^{(k+2)},\cdots,v_{n_m}^{(k+2)}$, we reconstruct the polynomial with the selected $(k + 1)$ points, and compare it with the true polynomial.
4. Steps (1) ~ (3) are repeated until the true polynomial is reconstructed.

However, the computations of the Gaussian elimination in step (2) take too much time to be implemented in real time. Fortunately, in order to obtain the values of $v_{k+2}^{(k+2)},\cdots,v_{n_m}^{(k+2)}$, we do not have to apply the Gaussian elimination. We have found the following recursive formula, so the computation time can be considerably reduced.

$$v_j^{(i+1)} = \begin{cases} v_j, & i = 0 \\ \dfrac{v_j^{(i)} - v_i^{(i)}}{u_j - u_i}, & i = 1,\cdots,\min(k+1, j-1) \end{cases} \tag{13}$$

This gives exactly the same solution as the Gaussian elimination. The proof can be found on the reference (Choi et al., 2008).

3.4 Experimental results

To evaluate the performance of the proposed polynomial reconstruction algorithm, we used DB1 and DB2 of FVC2002 (Maio et al., 2002). The fingerprint images were obtained in three distinct sessions with at least two week time separating for each session. During the first session, the fingerprint images were obtained by placing the fingerprints with a normal position. During the second session, the fingerprint images were obtained by requesting the individuals to provide their fingerprints with exaggerated displacement and rotation (not to

exceed 35 degrees). During the third session, fingers were alternatively dried and moistened. The characteristics of the databases are listed in Table 1. The databases were obtained from the optical sensors. The size of fingerprint image of DB2 is greater than that of DB1. The resolutions of the databases are about 500 dpi. Each database consists of 100 fingers, and 8 impressions per finger. Each sample was matched against the remaining samples of the same finger to compute the Genuine Acceptance Rate (GAR). Similarly, the first sample of each finger was matched against the first sample of the remaining fingers to compute the False Acceptance Rate (FAR). If the matching g against h is performed, the symmetric one (i.e., h against g) is not executed to avoid the correlation. For each database, the number of genuine tests was 2800, whereas the number of impostor tests was 4950. All experiments were performed on a system with a 3.2 GHz processor.

	DB1	DB2
Sensor Type	Optical	Optical
Image Size	388 × 374(142 Kpixels)	296 × 560(162 Kpixels)
Resolution	500 dpi	569 dpi
Sensor	"TouchView II" by Identix	"FX2000" by Biometrika
No. Fingers	100	100
No. Impressions	8	8

Table 1. Characteristics of FVC2002 Databases

Database	No. Minutiae (n_e)		
	Average	Min	Max
DB1	30.5	5	60
DB2	39.0	7	89

Table 2. The number of minutiae for FVC2002 Databases

Database	No. Chaff Minutiae	Test	No. Matched Minutiae (n_m)			Matching Time (sec)
			Total	Real	Chaff	
DB1	200	Genuine	19.7	18.8	0.9	0.76
		Impostor	9.3	3.3	5.9	0.87
	300	Genuine	20.0	18.6	1.4	1.48
		Impostor	10.9	2.9	8.0	1.73
	400	Genuine	20.4	18.4	2.0	2.45
		Impostor	12.3	2.5	9.8	2.89
DB2	200	Genuine	24.3	23.4	0.9	1.26
		Impostor	10.6	4.3	6.3	1.64
	300	Genuine	24.5	23.2	1.3	2.38
		Impostor	12.4	3.9	8.5	3.17
	400	Genuine	24.9	23.1	1.8	3.87
		Impostor	13.9	3.4	10.5	5.26

Table 3. Average number of matched minutiae and matching time for FVC2002 databases

To examine the effect of the insertion of chaff minutiae on the performance of a fingerprint recognition system, we selected the number of chaff minutiae as 200, 300 and 400. For each database the numbers of minutiae (average, minimum and maximum) are listed in Table 2. Since the images of DB2 are obtained from bigger sensor, more minutiae are extracted from DB2 than from DB1. Also, the average numbers of the matched minutiae according to the number of inserted chaff minutiae are listed in Table 3. The more chaff minutiae are added, the more minutiae are matched. In addition, the number of chaff minutiae increases while the number of real minutiae decreases slightly. Hence, we can predict that both of GAR and FAR will decrease as more chaff minutiae are added. Also, we can find that the matching time increases greatly as more chaff minutiae are added.

Polynomial degree	No. Chaff	FTER	Brute-force			Proposed			Reed-Solomon		
			GAR	FAR	HTER	GAR	FAR	HTER	GAR	FAR	HTER
7	200	0.3	93.1	7.4	7.1	92.2	4.8	6.3	92.0	2.6	5.3
	300		92.3	5.4	6.6	91.3	3.1	5.9	90.8	1.1	5.1
	400		91.0	4.1	6.6	90.2	2.3	6.1	89.2	0.2	5.5
8	200	0.6	90.7	4.1	6.7	89.6	2.2	6.3	89.4	1.3	6.0
	300		90.2	3.1	6.4	89.0	1.5	6.2	88.3	0.5	6.1
	400		89.2	2.1	6.5	88.0	0.9	6.5	86.9	0.1	6.6
9	200	1.3	88.1	2.2	7.0	87.0	1.2	7.1	86.7	0.7	7.0
	300		87.9	1.2	6.7	86.6	0.5	7.0	86.0	0.1	7.1
	400		87.3	0.9	6.8	85.4	0.5	7.6	84.2	0.0	7.9
10	200	1.9	85.0	0.8	7.9	84.0	0.4	8.2	83.5	0.3	8.4
	300		84.7	0.6	8.0	83.5	0.2	8.3	82.6	0.0	8.7
	400		83.8	0.5	8.3	81.3	0.1	9.4	78.2	0.0	10.9
11	200	2.6	82.0	0.2	9.1	80.4	0.1	9.9	79.4	0.1	10.4
	300		81.7	0.3	9.3	79.2	0.1	10.5	76.8	0.0	11.6
	400		81.1	0.1	9.5	78.5	0.1	10.8	75.1	0.0	12.4
12	200	3.1	78.9	0.1	10.6	76.9	0.0	11.6	75.8	0.0	12.1
	300		78.5	0.1	10.8	75.9	0.0	12.1	72.9	0.0	13.5
	400		77.6	0.1	11.2	74.6	0.0	12.7	71.2	0.0	14.4

Table 4. Recognition accuracies of the FFV system of FVC2002-DB1 (unit: %)

To examine the effectiveness of the proposed polynomial reconstruction algorithm, we compare it with both the brute-force search and the Reed-Solomon code. The error rates of the FFV system for DB1 and DB2 are listed in Table 4 and Table 5, respectively. During the enrollment, if the number of the fingerprint minutiae is less than or equal to the degree of the polynomial, the fingerprint is rejected to be enrolled. Failure To Enrollment Rate (FTER) is the ratio of the rejected fingerprints to the total fingerprints. The FTER of DB1 is much higher than that of DB2 since fewer minutiae are extracted from DB1. To compare the recognition accuracies of the three polynomial reconstruction algorithms, Genuine Acceptance Rate (GAR) and False Rejection Rate (FAR) are used. In addition, Half Total Error Rate (HTER) is adopted for the purpose of direct comparison (Poh & Bengio, 2006), which is the average of False Rejection Rate (FRR) and False Acceptance Rate (FAR). The values of GAR, FAR and HTER are obtained by excluding the fingerprints rejected at the enrollment phase. As predicted above, the more chaff minutiae is inserted, the lower both

GAR and FAR become. Since the proposed algorithm needs one more real minutia than the brute-force search, the recognition accuracy of the proposed algorithm using the k-degree polynomial should be exactly the same as that of the brute-force search using the $(k + 1)$-degree polynomial. In practice, however, the polynomial could not always be reconstructed even if the real minutiae are matched more than the degree of polynomial because the polynomial reconstruction process will be stopped after a pre-determined number of iterations to prevent from going into an infinite loop. Furthermore, another reason is the difference in FTER due to the different degree of polynomial. The overall recognition rates of the three algorithms are comparable. The averages of HTER of the brute-force search, the proposed algorithm and the Reed-Solomon code for DB1 are 8.1%, 8.5% and 8.8%, respectively, and 6.1%, 5.8% and 5.4% for DB2. The Reed-Solomon code is better for low degree polynomials, and the brute-force search is better for high degree polynomials.

Polynomial degree	No. Chaff	FTER	Brute-force			Proposed			Reed-Solomon		
			GAR	FAR	HTER	GAR	FAR	HTER	GAR	FAR	HTER
7	200	0.3	96.9	15.4	9.3	96.1	9.3	6.6	95.2	2.5	3.7
	300		96.3	11.3	7.5	95.3	7.0	5.8	94.2	1.1	3.5
	400		95.4	9.3	6.9	94.5	5.7	5.6	93.2	0.5	3.6
8	200	0.3	95.8	9.5	6.9	94.9	5.7	5.4	94.2	1.8	3.8
	300		95.1	6.8	5.9	94.3	4.2	4.9	93.3	0.6	3.7
	400		94.7	5.8	5.5	93.9	3.5	4.8	91.8	0.3	4.3
9	200	0.4	94.3	5.2	5.4	93.2	3.0	4.9	92.3	1.1	4.4
	300		94.2	4.3	5.1	93.2	2.5	4.7	91.7	0.4	4.3
	400		93.4	3.5	5.0	91.8	2.2	5.2	90.0	0.3	5.1
10	200	0.7	92.9	3.1	5.1	91.5	1.6	5.0	91.0	0.7	4.9
	300		92.3	2.5	5.1	90.6	1.5	5.4	89.2	0.3	5.6
	400		92.0	2.0	5.0	90.5	1.2	5.3	88.3	0.1	5.9
11	200	0.9	90.4	1.8	5.7	89.1	1.0	5.9	88.5	0.5	6.0
	300		89.9	1.1	5.6	88.5	0.6	6.1	86.8	0.1	6.7
	400		89.9	1.2	5.6	88.1	0.7	6.3	85.4	0.0	7.3
12	200	1.0	88.4	1.0	6.3	86.9	0.6	6.8	86.3	0.2	7.0
	300		88.0	0.7	6.4	86.1	0.5	7.2	84.6	0.1	7.8
	400		87.3	0.7	6.7	84.8	0.3	7.8	82.2	0.0	8.9

Table 5. Recognition accuracies of the FFV system of FVC2002-DB2 (unit: %)

Fig. 4 shows the execution times (average, min, max) and the average number of the Lagrange interpolations for the brute-force search, the proposed algorithm and the Reed-Solomon code for FVC2002-DB1, respectively. Fig. 5 is the results of FVC2002-DB2. Although the Lagrange interpolation is an efficient technique to interpolate a polynomial, it requires more time. In the case of genuine tests, the average time of success is fast enough to be performed in real time. At the worst case, however, the brute-force search spends too much time because a huge number of the Lagrange interpolations are needed to reconstruct the true polynomial. On the other hand, the proposed algorithm and the Reed-Solomon code spend very little time even at the worst case. In our experiments, the Lagrange interpolation time for a 9-degree polynomial is 0.6 milliseconds, but in a certain case, it takes

more than 284 seconds because 425,415 interpolations are performed to reconstruct the polynomial.

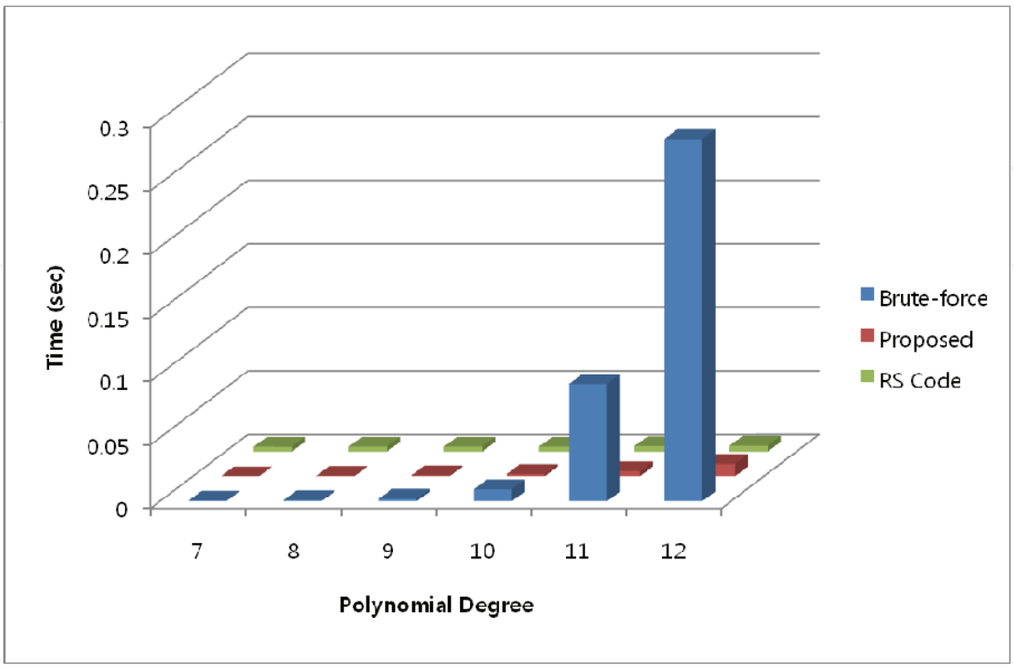


Fig. 4. Comparison of polynomial reconstruction time for the Brute-force search, the proposed algorithm, and the Reed-Solomon code (FVC2002-DB1)

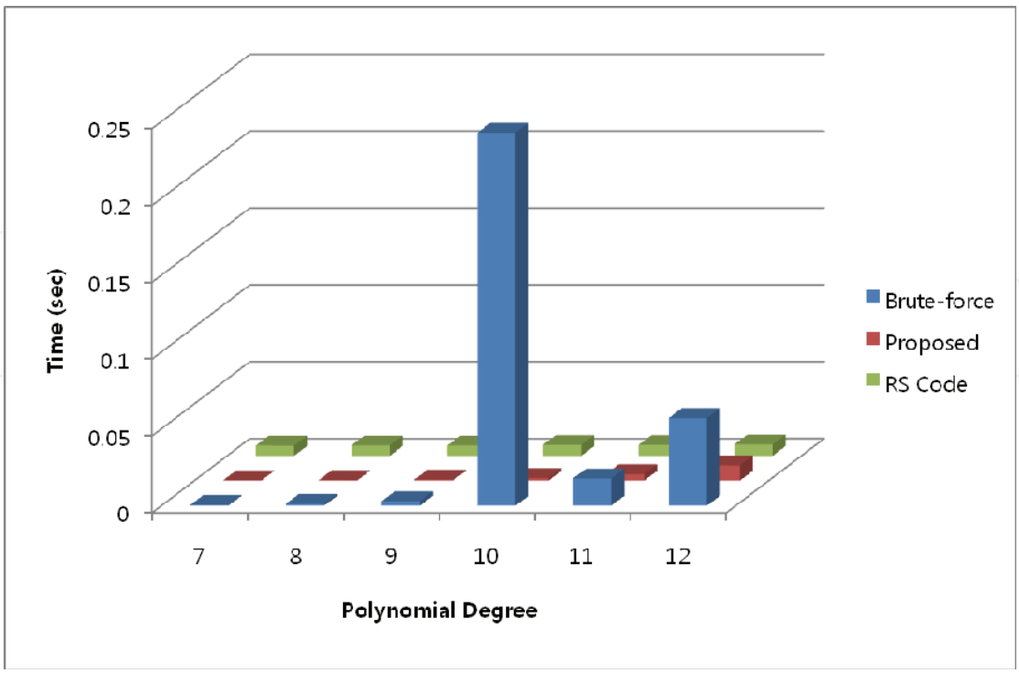


Fig. 5. Comparison of polynomial reconstruction time for the Brute-force search, the proposed algorithm, and the Reed-Solomon code (FVC2002-DB2)

4. Attack methods for fuzzy fingerprint vault

The attack of FFV is selecting real points more than the degree of polynomial. The efficient attack methods are to constitute a minutia set that contains many real points and a little chaff points. If the set contains real points more than the degree of polynomial, the polynomial can be reconstructed by the brute-force search. In addition, if the set contains more chaff points, more time is needed to reconstruct the polynomial. The correlation attack (Kholmatov & Yanikoglu, 2008) is known to be an efficient method that constitutes the minutia set using multiple vaults enrolled for different applications. On the other hand, when multiple vaults cannot be obtained and no information about the minutiae is available, the attacker should select the real minutiae from among the entire points including many chaff points.

4.1 Brute-force attack

The brute-force attack (Paar et al., 2010) is a method used to extract the real minutiae from the vault when no information is available. It tries to reconstruct the polynomial by using all the possible combinations until the correct polynomial is found. Given the $(k + 1)$ points, a k-degree polynomial is uniquely determined, which is computed by the Lagrange interpolation. Lagrange interpolating polynomial $p(x)$ of degree k that passes through $(k + 1)$ points $(x_1, y_1),\cdots,(x_{k+1}, y_{k+1})$ is given by

$$p(x) = \sum_{i=1}^{k+1} p_i(x) \tag{14}$$

where

$$p_i(x) = y_i \prod_{\substack{j=1 \\ j \neq i}}^{k+1} \frac{x - x_j}{x_i - x_j} \tag{15}$$

This is written explicitly by

$$\begin{aligned} p(x) = {} & \frac{(x-x_2)(x-x_3)\cdots(x-x_{k+1})}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_{k+1})} y_1 + \frac{(x-x_1)(x-x_3)\cdots(x-x_{k+1})}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_{k+1})} y_2 + \cdots \\ & + \frac{(x-x_1)(x-x_2)\cdots(x-x_k)}{(x_n-x_1)(x_n-x_2)\cdots(x_{k+1}-x_k)} y_{k+1} \end{aligned} \tag{16}$$

Recall that there are n_r points in the vault, and among these points, n_e points are real, and we want to reconstruct a k-degree polynomial. Then, the total number of trials which select $(k + 1)$ points from n_r points is $C(n_r, k + 1)$, and $C(n_e, k + 1)$ combinations can reconstruct the true polynomial. If we can randomly select $(k + 1)$ points and exclude the combination from the next selection, we need to try $0.5 \times C(n_r, k + 1) / C(n_e, k + 1)$ times on the average. However, it requires too much memory to check whether the selected combination is used or not. For example, if $n_r = 230$, $n_e = 7$, and char (1 byte) type is used, then $C(230, 8) \approx 172$ Terabytes, which is impossible to be allocated in RAM. Therefore, this attack can be seen as a Bernoulli trial with probability,

$$\frac{C(n_e, k+1)}{C(n_r, k+1)} \tag{17}$$

Hence, let N_L be the number of executions of the Lagrange interpolation until the correct polynomial is reconstructed, then, N_L has the geometric distribution with mean,

$$E(N_L) = \frac{C(n_r, k+1)}{C(n_e, k+1)} \tag{18}$$

In general, since n_r is much greater than n_e, this attack is time-consuming. For example, if n_r = 230, n_e = 30, and k = 9, then the average number of trials of the Lagrange interpolation is C(230, 10) / C(30, 10) ≈ 3 × 10^9. Even though the calculation of the Lagrange interpolation takes 1 millisecond, the attack will take about 36 days on the average.

4.2 Correlation attack

Scheirer et al. suggested the methods of attacks against fuzzy vault including the attack via record multiplicity, which is known as the *correlation attack* (Scheirer & Boult, 2007). Suppose that the attacker can obtain two vaults generated from the same fingerprint (different chaff minutiae and different polynomials), the real minutiae can be revealed by correlating two vaults. If the matching minutiae contain the real minutiae more than the degree of the polynomial, the true polynomial can be successfully reconstructed by the brute-force attack, and hence, all of the real minutiae will be revealed. In addition, even if the number of the real minutiae is larger than the degree of the polynomial, it would be computationally infeasible to reconstruct the polynomial when there are too many chaff minutiae in the matching minutiae with respect to the real minutiae.

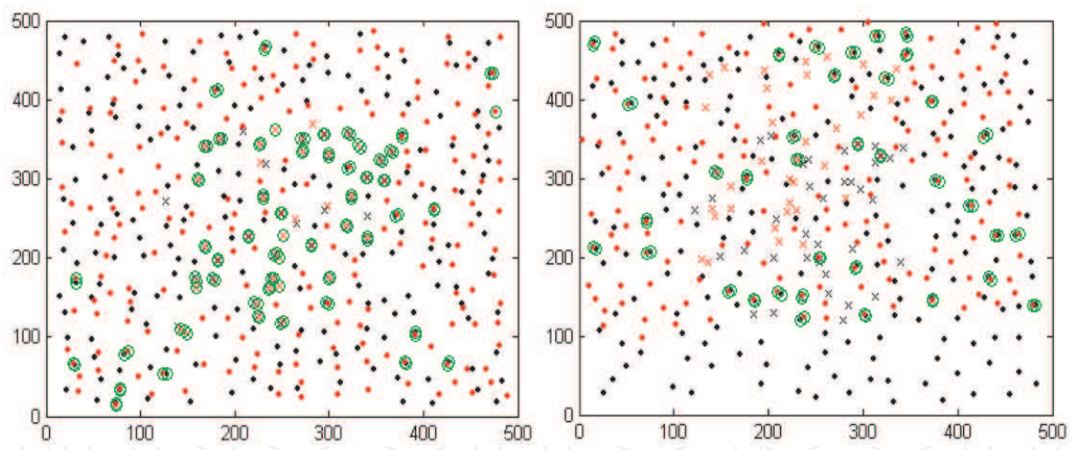


Fig. 6. Example of correlation attack. (a) shows the alignment of two vaults from the same fingerprint, and (b) from different fingerprints. (Kholmatov & Yanikoglu, 2008)

Kholmatov et al. have realized the correlation attack against a database of 400 fuzzy vaults (200 matching pairs) of their own making (Kholmatov & Yanikoglu, 2008). Fig. 6 shows an example of their experimental results. If two vaults which are generated from the same fingerprint are correlated, many real minutiae are obtained, while two vaults from different fingerprints do not have enough real minutiae in their common area to reconstruct the true polynomial. They reported that 59% of them were successfully unlocked with two matching vaults.

4.3 Fast polynomial reconstruction attack

The exhaustive search can be performed much faster than the brute-force search by using the method of the polynomial reconstruction described in Section 3.3. If a vault contains two more real points than the degree of the polynomial (i.e., $n_e \geq k + 2$), the true polynomial can be successfully reconstructed. The polynomial reconstruction time depends on the number of the real points and the number of the chaff points. The more chaff points and the less real points it contains, the more time it takes to reconstruct the polynomial. The process for the Fast Polynomial Reconstruction (FPR) attack algorithm is as follows.

First, k points are selected from n_r points with real and chaff points mixed. Second, the augmented matrix is obtained, and is reduced into the following row-echelon form.

$$\left[\begin{array}{cccccc|c} 1 & u_1 & u_1^2 & \cdots & u_1^{k-1} & u_1^k & v_1 \\ 0 & 1 & u_2^{2(2)} & \cdots & u_2^{k-1(2)} & u_2^{k(2)} & v_2^{(2)} \\ 0 & 0 & 1 & \cdots & u_3^{k-1(3)} & u_3^{k(3)} & v_3^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & u_k^{k(k)} & v_k^{(k)} \\ 0 & 0 & 0 & \cdots & 0 & 1 & v_{k+1}^{(k+1)} \\ 0 & 0 & 0 & \cdots & 0 & 1 & v_{k+2}^{(k+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & v_{n_r}^{(k+1)} \end{array}\right] \tag{19}$$

Third, if there exist the same values among $v_{k+1}^{(k+1)}, \cdots, v_{n_r}^{(k+1)}$, we reconstruct the polynomial with the k points selected and one of the two points which have the same $v^{(k+1)}$ value, and then, compare it with the true polynomial.

As in the brute-force attack, we can estimate the number of the Lagrange interpolations to be performed. First, the $(k + 1)$ points are chosen, and if at least one of the remaining $(n_r - k - 1)$ points lies on the line through the $(k + 1)$ points, the Lagrange interpolation is performed regardless of whether all these points are real points or not. Let us assume that the order of the Galois field is 2^n, then the probability that at least one of the $(n_r - k - 1)$ points lies on that line is

$$1 - \left(1 - \frac{1}{2^n}\right)^{n_r - k - 1} \tag{20}$$

Since the average number of trials until the $(k + 1)$ real points are drawn is $C(n_r, k + 1) / C(n_e, k + 1)$, the average number of the Lagrange interpolations to be performed until the real polynomial is reconstructed is as follows.

$$E(N_L) = \frac{C(n_r, k+1)}{C(n_e, k+1)} \times \left\{1 - \left(1 - \frac{1}{2^n}\right)^{n_r - k - 1}\right\} \tag{21}$$

Since 2^n is a very large number, equation (20) is very small. Hence, the Lagrange interpolation needs to be performed in a limited number of times. In addition, let N_G be the number of executions of the Gaussian elimination of equation (19), then, the expected value of N_G is as follows.

$$E(N_G) = \frac{C(n_r, k)}{C(n_e, k)} \tag{22}$$

It is smaller than the expected number of the Lagrange interpolation of the brute-force attack in equation (18). Also, the execution time per combination of the FPR attack is much faster than that of the brute-force attack. Therefore, this attack is more efficient than the brute-force attack. Experimental results show the efficiency of this attack method. For example, if n = 20, k = 9, n_r = 230, and n_e = 30, the average numbers of executions of Lagrange interpolation and the recursive formula are

$$E(N_L) = \frac{C(230,10)}{C(30,10)} \times \left\{ 1 - \left(1 - \frac{1}{2^{20}} \right)^{220} \right\} \approx 7 \times 10^5 \tag{23}$$

$$E(N_G) = \frac{C(230,9)}{C(30,9)} \approx 3 \times 10^8 \tag{24}$$

It is considerable reduction compared to the brute-force attack. The number of the Lagrange interpolation is reduced by the order of 10^4.

4.4 Experimental results

To compare the three attack algorithms mentioned in the previous section (the brute-force attack, the correlation attack, and the FPR attack), we used DB1 of FVC2002 (Maio et al., 2002), which consists of 8 impressions for each of the 100 distinct fingers. From among 8 impressions, the first impressions of each fingerprint were used for the brute-force attack and the FPR attack. For the correlation attack, on the other hand, the first and the second impressions were used to get the correlated minutiae when the correlation reached its peak. Once the matched minutiae are obtained, the polynomial reconstruction is performed by the brute-force search. We chose the number of chaff minutiae as 200, and the degree of the polynomial as 7. The tests are performed to find out whether the attack algorithms can reconstruct the true polynomial or not within 24 hours.

The results of the three attack algorithms are summarized in Table 6. The success rate is defined by the ratio of the number of successes to the total trials. The Correlation attack is known to be very efficient attack method for FFV. However, the brute-force attack and the FPR attack turn out to be much more efficient methods. Especially, the FPR attack cracked 100% of the vaults, and the attack time is only half of that of the brute-force attack.

Attack Method	Brute-force	Correlation	FPR
Success Rate (%)	95	17	100
No. Lagrange (N_L)	3.9×10^7	4.4×10^7	2.0×10^5
$E(N_L)$	6.6×10^7	2.4×10^8	9.6×10^4
No. Gaussian Elimination (N_G)	-	-	9.2×10^6
$E(N_G)$	-	-	2.1×10^7
Time (sec)	4,562	5,089	2,424

Table 6. The summary of the three attack algorithms (brute-force attack, correlation attack, and the FPR attack)

For the experiments of the correlation attack, after correlating two vaults, 23 tests extract more than 8 real minutiae, and perform polynomial reconstruction. The average correlation time is 42 seconds. Among 23 tests, 6 tests cannot reconstruct the true polynomial within 24 hours, so 17% of the vaults are cracked by the correlation attack. On the other hand, 100% of the vaults are cracked by the FPR attack within 24 hours, and the average time is 2,424 seconds, while the average time for the brute-force attack is 4,562 seconds. Since the fixed numbers of chaff minutiae are inserted, the smaller the number of real minutiae is, the more time the polynomial reconstruction requires. For the brute-force attack and the correlation attack, the attack time is proportional to the actual number of the Lagrange interpolations. Also, the time for FPR attack depends mainly on the number of the Gaussian eliminations which is computed by equation (13).

Fig. 7 shows the histogram of the actual number of Lagrange interpolation and its expected value for the case of successful attack. Although the two histograms have some fluctuations, they have similar distributions. Therefore, we can predict the attack time based on the expected number of the Lagrange interpolations when the number of chaff points is more than 200 or the degree of polynomial is greater than 7. In our experiments, the Lagrange interpolation times for the polynomial degree of 7, 8, 9, 10, 11 and 12 are 0.14, 0.28, 0.71, 1.6, 3.9 and 9.8 milliseconds, respectively. Also, the Gaussian elimination times for 200 and 400 chaff points are 0.26 and 0.52 milliseconds, respectively. The expected numbers of the Lagrange interpolations and the Gaussian eliminations can be calculated from the equations (18), (21) and (22). Table 7 shows the predicted time for the brute-force attack and the FPR attack. The security of FFV can be strengthened by adding more chaff points and by increasing the degree of polynomial. From the experimental result in the previous section, however, the more chaff points are added and the higher degree of polynomial is used, the recognition accuracy degrades significantly.

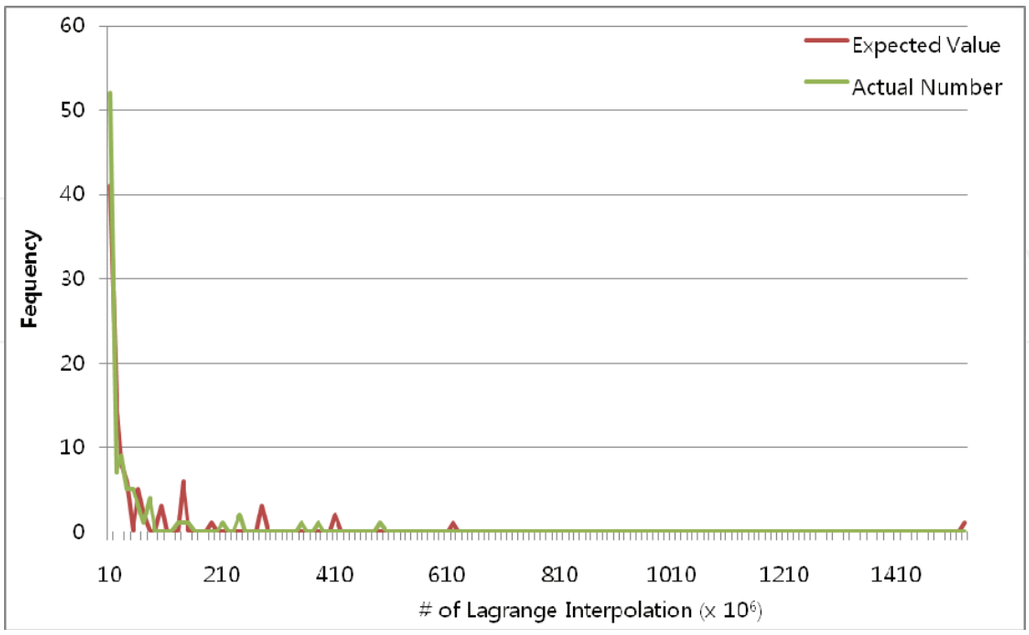


Fig. 7. Histograms of the actual number of the Lagrange interpolation and its expected value

No. Chaff	Polynomial Degree	Brute-force	FPR
200	8	10 hours	1 hour
	9	9 days	9 hours
	10	213 days	4 days
	11	14 years	35 days
	12	369 years	1 year
400	7	3 days	18 hours
	8	113 days	13 days
	9	14 years	214 days
	10	595 years	11 years
	11	26,964 years	200 years
	12	1,365,157 years	4,172 years

Table 7. Predicted time for the brute-force attack and the FPR attack

5. Conclusion

Biometrics is an efficient and convenient method for authenticating persons' identities. However, once the biometric template is compromised, the user's identity is permanently stolen. Hence, many scientists have studied to protect the biometric templates against attack. Fuzzy vault gave a promising solution to user privacy and fingerprint template security problems. Inspired from fuzzy vault, fuzzy fingerprint vault was proposed to securely store fingerprint minutiae in a database. However, there are two problems for the fuzzy fingerprint vault to be used in real world. First, by using brute-force search, the polynomial cannot be reconstructed in real time. Second, fuzzy vault is vulnerable to correlation attack.

In this work, we provided solutions to these problems. First, we proposed a fast polynomial reconstruction algorithm, which speed up the exhaustive search by using the Consistency Theorem. To reduce the execution time, it determines the candidate sets with chaff points by using Gaussian elimination and excludes them from the reconstruction trial. Since Gaussian elimination is a time-consuming process, we have found a recursive formula to perform Gaussian elimination effectively. We confirmed that the proposed algorithm can be performed in real time even at the worst case.

Second, fuzzy vault was found out to be cracked quickly by the correlation attack in 2008. The correlation attack acquires a minutiae set with many real minutiae by correlating two vaults. However, if the minutia set contains a little more chaff minutiae, the attack can hardly crack the vault. In our experiments, brute-force attack was rather more efficient. In addition, the fast polynomial reconstruction algorithm is used to crack the vault. The FPR attack algorithm records 100% attack rate. Therefore, fuzzy fingerprint vault cannot store fingerprint minutiae securely anymore. Furthermore, if we add more chaff points and use higher degree of polynomial to strengthen the security, in return, the recognition accuracy degrades significantly. Therefore, a solution for enhancing security of FFV is required. One possible solution is one-time template (Ueshige & Sakurai, 2006) whose notion is from one-time password. If we can define a one-time template for fingerprint and the corresponding transform, the security of fingerprint authentication system can be enhanced innovatively.

6. Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. 2009-0086 148).

7. References

Anton, H. (1994). *Elementary Linear Algebra (7th Edition)*, John Wiley & Sons, Inc., ISBN 0-471-30569-3, New York

Bolle, R.; Connell, J. & Ratha, N. (2002). Biometrics Perils and Patches. *Pattern Recognition*, Vol. 35, No. 12, (December 2002), pp. 2727-2738, ISSN 0031-3203

Choi, W.Y.; Lee, S.; Moon, D.; Chung, Y. & Moon, K.Y. (2008). A Fast Algorithm for Polynomial Reconstruction of Fuzzy Fingerprint Vault. *IEICE Electronics Express*, Vol. 5, No. 18, (2008), pp. 725-731, ISSN 1349-2543

Choi, W.Y.; Moon, D.; Moon, K.Y. & Chung, Y. (2009). A New Alignment Algorithm of Fuzzy Fingerprint Vault Without Extra Information, *Proceedings of IASTED International Conference on Artificial Intelligence and Applications*, pp. 197-201, ISBN 978-0-88986-780-2, Innsbruck, Austria, February 2009

Chung, Y.; Moon, D.; Lee, S.; Jung, S.; Kim, T. & Ahn, D. (2006). Automatic Alignment of Fingerprint Features for Fuzzy Fingerprint Vault, *LNCS 3822: Proceedings of 1st SKLOIS Conference on Information Security and Cryptology*, pp. 358-369, ISSN 0302-9743, March 2006

Clancy, T.; Kiyavash, N. & Lin, D. (2003). Secure Smartcard-based Fingerprint Authentication, *Proceedings of ACM SIGMM Workshop on Biometrics Methods and Applications*, pp. 45-52, ISBN 1-58113-779-6, 2003

Dodis, Y.; Ostrovsky, R.; Reyzin, L. & Smith, A. (2004). Fuzzy Extractors: How To Generate Strong Keys from Biometrics and Other Noisy Data, *LNCS 3027: Proceedings of Eurocrypt*, pp. 523-540, ISSN 0302-9743, Interlaken, Switzerland, 2004

Gao, S. (2003). A New Algorithm for Decoding Reed-Solomon Codes, *Communications, Information and Network Security (V. Bhargava, H.V. Poor, V. Tarokh and S. Yoon, Edition)*, pp. 55-68, Kluwer Academic Publishers, ISBN 978-1-4020-7251-2

Gathen, J. von zur & Gerhardt, J. (2003). *Modern Computer Algebra (2nd Edition)*, Cambridge University Press, ISBN 0-521-82646-2

Hildebrand, F. (1987). *Introduction to Numerical Analysis (2nd Edition)*, Dover Publications, ISBN 0-486-65363-3, New York

Juels, A. & Sudan, M. (2002). A Fuzzy Vault Scheme, *Proceedings of IEEE International Symposium on Information Theory*, p. 408, ISBN: 0-7803-7501-7, IEEE Press, Lausanne, Switzerland, 2002

Kanak, A. & Sogukpinar, I. (2007). Fingerprint Hardening with Randomly Selected Chaff Minutiae, *Proceedings of 12th International Conference on Computer Analysis of Images and Patterns*, pp. 383-390, ISBN 978-3-540-74271-5, 2007

Kholmatov, A. & Yanikoglu, B. (2008). Realization of Correlation Attack against the Fuzzy Vault Scheme, *Proceedings of SPIE Symposium on Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, Vol. 6819, pp. 1-7, ISBN 978-0-819-46991-5, 2008

Li, Q.; Liu, Z. & Niu, X. (2006). Analysis and Problems on Fuzzy Vault Scheme, *Proceedings of 2nd International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 244-250, ISBN 0-7695-2745-0, December 2006

Nanni, L. & Lumini A. (2009). Descriptors for Image-based Fingerprint Matchers, *Expert Systems with Applications*, Vol. 36, No. 10, (December 2009), pp. 12414-12422, ISSN 0957-4174

Maio, D.; Maltoni, D.; Cappelli, R.; Wayman, J. & Jain, A. (2002). FVC2002: Second Fingerprint Verification Competition, *Proceedings of 16th International Conference on Pattern Recognition*, pp. 811-814, ISSN 1051-4651, 2002

Nandakumar, K.; Jain, A. & Pankanti, S. (2007). Fingerprint-based Fuzzy Vault: Implementation and Performance. *IEEE Transactions on Information Forensics and Security*, Vol. 2, No. 4, (2007), pp. 744-757, ISSN 1556-6013

Paar, C.; Pelzl, J. & Preneel, B. (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*, Springer, ISBN 978-3-642-04100-6, New York

Pan, S.B.; Moon, D.; Gil, Y.; Ahn, D. & Chung, Y. (2003). An Ultra-low Memory Fingerprint Matching Algorithm and its Implementation on a 32-bit Smart Card. *IEEE Transactions on Consumer Electronics*, Vol. 49, No. 2, (July 2003), pp. 453-459, ISSN 0098-3063

Poh, N. & Bengio, S. (2006). Database Protocol and Tools for Evaluating Score-Level Fusion Algorithms in Biometric Authentication. *Pattern Recognition*, Vol. 39, No. 2, (2006), pp. 223-233, ISSN 0031-3203

Ratha, N.; Connell, J. & Bolle, R. (2001). Enhancing Security and Privacy in Biometrics-based Authentication Systems. *IBM Systems Journal*, Vol. 40, No. 3, (March 2001), pp. 614-634, ISSN 0018-8670

Ratha, N.; Chikkerur, S.; Connell, J. & Bolle, R. (2007). Generating Cancelable Fingerprint Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 4, (April 2007), pp. 561-572, ISSN 0162-8828

Scheirer, W. & Boult, T. (2007). Cracking Fuzzy Vaults and Biometric Encryption, *Proceedings of Biometrics Symposium*, pp. 1-6, ISBN 978-1-424-41548-9, Baltimore, MD, USA, September 2007

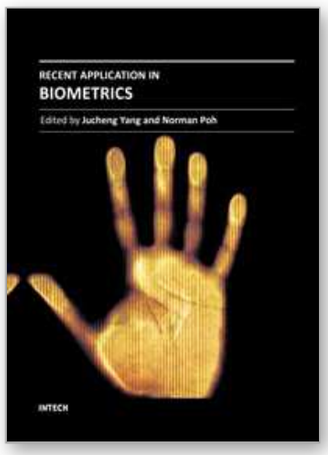
Shin, S.; Lee, M.; Moon, D. & Moon, K. (2009). Dictionary Attack on Functional Transform-Based Cancelable Fingerprint Templates. *ETRI Journal*, Vol. 31, No. 5, (October 2009), pp. 628-630, ISSN 1225-6463

Stallings, W. (2005). *Cryptography and Network Security: Principles and Practices (4th Edition)*, Prentice Hall, ISBN 978-0-131-87316-2

Ueshige, Y. & Sakurai, K. (2006). A Proposal of One-Time Biometric Authentication, *Proceedings of International Conference on Security and Management*, pp. 78-83, ISBN 1-60132-001-9, Las Vegas, Nevada, USA, June 2006

Uludag, U.; Pankanti, S. & Jain, A. (2005). Fuzzy Vault for Fingerprints, *LNCS 3546: Proceedings of 5th International Conference on Audio- and Video-Based Biometric Person Authentication*, pp. 310-319, ISSN 0302-9743, New York, USA, July 2005

Yang, J.C. & Park, D.S. (2008). A Fingerprint Verification Algorithm Using Tessellated Invariant Moment Features, *Neurocomputing*, Vol. 71, Issues 10-12, (June 2008), pp. 1939-1946, ISSN 0925-2312



Recent Application in Biometrics
Edited by Dr. Jucheng Yang

ISBN 978-953-307-488-7
Hard cover, 302 pages
Publisher InTech
Published online 27, July, 2011
Published in print edition July, 2011

In the recent years, a number of recognition and authentication systems based on biometric measurements have been proposed. Algorithms and sensors have been developed to acquire and process many different biometric traits. Moreover, the biometric technology is being used in novel ways, with potential commercial and practical implications to our daily activities. The key objective of the book is to provide a collection of comprehensive references on some recent theoretical development as well as novel applications in biometrics. The topics covered in this book reflect well both aspects of development. They include biometric sample quality, privacy preserving and cancellable biometrics, contactless biometrics, novel and unconventional biometrics, and the technical challenges in implementing the technology in portable devices. The book consists of 15 chapters. It is divided into four sections, namely, biometric applications on mobile platforms, cancelable biometrics, biometric encryption, and other applications. The book was reviewed by editors Dr. Jucheng Yang and Dr. Norman Poh. We deeply appreciate the efforts of our guest editors: Dr. Girija Chetty, Dr. Loris Nanni, Dr. Jianjiang Feng, Dr. Dongsun Park and Dr. Sook Yoon, as well as a number of anonymous reviewers.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Woo Yong Choi, Jin-Won Park and Yongwha Chung (2011). Protection of the Fingerprint Minutiae, Recent Application in Biometrics, Dr. Jucheng Yang (Ed.), ISBN: 978-953-307-488-7, InTech, Available from: http://www.intechopen.com/books/recent-application-in-biometrics/protection-of-the-fingerprint-minutiae

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen