IET Biometrics

Research Article

IET Journals
The Institution of Engineering and Technology

Cancellable fuzzy vault with periodic transformation for biometric template protection

ISSN 2047-4938
Received on 12th May 2015
Revised on 29th September 2015
Accepted on 2nd November 2015
doi: 10.1049/iet-bmt.2015.0029
www.ietdl.org

Tran Khanh Dang[1,2], Quynh Chi Truong[1], Thu Thi Bao Le[1] ✉, Hai Truong[1]

[1]*Faculty of Computer Science and Engineering, HCMC University of Technology, VNUHCM, Ho Chi Minh, Vietnam*
[2]*Institute for Application Oriented Knowledge Processing, Johannes Kepler University Linz, Linz, Austria*
✉ *E-mail: thule@hcmut.edu.vn*

Abstract: Nowadays, biometrics-based authentication is playing a potential approach for many modern applications such as banking, homeland security etc. However, the end-users may feel uncomfortable to deploy this technology because of not well-solved accurate rate and security problems. To overcome these issues, some significant techniques have been proposed such as biometric template protection, reducing biometric extraction noise etc. Fuzzy vault is one of the most popular methods for biometric template security, which binds a secret key with biometric features and produces one kind of data, called the helper data, for recovering the secret key. Unfortunately, the major drawback of this approach is the lacking of cancellable property. Furthermore, most of the fuzzy vault schemes are performed on two biometrics modalities: fingerprints and iris. Some techniques were introduced to transform the original biometric feature to cancellable one. However, the computational cost of these proposals was quite large. In this research, the authors introduce a periodic transformation attached to fuzzy vault to produce the new cancellable scheme. Their transformation is not only simpler but also suitable for many kinds of biometrics modalities. The experiments demonstrate that this approach is practical with a little better error rate in comparison with the original biometric feature.

1 Introduction

As we all know, the traditional authentication schemes usually based on something the user has (such as: smart card) or something the user knows (such as: password, PIN). However, those techniques have several limitations. For example, they cannot distinguish between an authorised user and those who know the correct password [1]. Therefore, we have to choose a strong password and always to keep it in mind.

In recent years, with the rapid development of technologies, biometrics-based authentication systems are becoming potential. Since biometrics is literally stuck to an individual, it can prevent the use of several identities by a single individual. The term biometric (from the Greek for bio = life, metric = degree) refers to authentication by means of biological (more accurately, physiological, or behavioural) features (such as: face, voice, fingerprint, …) [2]. Using biometrics can overcome the above limitations. However, it still raises some security and privacy concerns. For example, biometrics is secure but not secret, because voice, face, … can be easily recorded and may be misused without the user's consent. Another problem is that unlike passwords, cryptographic keys, or personal identification numbers (PINs), biometrics cannot be changed once compromised. In addition, a user can be tracked by means of cross-matching when he/she uses the same biometrics across all applications and the service-providers collude with each other.

Therefore, the security of biometric template has been emerging increasingly and a lot of research has been done in this field. There are four properties that an ideal biometric template protection scheme should possess [2]:

(i) *Diversity:* The secure template must not be the same in two different applications; therefore, the user's privacy is ensured.
(ii) *Cancellability:* It should be straightforward to revoke a compromised template and reissue a new one based on the same biometric data.
(iii) *Security:* An original biometric template must be computationally hard to recover from the secure template. This property guarantees that an adversary does not have the ability to create a physical spoof of the biometric trait from a stolen template.
(iv) *Performance:* The biometric template protection scheme should not degrade the recognition performance of the biometric system.

In biometric template protection, fuzzy vault [3] is considered as a popular method. It binds a key with the biometric template and obtains the helper data for authentication. The template is hidden in the helper data. However, there are some problems that fuzzy vault encounters with. One of these stems from the reason that fuzzy vault cannot provide the diversity and revocability properties. In this paper, that shortcoming is made good by applying a feature transformation in a fuzzy vault scheme. Strictly speaking, the main idea for the marriage of fuzzy vault with feature transformation was introduced in a few recent proposals. Nevertheless, majority of which focus on two biometrics modalities, fingerprints [4–6], and iris [7, 8]. The transformations for face-based fuzzy vault scheme are rare and very complicated. Therefore, this paper will present a hybrid scheme of face-based fuzzy vault and feature transformation. Our proposed transformation is not only simpler but also suitable for many kinds of biometrics modalities. The face biometric templates are protected by being hidden in the set of chaff points generated by fuzzy vault scheme. Besides, these templates are able to be changed or revoked if the owners are suspicious about being tracked or stolen. Our experimental result will show that the newly proposed scheme guarantees the cancellability property with an acceptable error rate.

The structure of this paper is organised as follows. Section 2 provides a brief review of related works. The details of our proposed scheme are described in Section 3. Next, in Section 4, security issues are discussed. Following them, the evaluation is discussed in Section 5. At last, Section 6 provides the conclusion and future works.

2 Related works

Biometric template protection is an important issue in a biometric system. Biometric template of a person cannot be replaced or used again once it is compromised. In [9], Jain *et al.* presented two approaches to deal with this issue, including feature transformation and biometric cryptosystem.

In the biometric cryptosystem approach, a key is derived from the biometric template or bound with the biometric template. Both the biometric template and the key are then discarded, and only the public helper data is stored in the database. Though public helper data does not reveal any information about the biometrics and the key, it is very useful to regenerate the key from another biometric sample which is similarly enough to the biometric template. The research in [10] introduced the concepts of secure sketch and fuzzy extractor, a combination of artificial neural networks (ANN) and secure sketch [11] which are kinds of biometric cryptosystem approach. The fuzzy commitment scheme [12] and fuzzy vault [3] are two examples of the key binding approach.

In the feature transformation approach, the biometric templates are transformed before being stored in the database. The transformed templates are hard to be recovered to the original template even with some knowledge of transformation function. Then, the transformed templates are safe to store in the database.

2.1 Fuzzy vault

Fuzzy vault is one of the most popular biometrics template protection schemas that were proposed [3]. The idea is that Alice places a secret k in a fuzzy vault and locks it using a set A of elements from some public universe U. To unlock the vault and retrieve k, Bob must present a set B similar to A, i.e. B and A overlap substantially.

To construct a fuzzy vault, first, Alice selects a polynomial p of variable x that encodes k. Considering the elements of A as distinct x-coordinate values, she computes the polynomial projections for the elements of A. Then, she adds some randomly generated chaff points that do not lie on p. The final set includes real points which lie on p and chaff points. The number of chaff points is far greater than the number of real points. It will make the attacker hard to find the real points.

When Bob wants to unlock the vault and learns k (i.e. find p), he uses his unordered set B. If B overlaps with A substantially, he will be able to locate many points in the vault that lie on p. By using error-correction coding (e.g. Reed–Solomon), it is assumed that he can reconstruct p and discover k.

There are many researches following this scheme to construct the vault for fingerprint [4–6], iris [7, 8, 13], face [14], and some other biometric types.

However, several attacks against fuzzy vaults have been discovered [15]. These are: attacks via record multiplicity, stolen key inversion attack, and blended substitution attack. In a stolen key inversion attack, if an adversary somehow recovers the key embedded in the vault, he can decode the vault to obtain the biometric template. Since the vault contains a large number of chaff points, it is possible for an adversary to substitute a few points in the vault with his own biometric features. In this case, the system allows both the genuine user and the adversary to be successfully authenticated. This attack is called blended substitution. In record multiplicity attack, an adversary can access to two different vaults generated from the same biometric data (from two different applications). He can easily identify the genuine points in the two vaults and decode the vault. Thus, the fuzzy vault scheme does not provide cancellability properties. In this research, we proposed a hybrid scheme where biometric templates are first transformed based on a periodic function to guarantee the cancellable property.

2.2 Feature transformation

One of the original solutions for privacy-preserving biometric authentication is using transformation. The biometric is transformed by applying one transformation function and stored that transformed template instead of the original biometric. If that transformed template is compromised, the adversary will still not obtain the original biometric. Transformation functions are classified into two types: invertible (or salting) and non-invertible transformation.

Salting is a method in which the biometric features are transformed using a function defined by a user-specific key or password [9]. With the key, we can invert the transform template to the original one. Therefore, the key must be kept secret. Salting can be considered as two-factor authentication in which the users must present both secret key and biometric trait to the authentication system. In [16], Jin *et al.* generate a user-based random orthonormal $(n * n)$ matrix $\boldsymbol{A}$, where n is the size of biometric feature vectors. Then, the original template feature vector $\boldsymbol{x}$ is transformed to a secure domain using matrix product: $y = \boldsymbol{A}x$. The random orthonormal matrix is generated from a user-based key or token using Gram–Schmidt algorithm. The security in this scheme is relied on the user-specific random matrix which plays as a secret key. Another example of salting is using a user-based shuffling key to transform an iris code in [17]. User-based shuffling key which is generated based on users' key or password is an n-bit string. An iris code is also divided into n blocks. The transformation works as follows: beginning from the first to the last block, if the bit ith is 1 (or 0), block ith will be moved to the first (or last) place of the code.

In non-invertible transformation, the biometric template is transformed by a one-way transformation function. A one-way function F is 'easy to compute' (in polynomial time) but 'hard to invert' [9]. The function F can be public. Non-invertible transformation for fingerprint is proposed in [1]. The authors presented three methods to transform fingerprint. In the first method, the fingerprint image is divided into rectangular grid cells. A shifting map is defined as a transformation function. The minutiae in each cell are moved to a new position which is defined in a shifting map. There may be some minutiae to be shifted to the same cell. Thus, even if the shifting map is public, the attacker cannot infer that a minutia in the transformed template is belonged to which cell in the original template. This is the characteristic of non-invertible transformation. Similarly, in the second method, the fingerprint image is divided into sectors, and the minutiae are shifted among sectors which have the same or nearly the same radius. The third method considers not only the position but also the direction of the minutiae. Sutcu *et al.* in [18] proposed a secure authentication based on robust hashing. The idea is to embed each component of a feature vector into a Gaussian function. After that, a number of fake Gaussians are added to hide the true Gaussian.

To all of non-invertible transformation, the most challenge is that how to preserve the similarity of distances among transformed templates and among original templates. It means that two transformed templates must be similar if the two original templates are similar. This characteristic keeps the error rates of the transformed biometric systems similar to the generic biometric systems but the transformed biometric systems protect the templates from being compromised.

3 Proposed scheme

3.1 General scheme

Our proposed scheme can be applied for many kinds of biometric data whose feature is in vectors. In this paper, we use face feature and principal component analysis (PCA) [19] for feature extraction. Our general scheme includes two main phases: enrollment and authentication. In the enrollment phase, first, a biometric feature vector $\boldsymbol{X}$ is extracted from the users' face images. Then, periods of elements in $\boldsymbol{X}'$, called P, are collected. To protect P, P is hashed before storing in database. We also calculate an error-correction code (Reed–Solomon error correction [20]) to recover P in the authentication phase. Next, the sine transformation (or other periodic functions such as cosine) is

performed on the biometric feature vector. The randomly generated key is used to construct a polynomial function and its hashing is stored for matching purpose in the authentication phase. The transformed feature vector $\boldsymbol{Y}$ is projected by this polynomial function to generate a set of genuine points in fuzzy vault. To complete the fuzzy vault encoding step, a set of chaff points is also added into fuzzy vault set. After that, all these points are stored in fuzzy vault database.

In the authentication phase, an image of the user is captured. This image is also extracted in order to gain the user's biometric feature vector $\boldsymbol{X}'$ and feature period string P. After that, P' will be corrected by using the error-correction code. Then, the system compares the two hashed values of P and P'. If they are matched, the sine transformation is performed on the extracted feature $\boldsymbol{X}'$. Otherwise, the authentication is not successful. If the transformed feature vector $\boldsymbol{Y}'$ has substantial overlap with Y, the secret key will be correctly retrieved by the fuzzy vault decoding step. Afterwards, the recovered key is hashed in order to compare with the hashed version of the key generated in the enrollment phase. If the new key is matched, the user is authenticated; otherwise, it fails to authenticate the user. The overview of the system is illustrated in Fig. 1.

In summary, our approach is to combine a transformation function and a fuzzy vault system. The transformation function possesses two important properties. The first one is the non-invertible property to protect the biometric template, which will be analysed clearly in Section 4. The second one is to preserve the similarity of distances among transformed templates and among original templates. That means the two transformed templates must be similar if the two original templates are similar, and vice versa. This property enables the transformed templates to be applied in the fuzzy vault. First, we employ a transformation function on the feature vector $\boldsymbol{X}$ to get the transformed vector $\boldsymbol{Y}$. After that, we use $\boldsymbol{Y}$ as an input of the fuzzy vault.

To obtain a successful authentication, the user's feature vector $\boldsymbol{X}'$ must be similar to original feature vector $\boldsymbol{X}$. The similarity reservation property of the transform function ensures that if the two feature vectors are similar, the two transformed vectors will be similar. In this situation, fuzzy vault [3] is possible to recover exactly the original key ($K' = K$) if vector $\boldsymbol{Y}'$ is close enough (under a threshold) to vector $\boldsymbol{Y}$. Owing to security reason, we store the hash value of the key K instead of the key K itself. Therefore, we compare the hash values of K and K' to decide whether the authentication process is successful.

3.2 Feature extraction

A feature extraction technique is utilised to extract the face feature. Among many different feature extraction techniques, PCA and independent component analysis are popular ones for face recognition. In this paper, we choose PCA because of its significant outperformance on human face recognition task [21].

In the Eigenfaces method, the PCA is applied to a training set to find a set of standardised face ingredients, called eigenfaces. The training set is a large number of images depicting different human faces, including Γ_1, Γ_2, Γ_3, …, Γ_M images. We defined the average face of set as

$$\psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i \tag{1}$$

The difference between each face and the average is shown by vector

$$\phi_i = \Gamma_i - \psi \tag{2}$$

Then, the covariance matrix is calculated by

$$\boldsymbol{C} = \frac{1}{M}\sum_{i=1}^{M}\phi_i\phi_i^{\mathrm{T}} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} \tag{3}$$

where the matrix $\boldsymbol{A} = [\phi_1\phi_2\ldots\phi_M]$.

We can obtain M eigenvector and eigenvalue of covariance matrix $\boldsymbol{C}$. Then, we sort out R ($R \leq M$) largest eigenvectors by the

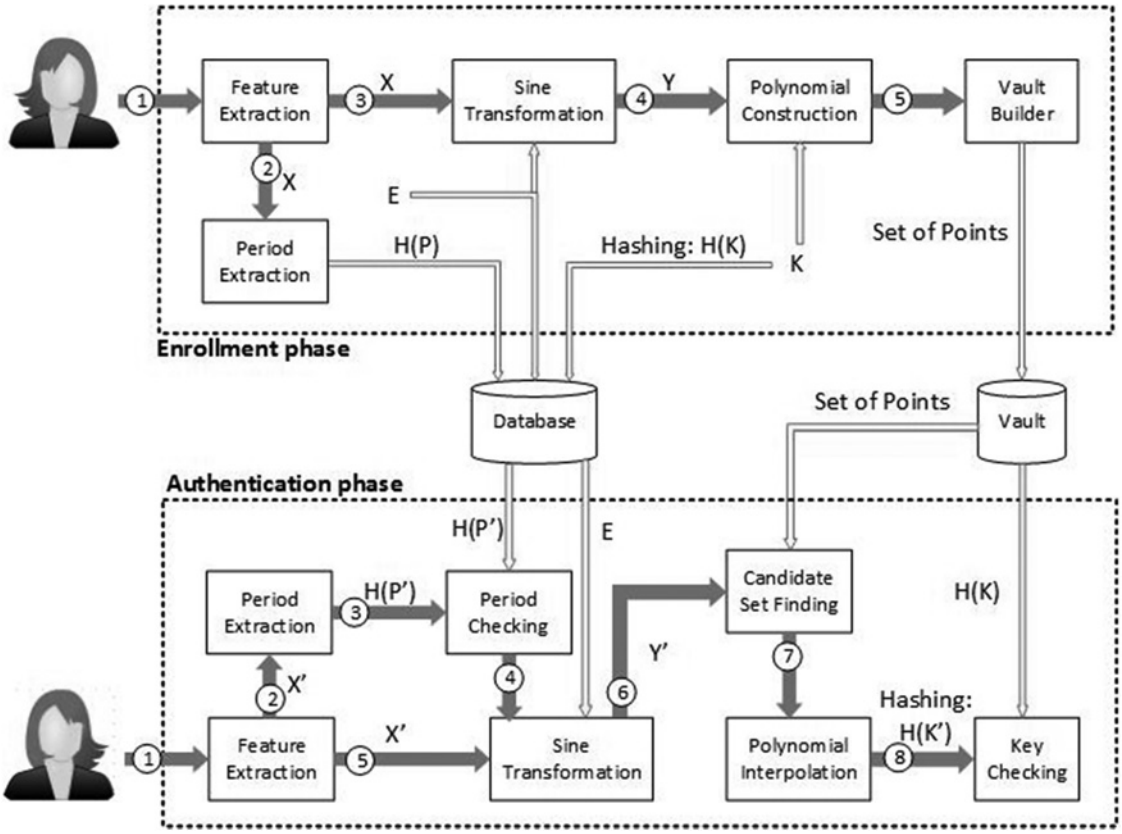

Fig. 1 *General architecture*

corresponding eigenvalues, denoted as: $U = [u_1 u_2 \ldots u_R]_{N^2 \times R}$. u_i is the eigenface; these eigenfaces are orthogonal to each other. The image of a user can be transformed to the R-dimensional face space by linear mapping

$$\Omega = U^{\mathrm{T}}(\Gamma - \psi) = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_R \end{bmatrix}_{R \times 1} \tag{4}$$

3.3 Sine transformation

Before going into details about our transformation, we introduce some notations which are used in this section:

$\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$: a biometric feature vector extracted from the face image of a user.

$\boldsymbol{Y} = \{y_1, y_2, \ldots, y_n\}$: a transformed vector after applied sine transformation on $\boldsymbol{X}$.

$\boldsymbol{E} = \{e_1, e_2, \ldots, e_n\}$: a random vector, in which e_i is chosen randomly between $[-1, 1]$.

$P = p_1, p_2, \ldots, p_n$: a string of period numbers of elements in $\boldsymbol{X}$.

3.3.1 Period extraction: Since we use the sine function which is periodic with the period of 2π, we need to know which period x_i belongs to. For each x_i in $\boldsymbol{X}$, we have

$$x_i = \alpha_i + p_i 2\pi \tag{5}$$

Therefore, we calculate p_i by the following equation

$$p_i = x_i \div 2\pi \tag{6}$$

p_i is the period number of x_i. After that, we convert p_i to binary. Then, we combine all period data p_i of feature vector $\boldsymbol{X}$ to get a binary string

$$P = p_1 \,||\, p_2 \,||\, \ldots \,||\, p_n \tag{7}$$

P is then hashed and stored in the database for checking later. Since the hashed value of P needs to be correct in the authenticate phase, we have to use the error-correction code (e.g. Reed–Solomon error-correction code [20]) for P so that we can check and recover the exact periods if they have some errors.

3.3.2 Sine transformation: When the feature extraction step completes, the feature vector $\boldsymbol{X}$ of a user will be transformed to vector $\boldsymbol{Y}$ by the function (8b)

For each element x_i in $\boldsymbol{X}$

$$\sin(x_i + y_i) = e_i \tag{8a}$$

Moreover, we can rewrite (8a) in the form of $y_i = f(x_i)$ in (8b)

$$y_i = \arcsin(e_i) - \alpha_i \tag{8b}$$

We have $\arcsin(e_i) \in [-\pi/2, \pi/2]$ and $\alpha_i \in [0, 2\pi]$. Then, the value of y_i is within $[-5\pi/2, \pi/2]$.

Owing to the periodic property of sine, for each value of x_i, we will find exactly one value y_i. However, given a value of y_i, we cannot derive an exactly x_i, because there are many values corresponding with that y_i. In other words, this transformation function is a non-invertible. We can also choose another periodic function has the same characteristic with sine function (such as cosine function). In this paper, we use the sine function to present our works.

The sine transformation is simply illustrated in Figs. 2 and 3. Note that, y_i can be positive or negative value.

One interesting point in our approach is that when using $\boldsymbol{Y}$ in the fuzzy vault, it also transforms an ordered data (vector $\boldsymbol{Y}$) to a disordered data (set of points in fuzzy vault). By this way, even if the fuzzy vault is compromised, it is hard for the attacker to discover the order of the genuine points to get the vector $\boldsymbol{Y}$.

3.4 Fuzzy vault encoding

The key K is 160 bit which is randomly generated by a random number generator. This key is used for polynomial construction in fuzzy vault scheme. At first, a nine-order polynomial, $f(x) = c_9 x^9 + c_8 x^8 + \cdots + c_1 x + c_0$, needs to be generated. The values of these coefficients are created by truncating the 160 bit K to ten non-overlapping 16 bit segments. Moreover then, each of them is mapped to the coefficients $c_9 - c_0$ in succession. For example, the first 16 bits is mapped to c_9, and so on. The order of the mapping should be preserved for encoding and decoding of the vault.

To create the genuine points, the polynomial $f(x)$ is evaluated on each of the transformed feature points x_i. As a result, the genuine set G consists of a set of pairs $\{x_i', f(x_i')\}_{i=1}^{M}$, where M is the dimension of the feature vector (also is the dimension of the transformed features). The next step we need to do is to generate the chaff points set $C = \{a_j, b_j\}_{j=1}^{N_C}$, where N_C is the number of the chaff points and

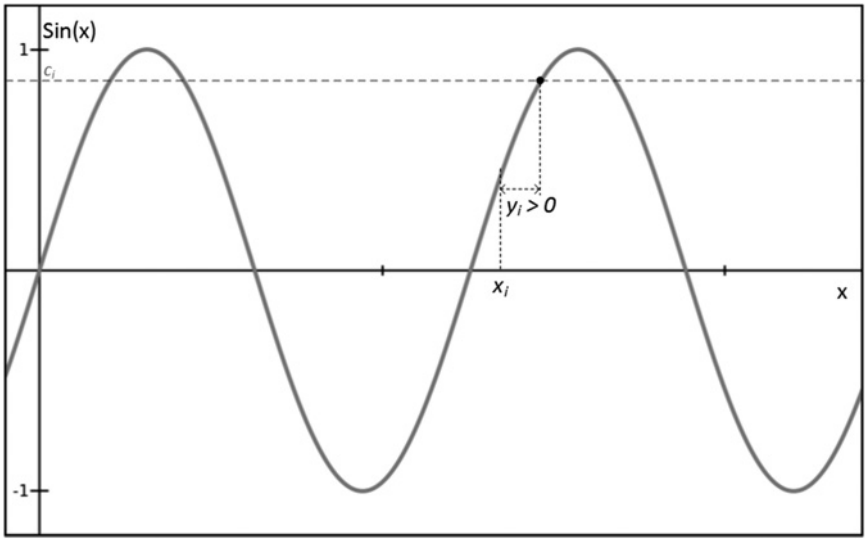

Fig. 2 *Sine transformation with $y_i > 0$*

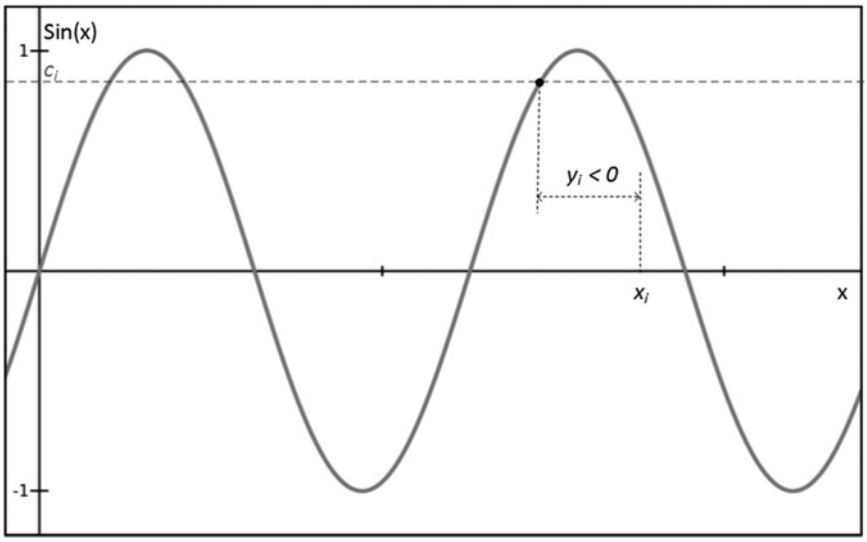

Fig. 3 *Sine transformation with $y_i < 0$*

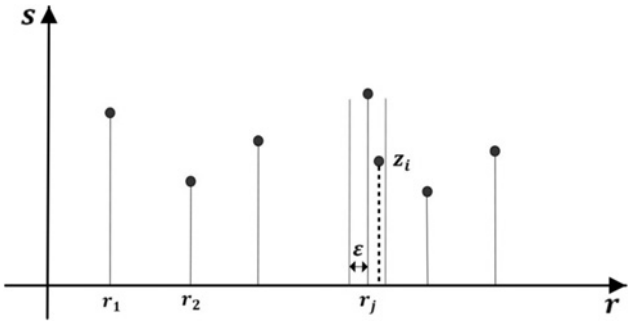

Fig. 4 *Fuzzy vault decoding*

$N_C \gg M$. The random generated set C needs to guarantee the following requirements

$$\begin{cases} \left|a_j - x_i'\right| > \Delta \ \forall i \ (\Delta \neq 0) \\ b_j \neq f\left(a_j\right) \ \forall j \end{cases} \tag{9}$$

The final vault V is obtained by taking the union of the two sets G and C

$$V = C \cup G = \left\{r_k, \ s_k\right\}_{k=1}^{M+N_C} \tag{10}$$

In our experiment, the size of feature vector is 40 ($M = 40$) and the number of chaff points is 200 ($N_C = 200$).

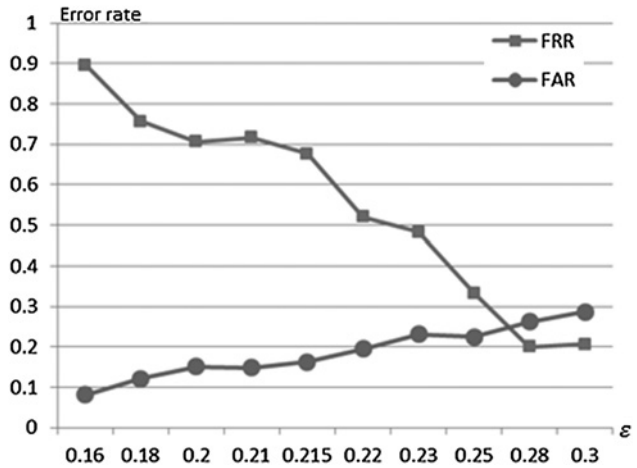

Fig. 5 *FAR and FRR of original scheme*

3.5 Fuzzy vault decoding

The fuzzy vault decoding (Fig. 4) is mainly based on the Lagrange polynomial interpolation. Assume that the transformed feature vector of a user is $Z = \{z_1, \ z_2, \ \ldots, \ z_M\}$. For each z_i, we find the x-coordinate (r_j) of one point in the vault set V in such a way that the r_j satisfy the following rules which is illustrated in Fig. 4:

$$\begin{cases} \left|z_i - r_j\right| \leq \varepsilon, \ \varepsilon \ll \Delta \\ \left|z_i - r_j\right| \leq \left|z_i - r_k\right|, \end{cases} \tag{11}$$

where ε is a designed threshold of the system, $k = \{1, \ 2, \ \ldots, \ M + N_C\}$ and $k \neq j$

As a result, the set of points $\{r_j, \ s_j\}_{j=1}^{L \leq M}$, whose r_j has just found, is the set of the candidate points. These points are ranked by the corresponding nearest distance between r_j and z_i. To recover the nine-order polynomial, the Lagrange interpolation technique [22] needs ten points. We choose the first I points ($10 \leq I$) of the ranked candidate set (the points have the highest possibility to be the real points) and then make the combinations of ten points from the I ranked candidate set (C_I^{10}). For each combination, we find one polynomial. Its coefficients are mapped back and concatenated in the same order as encoding phase in order to obtain a 160 bit key K'. To check whether the key K' is matched with the initial K or not, we hash K' and compare the result with the hashed versions of the keys in the database. The authentication is successful if and only if we can recover one key K' matched with K. It means that we do not have to compute all the combinations. Otherwise, if no matched key from all the combinations is found, the authentication is failed.

4 Security analysis

In [15], Scheirer and Boult discussed some techniques for cracking the fuzzy vault to obtain set of genuine points. These genuine points can be used to infer a user's feature set which is unique for

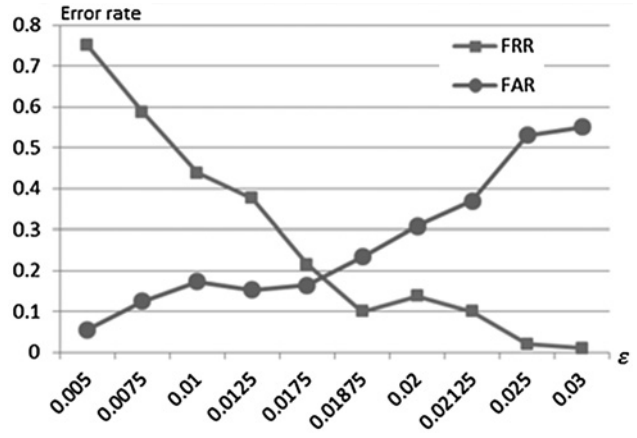

Fig. 6 *FAR and FRR of the proposed scheme with the capability of correcting up to four period errors*

each person and cannot be changed. In our approach, we transform user's feature set to a secure form by a non-invertible transformation function, i.e. the sine function. Furthermore, if the fuzzy is compromised, it is easy to change a new set of genuine points by replacing the random vector $\boldsymbol{E}$ in the transformation process. In other words, the attacker can crack the fuzzy vault to get the genuine points and the key but not the true biometric template. Then, the user can generate a new set of genuine points for the fuzzy vault. That is a reason why we said our approach supports the cancellablity properties for fuzzy vault.

A question we need to clarify is whether the transformation is non-invertible or how difficult an adversary can infer the original biometric template from the transformed one.

If someone gets the transformed values (namely $\boldsymbol{Y}$), he will try his best to find the correct original values (namely $\boldsymbol{X}$) based on the transformation function and public data. Our transformation is a many-to-one mapping. It means that one $\boldsymbol{X}$ is transformed to only one $\boldsymbol{Y}$ but one $\boldsymbol{Y}$ can correspond with many $\boldsymbol{X}$s. Let us consider (8a) or (8b), for each value c_i and y_i; it is hard for the adversary to infer which period p_i the value x_i belongs to.

In our implementation of face feature extraction, each element x_i of the feature vector $\boldsymbol{X}$ is in range of $[-200, 200]$ or about 2^6 periods of 2π. Therefore, an adversary needs to try $2^{6\times M}$ attacks, where M is the number of genuine points (or the size of the feature vector), in this paper, $M = 40$, to overcome checking period process in the authentication phase.

In case that the attacker can pass the checking period process and know the period vector P, he can infer the correct feature vector $\boldsymbol{X}$ only when he knows the transformed vector $\boldsymbol{Y}$. However, if the fuzzy is cracked, the attacker only knows the set of genuine points which are elements of vector $\boldsymbol{Y}$ but he does not know the order of these elements to make the vector $\boldsymbol{Y}$. To guess the correct order, he needs to test $M!$ cases.

In summary, a brute-force attack has to try roughly $2^{6\times 40} + 40!$ possibilities. This number is safe comparing with the power of computing nowadays. Moreover we can say, the sine transformation is non-invertible.

In terms of key deduction, the stored information which could be obtained by the intruder contains hash of period $H(P)$, random value E, hash of key $H(K)$, and the fuzzy vault data. It is obvious that $H(P)$ and E are not related to the key K at all, and the hash version $H(K)$ cannot be used to infer K. As a result, the only way to deduce the key without the original biometric is to utilise the fuzzy vault data. To crack the fuzzy vault, the attacker must detect the genuine points to process polynomial interpolation. In our scheme, a nine-order polynomial is used, and in the experiment, the number of genuine points and chaff points inside the vault is $(40 + 200 = 240)$ points. There are total C_{240}^{10} combinations with ten elements. The number of combinations can unlock the vault is C_{40}^{10}. Therefore, a brute-force attack of polynomial interpolation needs to calculate $(C_{240}^{10}/C_{40}^{10}) \sim 10^{10}$ possibilities which were demonstrated to be safe based on several researches [5, 23]. Furthermore, even in the worst case when the fuzzy vault is successfully cracked and the secret key is explored, the original biometric template still keeps secure in our proposed scheme. The system could easily generate a new key and combine it with the biometric template to construct a new fuzzy vault to replace for the cracked one.

5 Evaluation

The newly and original (e.g. without the periodic transformation) scheme is tested under the Face94 database [24]. In the training procedure, we use 100 images of 50 people, i.e. two images per person, to construct 40 eigenfaces. Then, we test the scheme with 152 people including 50 people participated in the training process and 102 new people. Each person has five images in which one is used to create the vault and four are used to unlock the vault. We measure the performance of this scheme include: false acceptance rate (FAR) and false rejection rate (FRR) [25]:

- The FAR defines the probability that the system incorrectly matches the input pattern to a non-matching template in the database. It measures the per cent of invalid inputs which are incorrectly accepted.
- By analogy, the FRR defines the probability that the system fails to detect a match between the input pattern and a matching template in the database. It measures the per cent of valid inputs which are incorrectly rejected.

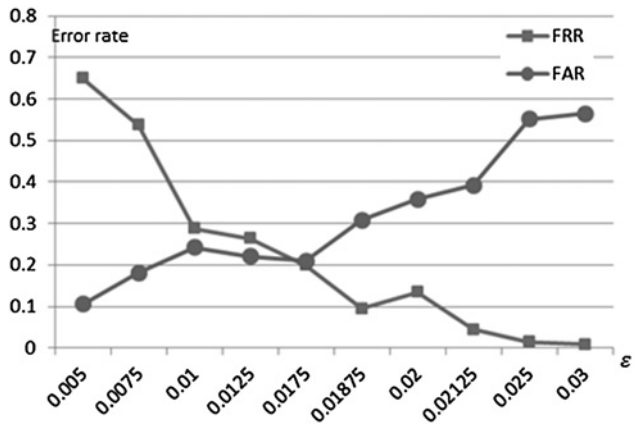

Fig. 7 *FAR and FRR of the proposed scheme with the capability of correcting up to five period errors*

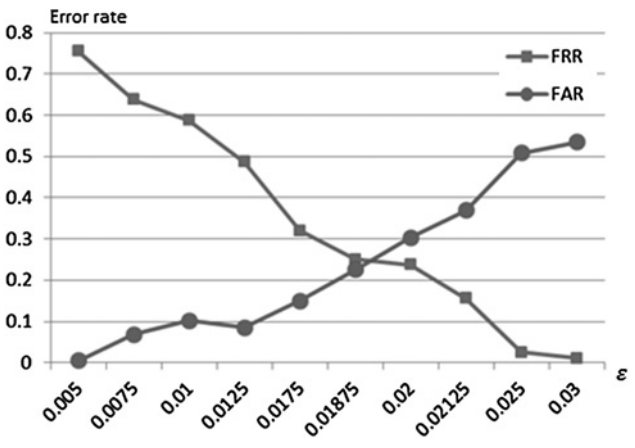

Fig. 8 *FAR and FRR of the proposed scheme with the capability of correcting up to three period errors*

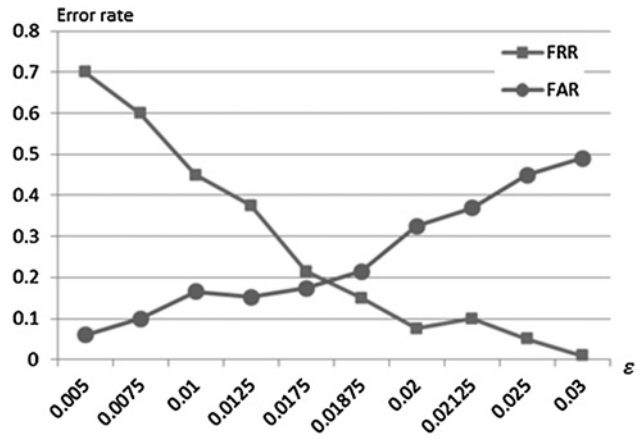

Fig. 9 *FAR and FRR of the proposed scheme with cosine function and capability of correcting up to four period errors*

The results of original schema which is without the transformation are shown in Fig. 5; meanwhile, the FAR and FRR of cancellable fuzzy vault is demonstrated in Figs. 6–8 respected to different correcting period errors capability of Reed–Solomon code. The vertical axis is the error rate which is range [0, 1] and the horizontal axis is the epsilon (ε) as defined in Section 3.5. Epsilon is the minimum distance among points in the vault.

As can be seen from the graphs, without the sine transformation, we can get the acceptable error rates at $\varepsilon = 0.25$. With this value, the error rates are: FRR $\simeq$ FAR $\simeq 0.23$.

Otherwise, using the new schema, we can get the better error rates FRR $\simeq$ FAR < 0.2 at $\varepsilon = 0.01875$ with the capability of correcting up to four period errors. Increasing the error tolerance, the FAR is also increased, the FRR, on the other hand, is decreased, and vice versa.

We also evaluate our solution with another periodic function, here is cosine and the results are illustrated in Fig. 9. It is clear that the FAR and FRR are a little equivalent with the sine function.

6 Conclusion and future works

In this research, we proposed a cancellable fuzzy vault scheme which performs a periodic transformation on the biometric template and then let it as an input to the fuzzy vault. By this way, we can strengthen the fuzzy vault with the cancellability property. Our transformation function is non-invertible because the transformation function, i.e. sine function, is periodic with the period of 2π. With the knowledge of the sine function, we cannot infer the true value of x. The results of the evaluation confirm the effective and the practical properties of our scheme to protect biometric template.

Our two future works are to reduce the error rates (FAR, FRR) so that it can be used in practice and to find a proper way to add chaff points to the vault. For the first task, we need more researches on the range of each component of the feature vectors so that we can adjust our parameters: namely, the minimum distance among points in the vault, the maximum range of x in the vault, and the most suitable for a specified biometric feature. If we can decide these parameters more precisely, we can archive lower error rates in our scheme. Another approach is that we can apply multimodal biometrics schema to improve performance of the system [26, 27]. For the second task, in this paper, the chaff points are added randomly without the consideration that the transform values (with respect to x value in the vault) of true points have a maximum distance of a predefined number of p_i to their nearest true points. This property can be exploited to limit an amount of possible cases in a brute-force attack. Therefore, we need to find a better solution to add chaff points to the vault in our scheme.

7 Acknowledgments

This research is funded by the Vietnam National University – Ho Chi Minh City (VNU-HCM) under grant number B2013-20-02. We also want to show a great appreciation to each member of the Data Security Applied Research (D-STAR) Lab (http://www.dstar.edu.vn) for their enthusiastic supports and helpful advices during the time we have carried out this research.

8 References

1 Ratha, N., Chikkerur, S., Connell, J., *et al.*: 'Privacy enhancements for inexact biometric templates' (Security with Noisy Data, Springer, London, 2007), pp. 153–168
2 Maio, D., Jain, A.K.: 'Handbook of fingerprint recognition' (Springer, 2009)
3 Juels, A., Sudan, M.: 'A fuzzy vault scheme', *Des. Codes Cryptogr.*, 2006, **38**, (2), pp. 237–257
4 Clancy, T.C., Kiyavash, N., Lin, D.J.: 'Secure smartcard-based fingerprint authentication'. Proc. of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications, 2003, pp. 45–52
5 Nandakumar, K., Jain, A.K., Pankanti, S.: 'Fingerprint-based fuzzy vault: implementation and performance', *IEEE Trans. Inf. Forensics Sec.*, 2007, **2**, (4), pp. 744–757
6 Uludag, U., Jain, A.K.: 'Fuzzy fingerprint vault'. Proc. of Workshop Biometrics: Challenges Arising from Theory to Practice, pp. 13–16
7 Lee, Y.J., Bae, K., Lee, S.J., *et al.* 'Biometric key binding: fuzzy vault based on iris images'. Advances in Biometrics, Berlin Heidelberg, 2007, pp. 800–808
8 Hao, F., Anderson, R., Daugman, J.: 'Combining crypto with biometrics effectively', *IEEE Trans. Comput.*, 2006, **55**, (9), pp. 1081–1088
9 Jain, A.K., Nandakumar, K., Nagar, A.: 'Biometric template security', *EURASIP J. Adv. Signal Process.*, 2008, **113**, Article No. 113, http://dl.acm.org/citation.cfm?id=1387883
10 Dodis, Y., Reyzin, L., Smith, A.: 'Fuzzy extractors: how to generate strong keys from biometrics and other noisy data'. Advances in cryptology-EUROCRYPT, Berlin Heidelberg, 2004, pp. 523–540
11 Huỳnh, V.Q.P., Thai, T.T.T., Dang, T.K., *et al.*: 'A combination of ANN and secure sketch for generating strong biometric key', *J. Sci. Technol., Vietnamese Acad. Sci. Technol.*, 2013, **51**, (4B), pp. 203–212
12 Juels, A., Wattenberg, M.: 'A fuzzy commitment scheme'. Proc. Sixth ACM Conf. on Computer and Communications Security, 1999, pp. 28–36
13 Vo, T.T.L., Dang, T.K., Josef, K.: 'A hash-index method for securing fuzzy vaults'. Proc. 11th Int. Conf. on Trust, Privacy & Security in Digital Business, Munich, Germany, 2014 (*LNCS*)
14 Wu, Y., Qiu, B.: 'Transforming a pattern identifier into biometric key generators'. IEEE Int. Conf. on Multimedia and Expo (ICME), 2010, pp. 78–82
15 Scheirer, W.J., Boult, T.E.: 'Cracking fuzzy vaults and biometric encryption'. Proc. Biometrics Symp., Baltimore, MD, USA, September 2007
16 Jin, A.T.B., Ling, D.N.C., Goh, A.: 'Bio hashing: two factor authentication featuring fingerprint data and tokenised random number', *Pattern Recognit.*, 2004, **37**, (11), pp. 2245–2255
17 Sanjay, K., Camara, D., Krichen, E., *et al.*: 'Three factor scheme for biometric-based cryptographic key regeneration using iris'. IEEE Biometrics Symp., 2008, pp. 59–64, http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4655523
18 Sutcu, Y., Sencar, H.T., Memon, N.: 'A secure biometric authentication scheme based on robust hashing'. Proc. Seventh Workshop on Multimedia and Security, 2005, pp. 111–116
19 Turk, M., Pentland, A.: 'Eigenfaces for recognition', *Cogn. Neurosci.*, 1991, **3**, (1), pp. 71–86
20 Reed–Solomon code from Wikipedia. Available at http://www.en.wikipedia.org/wiki/Reed%E2%80%93Solomon_error_correction, accessed 25 May 2015
21 Baek, K., Draper, B.A., Beveridge, J.R. PCA vs. ICA *et al.*: 'A comparison on the FERET data set'. Proc. Fourth Int. Conf. on Computer Vision (ICCV'02), 2002, pp. 824–827
22 Wolfram MathWorld, Lagrange Interpolating Polynomial. Available at http://www.mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html, accessed 25 May 2015
23 Umut, U., Pankanti, S., Jain, A.K.: 'Fuzzy vault for fingerprints'. Audio-and Video-Based Biometric Person Authentication, Berlin Heidelberg, 2005
24 Libor Spacek's Faces94 database. Available at http://cswww.essex.ac.uk/mv/allfaces/faces94.html, accessed 26 May 2015
25 Biometrics from Wikipedia. Available at http://www.en.wikipedia.org/wiki/Biometrics, accessed 26 May 2015
26 Nguyen, V.N., Nguyen, V.Q., Nguyen, M.N.B., *et al.*: 'Fuzzy logic weight estimation in biometric-enabled co-authentication systems'. Proc. of Information and Communication Technology, Indonesia, 2014 (*LNCS*), pp. 365–374
27 Militello, C., Conti, V., Sorbello, F., *et al.*: 'A fast fusion technique for fingerprint and iris spatial descriptors in multimodal biometric systems', *Comput. Syst. Sci. Eng.*, 2014, **29**, (3), pp. 205–217