IET Biometrics

Research Article

ISSN 2047-4938
Received on 27th April 2015
Revised on 15th July 2015
Accepted on 26th October 2015
doi: 10.1049/iet-bmt.2015.0023
www.ietdl.org

Chaff point generation mechanism for improving fuzzy vault security

Tran Khanh Dang[1,2] ✉, *Minh Tan Nguyen*[2], *Quang Hai Truong*[2]

[1]*Faculty of Computer Science and Engineering, FAW Institute, JKU Linz, Linz, Austria*
[2]*Faculty of Computer Science and Engineering, HCMC University of Technology, Ho Chi Minh, Vietnam*
✉ *E-mail: khanh@hcmut.edu.vn*

Abstract: A combination of cryptographic and biometric systems, by performing specific binding technique on cryptographic key and biometric template, the fuzzy vault framework enhances the security level of current biometric cryptographic systems in terms of hiding secret key and protecting the template. Although the original scheme suggests the use of error-correction techniques (e.g. the Reed–Solomon code) to reconstruct the original polynomial, recent implementations do not share the same point of view. Instead, cyclic redundant code (CRC) is applied to identify the genuine polynomial from a set of candidates due to its simplicity. Within the scope of this study, the authors address a significant flaw of current CRC-based fuzzy vault schemes, which allows the potential of successful blend substitution attack. To overcome this problem, an integration of two novel modules into general fuzzy vault scheme, namely chaff point generator and verifier, is proposed. The new modules are designed to be integrated easily into the existing systems as well as simple to enhance. The proposed scheme can detect any modification in vault and, as a result, eliminate the blend substitution attack to improve general security. Moreover, the experimental results of this study with real-world datasets show an increase in genuine acceptance rates.

1 Introduction

Nowadays, although biometrics is widely applied in authentication systems, there are still several concerns regarding their practical applications. Besides the hardware limitation as described in [1], most of the worries are related to the lack of robust security schemes to protect efficiently biometric template and handling errors caused by noises. Reviewing those concerns leads us to two main solutions: biometric cryptosystems (BCSs) and cancellable biometrics [2]. Towards protecting biometric template, BCSs try to integrate biometrics with cryptography. Specifically, a cryptographic key will be bound directly with the biometric template in key binding schemes or generated from biometric information in key generator schemes. Fuzzy vault, one of the most popular key binding techniques, is the focus of this paper. Fuzzy vault, introduced first by Juels and Sudan [3], is well designed for an unordered input set. It can also handle noisy data by applying an error-correction code. The scheme includes two main phases: enrolment (encoding) and authentication (decoding). Fuzzy vault scheme is based on the complexity of the polynomial reconstruction problem to achieve high security levels. Recent researches on fuzzy vault [4–7], as opposed to its first practical implementation by Clancy *et al.* [8], propose another approach using a combination of cyclic redundant code (CRC) and Lagrange interpolation for decoding (Fig. 1). CRC-based decoder is well designed and is effortless to implement. Moreover, within the new approach, authentication result will be evaluated from a list of sorted candidates, not just based on one specific error correction result. However, significant performance and security problems as reported in [9] are left behind. CRC-based fuzzy vault is confirmed as one of the main approaches for implementing fuzzy vault in [10]. Within this research, Benhammadi and Bey proposed a password hardened fuzzy vault based on an improvement of fingerprint feature representation with novel minutiae pairwise structure. The proposed approach overcomes limitations of traditional prealignment fingerprint matching algorithm and helps prevent statistical analysis attack. Non-random chaff point generator has been recently considered by Khalil-Hani *et al.* [11]. Nevertheless, the work only focuses on building a new chaff generation algorithm which is computationally fast by replacing Euclidian distance calculation and adding chaff points with the boundary matrix. Generally, further work on improving CRC-based fuzzy vault still needs to be addressed in order to prevent traditional attacks [12–15], especially with blend substitution attack.

This article proposes a novel method to solve current CRC-based fuzzy vault problems. A modified chaff point generator and verifier are introduced. Continuous hashing and linear projection will be used to structurally generate chaff points at the enrolment phase. As a consequence, at the authentication phase, the same chaff point sets will be regenerated in assistance for giving final decisions. Within the proposed scheme, chaff points will be treated as a signature for a combination of biometric template and secure key. Any modification on decoded vault will be detected at authentication phase, preventing the blend substitution attack.

Turning to the main structure of this paper, after a brief review of the technical approaches of fingerprint-based fuzzy vault implementations, a significant security flaw of CRC-based fuzzy vault scheme will be described in Section 2. Assuming that the attacker has read/write permissions to the template database, by using blend substitution attack, he/she can exploit the above flaw to gain access to our system without affecting the current user, thus avoiding detection. Within Section 3, the proposed approach to overcome that flaw as well as to totally enhance the security level of the whole system will be introduced. Two additional modules and major change in chaff points generating process will be described in detail. This section also contains security analysis of the proposed scheme. Experimental results will be reported in Section 4. Finally, Section 5 concludes the paper.

2 Related work

2.1 Attack against fuzzy vault systems

(i) *Brute force attack:* Within the fuzzy vault schema, naive brute force supposes that an attacker only has knowledge of vault V. An

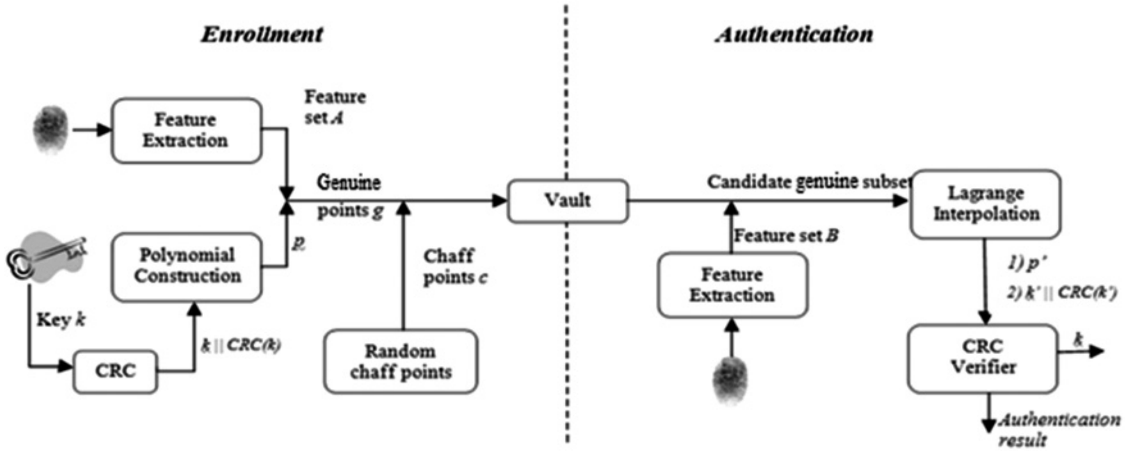

Fig. 1 *Original CRC-based fuzzy vault*

attacker will try to apply the decoding algorithm (RS or Lagrange decoder) on a subset of vault points to find out the one that can regenerate the original secret key. By implementing a brute force attack model on fuzzy vault implementation, Mihailescu in [12] concludes that the attack's complexity grows when increasing the polynomial degree, the number of chaff points per vault or reducing the number of genuine points. On the other hand, choosing insufficient combination of configurations can lead to significant decrease of complexity.

(ii) *Statistical analysis attack:* On the basis of the definition of free area, the available space to put in new points without violating the minimum distance restriction between points in vault, Chang *et al.* [13] provided a novel method to narrow the size of candidates for genuine point when applying brute force attack. From their observation, points that are inserted later into a vault are more likely to have smaller free space. As a result, when receiving a vault, the attacker first tries to compute free area of all points. Brute force attack, as the later step, first chooses points that have large free area as input.

(iii) *Collusion attack:* Collusion attack on fuzzy vaults considers the case when the attacker gains access to multiple vaults of the same biometric template and/or same secret case. By comparing points across vaults, the attacker can identify and remove chaff points, thus reducing the number of spurious polynomials. Hoi Ting Poon, in [14], examines in detail each collusion attack scenario and analyses the security loss in practical implementation. To prevent collusion attack, Nandakumar *et al.* [7] introduced a hybrid fuzzy vault schema using password for biometric transformation. However, there is security concern regarding this approach reported in [16]. Another considerable approach is introduced by Tams in [17]. The implementation inherits the work of Tams *et al.* [18], then improves with Guruswami–Sudan algorithm-based decoder on multiple fingerprints to provide a notable result in preventing collusion attack.

(iv) *Key inversion attack:* Key inversion attack addresses attacking scenarios when the secret key is compromised. Knowing the secret key, the attacker can regenerate the original polynomial, and thus identify genuine points. Owing to the ability to directly disclose the user's real biometric template, the key inversion attack is a critical attack on fuzzy vault systems.

Table 1 Summary of notations

Notation	Description
V	vault with g genuine points and c chaff points
k	cryptographic secret key to generate V
p	appropriate polynomial
n	degree of p
P	probability of successfully generating a secret key within naive brute force attack
T	number of subset of V that contains exactly $(n+1)$ points
G	number of subset of V that contains exactly $(n+1)$ points and all are genuine points
l	bit length of key k
t	default bit length of each coefficient

a. *Substitution attack:* By injecting fake points into the vault, the attacker can gain access to the system without affecting the current user's authentication process. By applying this attack, the attacker can generate a backdoor to continuously penetrate our authentication mechanism without being detected. In [15], blend substitution attack via biometric systems is reported and analysed with significant notice.

2.2 CRC-based fuzzy vault's problems

Poon and Miri, in [9], pointed out two main security problems related to CRC-based fuzzy vault scheme.

First, the current CRC implementation results in a checking code with short lengths such as 16 or 8 bits; thus the probability of CRC collision is considerable. The article's implementation shows that, within 16 bits CRC, when working with a large number of interpolations to decode, the result keys can have up to 50% CRC collision. Within a CRC-based fuzzy vault system, a CRC collision can lead to accepting illegal result key, thus increasing the false acceptance rate (FAR).

Second, adding CRC to the secret key leads to increasing the length of the input for the polynomial generation module. With the longer input key, a higher degree of polynomial is needed because the number of bits for each coefficient is limited. As a result, to assure sufficient security level, encoding and decoding algorithms have to work with a higher degree polynomial. Thus, more genuine points are needed to successfully decode the same vault, leading to increasing false reject rate (FRR). For details, see Table 1.

To construct the polynomial, key k is split into parts of a coefficient. From the above assumption, a polynomial with below degree will be generated

$$n = \left\lceil \frac{l}{t} \right\rceil - 1$$

As stated above, $n + 1 = \lceil l/t \rceil$ genuine points will be needed to successfully reconstruct this polynomial. As a consequence:

Formula 1: The probability of successfully generating a genuine secret key within naive brute force attack based on bit length of key is

$$P_2 = \frac{G}{T} = \frac{\binom{g}{(l/t)}}{\binom{g+c}{(l/t)}}$$

In case of adding i bits CRC to key k, *formula 1* will become *formula 2* as follows:

Formula 2: The probability of successfully generating a genuine secret key within naive brute force attack based on bit length of key with CRC is

$$P_3 = \frac{G}{T} = \frac{\binom{g}{(l+i/t)}}{\binom{g+c}{(l+i/t)}}$$

Implementations of fuzzy vault mostly need to satisfy two conditions:

- $n < g/2$: to control FRR, because a successful login requires at least $(n+1)$ genuine points matched.
- $c \gg g$: for hiding genuine points, a very large number of chaff points in comparison with the number of genuine points will be added.

Considering the above conditions together with *formulae* 1 and 2, when adding CRC, both G and T are reduced (because $n < g/2$). However, because $c \gg g$, G will be reduced faster in comparison with T. As a consequence, within the same parameters, $(P_2/P_3) > 1$ and, of course, $\mathrm{FRR}_3 > \mathrm{FRR}_2$.

2.3 Blend substitution attack on CRC-based fuzzy vault

The number of points per vault of a fuzzy vault system normally is a predefined parameter of encoding/decoding algorithms. Therefore, any modification that leads to significant changes of this number on specific vault will be easily detected by the system. Within blend substitution attack, the main target is to bypass the authentication system without being detected. To achieve this goal, the attacker has to apply fake injection into the vault without modifying too many points while keeping the current authentication behaviour of genuine users.

To attack CRC-based fuzzy vault systems, the attacker, first, generates his/her fake key and the corresponding CRC. Applying fuzzy vault encoding algorithm on fake key and biometric, fake genuine points with sufficient information to bypass the authentication system will be generated and injected into the current vault after randomly deleting some vault's points. Consider vault V with g being the genuine points, c the chaff points and encoding polynomials with degree n as above; for a successful regenerate key from that vault, a genuine user needs to have at least $(n+1)$ matching genuine points from their biometric information. Consequently, the attacker will need to delete exactly $(n+1)$ points from vault V to inject his own fake points. Fig. 2 demonstrates the status of the vault before and after being attacked by blend substitutions. After successful attack attempt, the vault contains at least two different lines that represent two polynomials (of both the attacker and the legitimate user). A legitimate user can still login to the system as usual and so does the attacker. Actually, reducing the number of original vault points can lead to increasing the FRR for the legitimate user. However, it is difficult for the end user to realise the difference in FRR before and after because of the distraction from the original FRR caused by noise. Meanwhile, the attacker will always perform successful login with just a small number of fake genuine points.

Considering the worst case when all the deleted points are genuine points can help to clarify the situation. Within the legitimate user's side, the number of subsets of vault which have $(n+1)$ points and all of them are genuine points will be

$$G = \binom{g-(n+1)}{n+1}$$

As a result, *formula 3* is constructed:

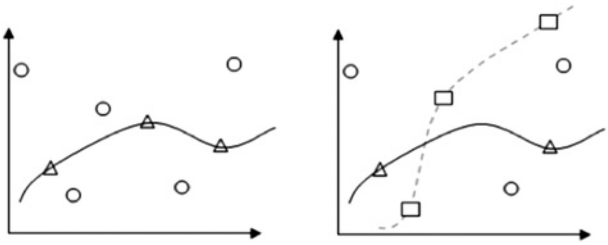

Fig. 2 *Blend substitution attack on CRC-based fuzzy vault systems. Left: original vault with only one genuine polynomial; right: blend attacked vault with genuine polynomial together with fake polynomial*

Formula 3: The probability of successfully generating a genuine secret key within the naive brute force attack after blend substitution attack is

$$P_4 = \frac{G}{T} = \frac{\binom{g-(n+1)}{n+1}}{\binom{g+c}{n+1}}$$

As can be seen from *formula 3*, there is a reduction of successful login probability for the legitimate user which leads to increasing FRR. However, recalling that $n < g/2$ and a successful reconstruction of a polynomial with n degrees just requires the legitimate user to provide $(n+1)$ genuine points, the change in FRR causes a very limited effect on the user's authentication process. Furthermore, the probability of continuously deleting $(n+1)$ genuine points is extremely small as

$$P_{\mathrm{D}} = \prod_{i=0}^{n} \frac{g-i}{g+c-i}$$

Let us consider a practical case, suppose that $n = 10$, $g = 30$ and $c = 300$

$$P_{\mathrm{D}} = \prod_{i=0}^{n} \frac{g-i}{g+c-i} = \prod_{i=0}^{10} \frac{30-i}{330-i} \simeq 5.1 \times 10^{-13}$$

As can be seen from the above calculation, within the fuzzy vault system, the attack of blend substitution is hard to be detected within a practical environment.

3 New fuzzy vault framework

In addition to traditional fingerprint-fuzzy vault modules, we modify and integrate two specific modules: *chaff point generator* and *chaff point verifier*. New modules are used to replace for current CRC mechanism of CRC-based fuzzy vault systems:

(i) *Chaff point generator module* accepts a feature set extracted from the biometric template and secret key as an input, then outputs the corresponding set of chaff points.
(ii) *Chaff point verifier module* applies the same algorithm with a chaff point generator to verify vault's chaff points after successfully regenerating the secret key.

Within the new scheme, chaff points can be considered as a *virtual 'CRC'* for genuine points and secret key. By checking chaff points at the authentication phase, we can not only verify the secret key but also detect any unauthorised modifications to the vault and as a result, efficiently prevents blend substitution attack.

Algorithms to implement new modules can be varied; however, they have to satisfy the basic conditions:

- On the basis of a combination of biometric information and secret key for chaff point generation: to prevent collusion attack when knowing the part of biometric template or the secret key.
- Be evaluated with traditional BCSs' attacks such as brute force, collusion, substitution etc.
- Restrict entropy loss: additional implementations mostly lead to additional public data and/or providing new potential attacking methods. New algorithm needs to eliminate all those kinds of entropy loss.

Integration of new modules can be found in Fig. 3; verification strategies, at authentication phase, will be based directly on the implementation of the generator module but can be enhanced to reduce the total working time.

First of all, we take a look into the basic fuzzy vault algorithm. Assume that genuine biometric information is extracted and quantised to map to a finite field with q elements, denoted by $\mathbb{F}_q$. A mono polynomial with degree $k-1$ over $\mathbb{F}_q$ can be denoted as $P(X) = \sum_{i=0}^{k-1} p_i X^i$, with k coefficients $p_0, \dots, p_{k-1} \in \mathbb{F}_q$. Fuzzy vault maps a subset $\boldsymbol{g} \subseteq \mathbb{F}_q$ by a polynomial $P(X) = \sum_{i=0}^{n-1} k_i X^i$, formed by a secret key $k = (k_0, \dots, k_{n-1}) \in \mathbb{F}_q^n$, into a genuine set $G = \{(g_i, P(g_i)) | g_i \in \boldsymbol{g}\}$. Then, another set of random points $C = \{(c_i, P'_i) | c_i \in \mathbb{F}_q \backslash (g), P'_i \in \mathbb{F}_q \backslash (P(c_i))\}$ is generated as a noisy factor. Finally, vault V is the result of $G \cup C$. Polynomial $P(X)$ can be regenerated by applying the polynomial construction on a subset $\boldsymbol{g}' \subseteq \mathbb{F}_q$ if and only if $\boldsymbol{g}'$ contains at least k roots of $P(X)$.

From these assumptions, we introduce a novel method to implement the new proposed scheme by applying continuous hashing and linear projection.

At the enrolment phase, instead of being chosen randomly, the set of chaff points C are generated systematically by the following method:

- *Step 1:* Genuine set $G = \{(g_i, P(g_i)) | g_i \in \boldsymbol{g}\}$ is sorted in ascending order by g_i.
- *Step 2:* An original input is generated from a combination of $\boldsymbol{g}$ and k as
  ○ $h_0 = (g_0, \dots, g_{m-1}, k_0, \dots, k_{n-1}) \in \mathbb{F}_q^{n+m}$.
- *Step 3:* A predefined hashing function
  ○ $f : \mathbb{F}_q^{n+m} \to \mathbb{F}_q^{n+m}$

  is applied on h_0 results in a new hashing input value

○ $h_1 = (p_0, \dots, p_{n+m-1}) \in \mathbb{F}_q^{n+m}$

- *Step 4:* Forming a line by using p_0 and p_{n+m-1} as
  ○ $P_{\text{line}} = p_0 X + p_{n+m-1}$
- *Step 5:* Following the predefined order, one genuine point $(g_i, P(g_i)) \in \mathbb{F}_q^2$ from G is chosen. This point is projected directly onto P_{line} (step 4) to get a candidate chaff point $(c_i, c_j) \in \mathbb{F}_q^2$. The new candidate chaff point will be added to C if and only if it satisfies all preconditions of a valid chaff point. Note that through the whole algorithm, this step will be repeated many times until getting sufficient number of chaff points. Therefore, G will be traversed several cycles. Within each cycle, points are chosen one by one following the sorted order (ascending order by default).
- *Step 6:* In case of getting sufficient number of chaff points or reaching looping limited, the algorithm will stop. Otherwise, h_0 are recomputed by taking half of h_1 and original k as
  ○ $h_0 = (p_0, \dots, p_{m-1}, k_0, \dots, k_{n-1}) \in \mathbb{F}_q^{n+m}$

Then, we return to step 3 again to get new chaff points. Matrix of projecting a point $(g_i, P(g_i))$ onto a line $P_{\text{line}} = p_0 X + p_{n+m-1}$ within a two-dimensional (2D) space can be generated by using the orthogonal matrix as follows:

Formula 4: Calculate raw chaff from project genuine point $(g_i, P(g_i))$ onto line $P_{\text{line}} = p_0 X + p_{n+m-1}$

$$\begin{bmatrix} x_C \\ y_C \end{bmatrix} = \left(\left(\frac{1}{1+p_0^2} \right) \begin{bmatrix} 1 & p_0 \\ p_0 & p_0^2 \end{bmatrix} \right) \left(\begin{bmatrix} g_i \\ P(g_i) \end{bmatrix} - \begin{bmatrix} 0 \\ p_{n+m-1} \end{bmatrix} \right) + \begin{bmatrix} 0 \\ p_{n+m-1} \end{bmatrix}$$

To make new chaff points uniform with the others in the same vault, modulo function to maximum value of each genuine point's coordinate is applied. The candidate chaff point (c_i, P'_i) will be considered as real chaff points if it satisfies all validation rules:

- X-coordinate rule: $c_i \in \mathbb{F}_q \backslash (g)$.
- Y-coordinate rule: $P'_i \in \mathbb{F}_q \backslash (P(c_i))$.
- Minimum distance rule (α minimum distance constant)

$$d = \sqrt{(c_i - g_i)^2 + (P'_i - P(g_i))^2} > \alpha \, \forall g_i \in \boldsymbol{g}$$

New conditions or transformations based on specific implementations could be applied easily within this phase. The hashing result will, continuously, be concatenated with the secret key and put in hashing module again to get the new chaff points, following the same procedure. (see Fig. 4) At the authentication phase, after applying the filtering algorithm, a set of candidate points are chosen. To obtain the genuine key, a verifier algorithm could be used as below:

Step 1: From the set of candidate points $D = \{(d_i, P'_i)\} \in \mathbb{F}_q^2$, select one subset $D_1 \subseteq D$ (which has not been chosen yet) with at least t points [when dealing with a $(t-1)$-degree polynomial].

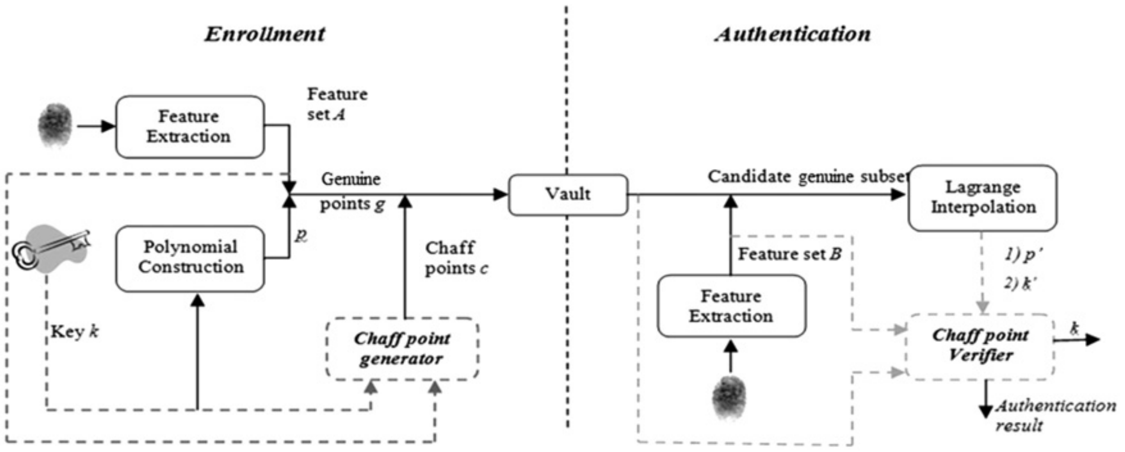

Fig. 3 *Proposed fuzzy vault scheme with chaff point generation and chaff point verifier*

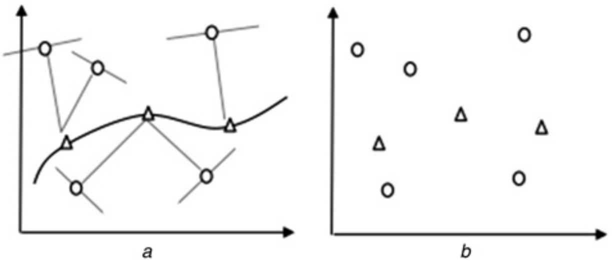

Fig. 4 *Demo of chaff points generation process and its result (circles represent genuine points; triangle points are chaff ones)*

Step 2: Apply the polynomial reconstruction algorithm (Lagrange interpolation algorithm) with D_1 as input to get a candidate key k_1.
Step 3: Select the subset $G_1 \subseteq V$ that contains points belonging to a graph of polynomial corresponding to the candidate key k_1.
Step 4: Apply the chaff points to the generator algorithm with input as G_1, k_1. Within this step, the generator algorithm can stop immediately if:

- *Missing valid chaff points* $\exists(c_i, c_j) \in \mathbb{F}_q^2|(c_i, c_j) \notin V$, Verify $((c_i, c_j)) ==$ true
- *Existing strange points cannot be generated by a generator* $\exists(s, s_j) \in \mathbb{F}_q^2|(s_i, s_j) \notin V_{\text{generate}}$

Step 5: If step 4 succeeds, return true. Otherwise, try with another subset of D if any or return false if nothing left to choose.

In the case of a combination of candidate key and candidate genuine points regenerate exactly the other points of vault, then the authentication will be successful.

4 Security analysis

4.1 Blend substitution attack problem

Suppose that the attacker has permission to access directly to template database server. To successfully attack our system via blend substitution method, the attacker has to randomly delete at least $(n+1)$ points and injects his own fake points into our vault.

- In case deleted points are genuine, take a look at h_0 of a genuine user before and after his vault being attacked:

Before

$$h_0 = \left(g_0, \ldots, g_{m-1}, k_0, \ldots, k_{n-1}\right) \in \mathbb{F}_q^{n+m}$$

After

$$h_0' = \left(g_0, \ldots, g_{m-2}, k_0, \ldots, k_{n-1}, r\right) \in \mathbb{F}_q^{n+m-1}$$

The difference between h_0 and $h_0{}^{\wedge'}$ leads to a failed result from a *chaff_verifier* right after the first valid chaff points are generated.

- If deleted points are chaff points, there is no change in h_0 as in the previous case. However, when generating this specific chaff points, algorithms will stop immediately by the condition of missing valid chaff points

$$\exists\left(c_i, c_j\right) \in \mathbb{F}_q^2\Big|\left(c_i, c_j\right) \notin V,\ \text{Verify}((c_i, c_j)) == \text{true}$$

Within both cases, a genuine user will be locked and the problem will be reported and detected.

On the other hand, when an attacker just tries to insert his fake information without deleting the original vault's points, our verifier can detect strange points at the final step when comparing a generated vault with the original one. The algorithm stops with a failed result because of the following condition

$$\exists\left(s, s_j\right) \in \mathbb{F}_q^2\Big|\left(s_i, s_j\right) \notin V_{\text{generate}}$$

4.2 CRC problems

By removing CRC from the general schema, all previous reported problems related to CRC are eliminated. Furthermore, the input of polynomial construction will be reduced in length while still keeping sufficient degree to ensure the security level, as a result, decreasing FRR.

4.3 Entropy evaluation

Hash function, in general, is sensitive to bit change. So, by continuous concatenating hash result with the original secret key, we can vary the linear function and chaff points, as a consequence, keeping the 'random characteristic' for chaff point generator module.

In case the attacker tries a brute force attack by investigating all possible hashing values of current hash function, the linear projection will be an additional protection layer. Linear projection helps to remove all links between hash values. Recall that the chaff points are the result of project genuine points to 2D line but we save nothing related to the 2D line as the attacker only has information about V with points $(p_x, p_y) \in \mathbb{F}_q^2$. Information about the linear polynomial is still kept secret

$$P_{\text{line}} = p_0X + p_{n+m-1}$$

Furthermore, p_0 and p_{n+m-1} in $P_{\text{line}} = p_0X + p_{n+m-1}$ are just small parts of

$$h_1 = \left(p_0, \ldots, p_{n+m-1}\right) \in \mathbb{F}_q^{n+m}$$

In case the attacker has information about one specific genuine point or chaff point, other chaff points of the current hashing chain are still kept safe because of the combination with secret key when performing hashing. Recall that within the chain of the hashing inputwhen the output of the current step is

$$h_1 = \left(p_0, \ldots, p_{n+m-1}\right) \in \mathbb{F}_q^{n+m}$$

Next step's input will be

$$h_0 = \left(p_0, \ldots, p_{m-1}, \boldsymbol{k}_0, \ldots, \boldsymbol{k}_{n-1}\right) \in \mathbb{F}_q^{n+m}$$

5 Experiments

To evaluate the proposed scheme, two simultaneous systems are built up:

- An original CRC-based fuzzy vault using Lagrange interpolation and 16 bits CRC to retrieve and verify the secret key.
- Our proposed system with application of chaff generator and verifier.

First of all, FVC2002-DB1 is chosen as our main testing database. FVC2002-DB1 [19] is a public domain database with 800 images (100 fingers ×8 impressions/fingers) of size 560 × 296 and resolution 569 dpi. Within our experiments, only the first two impressions of each finger were used (impression 1 and 2).

The evaluation of results is based on the two main criteria: genuine accept rate (GAR) and FAR. GAR represents the percentage of successful authentication of genuine users. For testing GAR, one impression is used as an encoding template and the other of the same fingerprint is the decoding one. On the other hand, FAR shows the amount of successful authentication of

Table 2 Summary of experiment's parameters

	c	*G*	*D*	*N*
value	300	30	10	[8–11]

illegal users. Within this case, the decoding template will be selected from other users of template database.

Turning to the basic parameter of fuzzy vault scheme, as can be seen from Table 2, the number of chaff points c is set to be ten times as many as the number of genuine points. The minimum distance of points in vault is $d = 10$. Finally, the degree n of the main polynomial p, which has a significant effect on GAR and FAR, is chosen within the range from 8 to 11.

By changing the secret keys, ten experiment cycles are applied on 200 chosen impressions (100 fingers ×2 impressions/fingers). Within each cycle, for an evaluation of GAR, impressions of the same fingerprint are used for both encoding and decoding (four combinations/fingerprint). Meanwhile, FAR's test cases are constructed by choosing impressions of one fingerprint (two impressions/fingerprints) for encoding and two random impressions of another fingerprint as decoding templates. Totally, experimentalists performed 4000 test cases for GAR and 4000 test cases for FAR.

From Table 3, it can be seen that the proposed scheme has significant enhancement on GAR, or decreasing FRR. The additional 16 bits CRC on secret key of the original system can be seen as the main reason for the difference. Within the same original secret key, polynomial generation module of the original scheme needs to work with longer input (secret key adds 16 bits CRC) and outputs a higher degree of polynomial. As a consequence, the system required additional genuine points for successful authentication.

The proposed method mainly focuses on chaff point generation. Chaff points analysis attack is reported as one of the most potential attacks when dealing with chaff points. Therefore, within this experiment, we put on testing two important characters of chaff as:

- First, distribution of chaff points over vault space.

As in [13], by statistically evaluating the space between points in vault, chaff points can be eliminated by filtering those that have lower 'free space'. To assess this flaw on the proposed system, we quantised position information of chaff points by classifying each point into small, identical rectangles. For example, with fingerprint image of 560 × 296 as in FVC2002-DB1, splitting the whole space into rectangles of size 30 × 30, we will have totally 20 × 10 = 200 rectangles.

Consider all vaults generated by each scheme, chaff points' position are collected and counted to the corresponding rectangle. The result of counting on two systems is reported in Fig. 5.

From the above two figures, it can be seen that there are no significant changes in the distribution of chaff points when applying the new proposed scheme.

- Second, the distance between all vault points as well as genuine and their chaff points.

The mean value of the distance between all points, genuine points versus chaff points, of both schemes are calculated, the results of evaluation for each scheme are reported in Table 4. Particularly, within each system (*original* and *proposed* systems), the mean of distance between the points of each vault are calculated using the following formula

$$M_s = \frac{\sum_{\substack{1\le i\le g+c\\ 1\le j\le g+c}} d(i,j)}{(g+c)^2}$$

For the whole experiments, we have

$$M = \frac{\sum_{1\le s\le S} M_s}{S}$$

The mean of distance between genuine points and their corresponding chaff points of each vault

$$m_s = \frac{\sum_{\substack{1\le i\le g\\ 1\le j\le c}} d(i,j)}{g\times c}$$

For whole experiments

$$m = \frac{\sum_{1\le s\le S} m_s}{S}$$

where S is the total number of test cases performed; and $d(i, j)$ is the distance between two points i, j of vault.

The proposed system does not have any effect on the current distribution of points in vaults. The mean values of all points within the vault are around 200 (*original* 209.33, *proposed* 190.67). There is just a soft decrease when comparing the proposed and original schemes. Considering the proposed system,

Table 3 GAR and FAR of original and proposed systems for FVC2002-DB1

	n = 8		*n* = 9		*n* = 10		*n* = 11	
	GAR	FAR	GAR	FAR	GAR	FAR	GAR	FAR
original, %	80	9	77	6	73	2	69	0
proposed, %	92	7	81	13	79	3	74	0

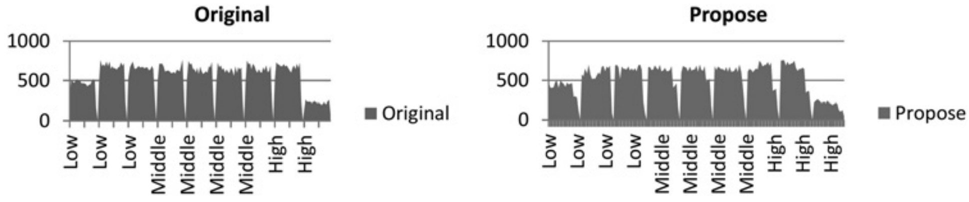

Fig. 5 *Distribution of chaff points over vault space of original scheme (n = 9, c = g × 10. Following the position of each rectangle we classified them into groups for the easiness of sum up. Rectangles at the bottom of the image will be marked as low; those in the middle are marked as middle; the others are the high ones). The y-axis shows the amount of chaff that lay in the corresponding rectangles*

Table 4 Mean distance of points in vault for original and proposed systems with $n = 8$

	$n = 8$	
	M	m
original	209.33	205.77
proposed	190.67	189.47

the mean distance between chaff points and their corresponding genuine points (189.47) is almost the same with the one of all vault points (190.67). From this observation, it is quite difficult for the attacker to apply statistical attack on vault generated by our proposed system.

6 Conclusions

In this paper, our main contribution is to address a significant flaw of the current CRC-based fuzzy vault system, the blend substitution attack. As a consequence, additional non-random chaff point generator and verifier modules are introduced. By applying continuous hashing and linear projection, the new modules can detect any modification from the original vault, thus eliminating blend substitution attack on traditional CRC-based fuzzy vault. Security analysis of our new scheme and experimental results on FVC2002-DB1 datasets are provided. The analysis and practical results show that the proposed scheme's overall GAR increases by an order of magnitude (e.g. about 12% with the main polynomial of degree 8).

7 Acknowledgments

This research was funded by the Vietnam National University, Ho Chi Minh City (VNU-HCM) under grant number B2013-20-02. The authors thank each member of D-STAR Lab (www.dstar.edu.vn) for their enthusiastic support and helpful advices during the time of this research.

8 References

1 Sousedik, C., Busch, C.: 'Presentation attack detection methods for fingerprint recognition systems: a survey', *IET Biometrics*, 2014, **3**, (4), pp. 219–233

2 Rathgeb, C., Uhl, A.: 'A survey on biometric cryptosystems and cancelable biometrics', *EURASIP J. Inf. Secur.*, 2011, **3**, (1), pp. 1–25

3 Juels, A., Sudan, M.: 'A fuzzy vault scheme', *Des. Codes Cryptogr.*, 2006, **38**, (2), pp. 237–257

4 Uludag, U., Pankanti, S., Jain, A.K.: 'Fuzzy vault for fingerprints'. Proc. of Audio-and Video-Based Biometric Person Authentication, January 2005, pp. 310–319

5 Uludag, U., Jain, A.: 'Securing fingerprint template: fuzzy vault with helper data'. Proc. Computer Vision and Pattern Recognition Workshop, June 2006, pp. 163–163

6 Nandakumar, K., Jain, A.K., Pankanti, S.: 'Fingerprint-based fuzzy vault: implementation and performance', *IEEE Trans. Inf. Forensics Secur.*, 2007, **2**, (4), pp. 744–757

7 Nandakumar, K., Nagar, A., Jain, A.K.: 'Hardening fingerprint fuzzy vault using password'. Proc. Advances in Biometrics, 2007, pp. 927–937

8 Clancy, T.C., Kiyavash, N., Lin, D.J.: 'Secure smartcard based fingerprint authentication'. Proc. ACM SIGMM Workshop on Biometrics Methods and Applications, November 2003, pp. 45–52

9 Poon, H.T., Miri, A.: 'On efficient decoding for the fuzzy vault scheme'. Proc. Information Science, Signal Processing and their Applications (ISSPA), July 2012, pp. 454–459

10 Benhammadi, F., Bey, K.B.: 'Password hardened fuzzy vault for fingerprint authentication system', *Image Vis. Comput.*, 2014, **32**, (8), pp. 487–496

11 Khalil-Hani, M., Marsono, M.N., Bakhteri, R.: 'Biometric encryption based on a fuzzy vault scheme with a fast chaff generation algorithm', *Future Gener. Comput. Syst.*, 2013, **29**, (3), pp. 800–810

12 Mihailescu, P.: 'The fuzzy vault for fingerprints is vulnerable to brute force attack', arXiv preprint arXiv:0708.2974, 2007

13 Chang, E.C., Shen, R., Teo, F.W.: 'Finding the original point set hidden among chaff'. Proc. ACM Symp. on Information, Computer and Communications Security, March 2006, pp. 182–188

14 Poona, H.T., Miria, A.: 'A collusion attack on the fuzzy vault scheme', *ISC Int. J. Inf. Security*, 2009, Bd, 1, (1), pp. 27–34

15 Scheirer, W.J., Boult, T.E.: 'Cracking fuzzy vaults and biometric encryption'. Proc. Biometrics Symp., September 2007, pp. 1–6

16 Hong, S., Jeon, W., Kim, S., *et al.*: 'The vulnerabilities analysis of fuzzy vault using password'. Proc. Future Generation Communication and Networking, December 2008, vol. 3, pp. 76–83

17 Tams, B.: 'Unlinkable minutiae-based fuzzy vault for multiple fingerprints', *IET Biometrics*, 2015 DOI: 10.1049/iet-bmt.2014.0093, Online ISSN 2047-4946, Available online: 23 June 2015

18 Tams, B., Mihailescu, P., Munk, A.: 'Security considerations in minutiae-based fuzzy vaults', *IEEE Trans. Inf. Forensics Secur.*, 2015, **10**, (5), pp. 985–998

19 Maio, D., Maltoni, D., Cappelli, R., *et al.*: 'FVC2002: second fingerprint verification competition', *Proc. Pattern Recognit.*, 2002, **3**, pp. 811–814