World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:11, 2009

Fuzzy Fingerprint Vault using Multiple Polynomials

Daesung Moon, Woo-Yong Choi and Kiyoung Moon

***Abstract*—**Fuzzy fingerprint vault is a recently developed cryptographic construct based on the polynomial reconstruction problem to secure critical data with the fingerprint data. However, the previous researches are not applicable to the fingerprint having a few minutiae since they use a fixed degree of the polynomial without considering the number of fingerprint minutiae. To solve this problem, we use an adaptive degree of the polynomial considering the number of minutiae extracted from each user. Also, we apply multiple polynomials to avoid the possible degradation of the security of a simple solution(*i.e.*, using a low-degree polynomial). Based on the experimental results, our method can make the possible attack difficult 2^{192} times more than using a low-degree polynomial as well as verify the users having a few minutiae.

***Keywords*—**Fuzzy Vault, Fingerprint Recognition Multiple Polynomials.

I. Introduction

IN spites of many advantages of biometric systems, they are hampered by their security and privacy problems[1]. That is, once the systems are compromised, the biometric data is compromised permanently and cannot be reissued. This is urgent problem in the biometric community.

In the *fuzzy vault* proposed by Juels and Sudan[2], Alice can place a secret value *S* in a vault and lock it using an unordered locking set ***L***. Bob, using an unordered unlocking set ***U***, can unlock the vault only if ***U*** overlaps with ***L*** to a great extent.

Based on the fuzzy vault, some implementation results for fingerprint have been reported[3-5]. However, the previous researches are not realistic in the sense that they use a fixed-degree polynomial without considering the number of minutiae extracted from each user. That is, their system cannot handle the fingerprint having less number of minutiae than the polynomial degree. For large-scale applications which should handle users having a few minutiae, this is a very important issue, but has not been addressed.

In this paper, we propose a method for applying the fingerprint having a few minutiae to the fuzzy vault without sacrificing the security. To make the system more practical, we use multiple polynomials adaptively, instead of using a fixed-degree polynomial.

Daesung Moon is with Biometrics Technology Research Team, ETRI, 161, Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea (phone : +82-42-860-1083; fax : +82-42-860-1471; e-mail: daesung@etri.re.kr).

Woo-Yong Choi is with Biometrics Technology Research Team, ETRI, 161, Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea (e-mail: wychoi4@etri.re.kr).

Kiyoung Moon is with Biometrics Technology Research Team, ETRI, 161, Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea (e-mail: kymoon@etri.re.kr).

II. Fuzzy fingerprint vault

A fingerprint minutia represented by $m_i=(x_i,y_i,\theta_i,t_i)$ is composed of four elements: *x-*, *y-coordinates*, *angle*, and *type*. The fuzzy fingerprint vault system is composed of two steps, locking and unlocking. For the purpose of explanation of the proposed method in the following section, each step of the fuzzy fingerprint vault is explained in the following.

Locking Processing:

① Extract minutiae from a template fingerprint image of a user. These minutiae are called as real minutiae.

$$L=\{(x_i,y_i,\theta_i,t_i)\,|\,i=1,...,n\}, \tag{1}$$

where *n* denote the number of minutiae.

② Generate a degree-*k* polynomial from a secret(*S*), and compute a hash value κ from a hash function *hash*(*S*)

$$p(x)=a_0+a_1x+\cdots+a_kx_k \tag{2}$$

$$S=(a_0\,||\,a_1\,||\cdots||\,a_k) \tag{3}$$

$$a_i\in \mathrm{GF}(p^2) \tag{4}$$

$$\kappa=hash(S) \tag{5}$$

③ Compute the polynomial projections, $p(x)$, after converting all elements of L to an element of GF(p^2), and define this result as Set R_L. For example, if an element of GF(p^2) is represented as $AX+B$ ($A,B\in$ GF(p^2)), we can replace *x* and *y* coordinates of the minutia to *A* and *B*, respectively.

$$R_L=\{(r_i,v_i)\,|\,i=1,...,n\},\ \ r_i=(x_i,y_i,\theta_i,t_i) \tag{6}$$

$$v_i=p(X_i),\ \ X_i=x_iX+y_i\in \mathrm{GF}(p^2),\ \ i=1,...,n \tag{7}$$

④ Randomly generate chaff minutiae that do not lie on $p(x)$ to protect real minutiae.

$$C=\{(c_i,v_i)\,|\,i=n+1,...,r\},\ \ c_i=(x_i,y_i,\theta_i,t_i) \tag{8}$$

$$v_i=p(X_i)+\alpha_i, X_i=x_iX+y_i\in \mathrm{GF}(p^2),\ \ i=n+1,...,r, \tag{9}$$

where α_i is a non-zero element over finite fields of the form GF(p^2).

⑤ Randomly generate Set *R* that is integrated with R_L and *C*.

$$R=\{(r_i,v_i)\,|\,i=1,...,r\},\ \ r_i=(x_i,y_i,\theta_i,t_i) \tag{10}$$

⑥ Finally, the vault is constituted by the real and chaff

minutiae, the degree k of the polynomial and the secret. The secret should be stored in a hashed form.

$$V = \{(r_i, v_i), \kappa, k \mid i = 1,..., r\} \tag{11}$$

Unlocking Processing:

Unlocking processing is the step that reconstructs the polynomial from minutiae of input fingerprint image.

① Extract minutiae from input fingerprint image.

$$U = \{(x'_i, y'_i, \theta'_i, t'_i) \mid i = 1,..., m\}, \tag{12}$$

where m denotes the number of minutiae.

② Execute fingerprint matching with Set U and r_i of Set V stored in the Locking processing. The matching results are stored in Set M with t matched minutiae and corresponding v_i in Set V.

$$M = \{(m_i, v_i) \mid i = 1,..., t\}, m_i = (x_i, y_i, \theta_i, t_i), \tag{13}$$

where $M \in R, t \le r$.

③ If k and M are used as input values for the RS_{DECODE}, the degree-k polynomial($p'(x)$) will be returned. Then, κ' is computed by Eq. 16.

$$p'(x) = \mathrm{RS}_{\mathrm{DECODE}}(k, M) \tag{14}$$

$$p'(x) = a'_0 + a'_1 x + \cdots + a'_k x_k \tag{15}$$

$$\kappa' = hash(a'_0 \| a'_1 \| \cdots \| a'_k) \tag{16}$$

④ If κ' and κ are exactly same, the user is accepted. Otherwise, he is rejected.

$$\text{Decision} = \begin{cases} \text{Accept}, & if \quad \kappa' = \kappa \\ \text{Reject}, & otherwise \end{cases} \tag{17}$$

If Set M contains k+1 real minutiae, the fuzzy fingerprint vault can reconstruct the same polynomial used in the locking process.

Eq. 18 shows a complexity of the system in case of a brute-force attack to select k+1 real minutiae from the vault including both real and chaff minutiae. From Eq. 18, the complexity can be increased as adding more chaff minutiae and using a higher-degree of the polynomial under the $n>k$ condition.

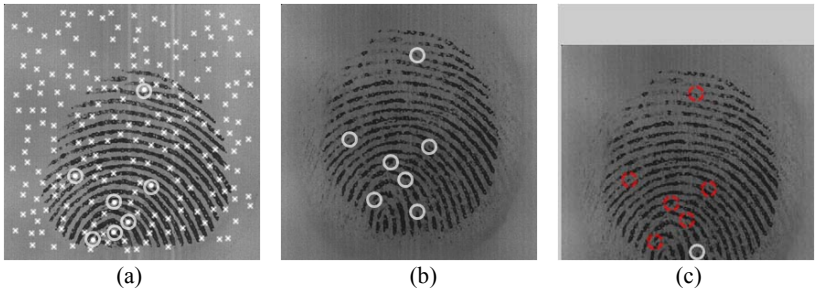
$$Complexity\ 1 = {}_rC_{k+1} / {}_nC_{k+1} \tag{18}$$

We can consider another possible attack, called *coefficient attack*, that reconstructs the polynomial directly by guessing the coefficients of the polynomial. In this case, the complexity can be represented as Eq. 19, where k and l are the degree of the polynomial and the bit length of the coefficient, respectively. The complexity can be increased with a higher-degree of the polynomial because l is decided by the image size.

$$Complexity\ 2 = 2^{((k+1)*l)} \tag{19}$$

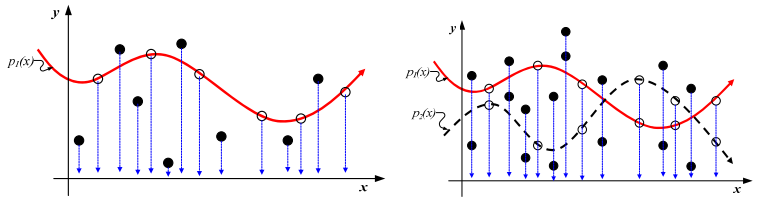
Note that the previous researches[3-5] are not applicable to the fingerprint having a few minutiae since they use a fixed-degree of the polynomial without considering the number of minutiae extracted from each user. For example, Fig. 1 shows the result of the fingerprint verification of the fuzzy fingerprint vault. As shown in Fig. 1(a), 7 real minutiae are extracted from a template fingerprint image, and then 200 chaff minutiae are added. In the verification stage, 7 minutiae are extracted from input fingerprint image(Fig. 1(b)), and then 6 real minutiae(represented as dotted red circle) are matched between Fig. 1(a) and (b) after aligning them exactly(Fig. 1(c)). If the system uses a fixed-degree of the polynomial(*i.e.*, degree of more than 7), the user cannot be accepted as a legitimated user, in spite of using the same finger.

The straightforward method to solve this problem is to set the fixed-degree of the polynomial to be one less than the minimum number of minutiae extracted from all users. However, this simple method may degrade the security of the fuzzy fingerprint vault because the security of the system is based on the difficulty of the polynomial reconstruction problem. For example, let's consider two scenarios when the number of real and chaff minutiae are 30 and 200, respectively. Scenario A uses a degree-3 polynomial with 16-bit length coefficients and scenario B uses a degree-12 polynomial. According to Eq. 18, the complexities of scenario A and scenario B are about 4.1×2^9 and 4.7×2^{36}, respectively. Also, according to Eq. 19, the complexities of scenario A and scenario B are $2^{64}(=2^{((3+1)\times16)})$ and $2^{208}(=2^{((12+1)\times16)})$ evaluations, respectively. Note that, the time for each evaluation using Eq. 18 is much larger than that of



(a) (b) (c)

Fig. 1. An example of the fuzzy fingerprint vault with a few minutiae. (a) template fingerprint image(O : real minutiae, X : chaff minutiae), (b) input fingerprint image, (c) result of verification after alignment



(a) (b)

Fig. 2. Fuzzy fingerprint vault and the polynomials(white circle : real minutiae, black circle : chaff minutiae); (a) previous method (b) proposed method

Eq. 19, and thus it is difficult to compare the complexities directly between Eq. 18 and Eq. 19. However, it is clear that the security of scenario A is degraded significantly from scenario B using either Eq. 18 or Eq. 19 (although scenario A can handle the fingerprints having a few minutiae).

III. A Proposed Method Using Multiple Polynomials

In this paper, we propose a method for applying the fingerprint having a few minutiae to the fuzzy vault without sacrificing the security. To implement a practical fuzzy fingerprint vault system, we use multiple polynomials in the locking process as shown in Fig. 2(b), instead of the typical method using only one polynomial as shown in Fig. 2(a)[3-5]. That is, each element of the real minutiae is projected on the two different polynomial $p_1(x)$ and $p_2(x)$.

In the typical fuzzy fingerprint vault using one polynomial, each x value has its corresponding y value projected on $p_1(x)$ as shown in Fig. 2(a). In this case, if the degree of the polynomial is low, the security problem can be occurred in the fuzzy fingerprint vault system as explained in the previous section. However, if each x value has its corresponding y values projected on both $p_1(x)$ and $p_2(x)$ as shown in Fig. 2(b), we can solve the security problem. Because these two polynomials are independent with each other, it is difficult for an attacker to generate the two polynomials correctly. Chaff minutiae protecting real minutiae also have two y values that do not lie on both $p_1(x)$ and $p_2(x)$ in the proposed method. Real and chaff minutiae are represented as white and black circles in the Fig. 2, respectively.

Therefore, Eq. 11 of the locking process ⑥ has to be modified as Eq. 20. v_{1i} and v_{2i} are computed by Eq. 21 and 22. That is, if r_i are real minutiae, v_{1i} and v_{2i} are the results projected on the polynomial $p_1(x)$ and $p_2(x)$. Otherwise, v_{1i} and v_{2i} are the generated values that do not lie on both $p_1(x)$ and $p_2(x)$. Also, Eq. 12 in the unlocking process ② is redefined as Eq. 23.

In the unlocking process ③, $\mathrm{RS}_{\mathrm{DECODE}}$ may return two polynomials $p'_1(x)$ and $p'_2(x)$ using k and M that is redefined in Eq. 24.

$$V = \{(r_i, v_{1i}, v_{2i}), \kappa, k \mid i = 1,...,r\} \tag{20}$$

$$v_{1i} = \begin{cases} p_1(X_i) & , X_i \text{ is real minutiae} \\ p_1(X_i) + \alpha & , X_i \text{ is chaff minutiae} \end{cases} \tag{21}$$

$$v_{2i} = \begin{cases} p_2(X_i) & , X_i \text{ is real minutiae} \\ p_2(X_i) + \alpha & , X_i \text{ is chaff minutiae} \end{cases} \tag{22}$$

$$X_i = x_i X + y_i \in \mathrm{GF}(p^2), i = 1,...,r \tag{23}$$

$$M = \{(m_i, v_{1i}, v_{2i}) \mid i = 1,...,t\}, m_i = (x_i, y_i, \theta_i, t_i) \tag{24}$$

TABLE I. Parameter Setting for the Multiple-Polynomials Method.

# of Real Minutiae	4~5	6~10	11~15	16~
Degree of Polynomial	3	4	6	7~12
# of Polynomial	4	3	2	1

TABLE II. Performance Comparison

	Previous method		Proposed method	
Degree	FAR (%)	GAR (%)	FAR (%)	GAR (%)
7	0.18	90.79	0.18	91.68
8	0.12	86.93	0.12	88.21
9	0.08	82.86	0.08	84.54
10	0.02	78.36	0.02	80.50
11	0.02	72.46	0.02	74.68
12	0.00	70.25	0.00	72.57

IV. Experimental Results

To evaluate the performance of the proposed method using multiple polynomials, we used FVC2002-DB1 Set A[6]. The database is consisted 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). All experiments were performed on a system with a 2.66GHz processor, and the number of chaffs was 200.

To examine the effectiveness of the proposed method, we implemented multiple configurations(*e.g.*, the degree of the polynomial, the number of polynomials) based on the number of real minutiae(see Table I). Note that, for fingerprints having a few minutiae, we determined these parameters to make the total degree(*i.e.*, sum of each degree of multiple polynomials) to be 12(*i.e.*, the maximum degree of the previous methods). We applied our multiple-polynomials method to the fingerprints having less than 15 minutiae because of having too little overlap between different acquisitions of the same fingerprints. For fingerprints having more than 15 minutiae, the fixed-degree polynomial method was applied like the previous researches.

As shown in Table II, the proposed method can improve the

performance of GAR without any degradation of FAR(Note that the degree of the polynomial in the proposed method is not shown in Table II because each fingerprint has a different number of degree). Especially, if the system uses a degree-10 polynomial without considering the number of template minutiae, the previous method cannot handle the fingerprints having less than 10 minutiae(i.e., regard the legitimate user as an attacker). In the experiment, there were only 35 cases for the fingerprints having less than 10 template minutiae. If the test scenario has more cases for the fingerprints having a few minutiae, then the GAR gap between the previous and the proposed methods can be increased.

Also, if the system uses only one polynomial of degree 3, the security against the coefficient attack with the 4 coefficients is 2^{64} according to Eq. 19. The security of our system that uses 4 polynomials of degree 3 can be increased to 2^{256}.

V. Conclusion

Although the fuzzy fingerprint vault is one of the best solutions to protect the fingerprint template securely, it is difficult to implement a practical system without considering the number of minutiae extracted from each user. The previous results cannot handle the fingerprint having a few minutiae, because they use a fixed-degree polynomial. To solve this problem, we decided the degree of polynomial adaptively by considering the number of minutiae. Furthermore, we applied multiple polynomials to improve the security of the fuzzy fingerprint vault with a low-degree polynomial. Based on the experimental results, we confirm that the proposed method can enhance not only the security but also the performance of GAR without any degradation of FAR. Furthermore, with the advance of computing power, our method can maintain the required security(*i.e.*, scalable) by adjusting the number of polynomials and the degree of each polynomial.

Acknowledgment

This work was supported by the IT R&D program of MKE/IITA[2007-S-020 Development of privacy enhanced biometric system].

References

[1] D. Maltoni, et al., *Handbook of Fingerprint Recognition*, Springer, 2003.
[2] A. Juels and M. Sudan, "A Fuzzy Vault Scheme," *Proc. of Symp. on Information Theory*, 2002, pp. 408.
[3] T. Clancy, N. Kiyavash, and D. Lin, "Secure Smartcard-based Fingerprint Authentication," *Proc. of ACM SIGMM Multim., Biom. Met. & App*., 2003, pp. 45-52.
[4] U. Uludag, S. Pankanti, and A. Jain, "Fuzzy Vault for Fingerprints," *LNCS 3546 - Proc. of AVBPA*, 2005, pp. 310-319.
[5] Woo Yong Choi, et al., "A fast algorithm for polynomial reconstruction of fuzzy fingerprint vault," *IEICE Electronics Express*, Vol. 5, No. 18, 2008, pp. 725-731.
[6] http://bias.csr.unibo.it/fvc2002/databases.asp.