Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 4722 – 4733
© TÜBİTAK
doi:10.3906/elk-1609-258

Neoteric chaff generation method of fingerprint fuzzy vault

Nancy SINGLA*, Sanjeev SOFAT, Manvjeet KAUR
Department of Computer Science & Engineering, PEC University of Technology, Chandigarh, India

Received: 27.09.2016 • **Accepted/Published Online:** 15.09.2017 • **Final Version:** 03.12.2017

Abstract: Authentication using biometrics has an edge over traditional authentication mechanisms. Fuzzy vault is an eminent biometric cryptosystem technique that aims to protect a secret using a biometric template. Authorized users can, however, access the secret by providing the valid biometric. In a fuzzy vault, noise is incorporated in the form of chaff points in order to increase the security. While implementing fuzzy vault schemes in real scenarios, the most critical but computing-intensive and time-consuming task is chaff generation. In this paper, a neoteric method for chaff generation is proposed, which is less time-consuming in generating large numbers of chaff points.

Key words: Biometrics, biometric template, fingerprint security, fuzzy vault, chaff generation

1. Introduction

In today's world, the authentication and verification of a person is of more concern. As the level of security breaches increases, biometric technologies are becoming a base in providing large numbers of highly secure solutions for identification and personal verification. Biometrics refers to the automatic recognition of individuals based on their physiological and/or behavioral characteristics [1]. A biometric system is defined as a technological system that uses unique biological traits to identify an individual. During the recognition process in a biometric system, first the user presents the biometric trait and the biometric sample is obtained by the sensor. Then a feature vector (template) is formed, which is obtained by grouping the quantified characteristic parameter of the sample. The template is then matched against other templates stored in the database that correspond to the individual whose identity is claimed and, depending upon the level of similarity, a score is generated. Finally, based on the generated score, the biometric system makes a decision about whether to accept the sample as a genuine user or to reject it as imposter.

Biometric systems have enhanced the security level but are also vulnerable to various types of attacks [2]. Attack on the biometric template stored in the system database is one of the most potentially damaging attacks for biometric systems [3]. It becomes difficult to revoke the compromised template as biometric traits are immutable in nature. Therefore, it is very important to secure the biometric template in any biometric based authentication system.

Fuzzy vault is considered as a popular method for biometric template protection. It is a key binding crypto-biometric system in which fuzzy unordered sets of genuine and chaff points are used to encrypt and decrypt the secret information securely [4,5]. The process of locking and unlocking the fuzzy vault is illustrated in Figure 1 and described below.

*Correspondence: ernancysingla17@gmail.com

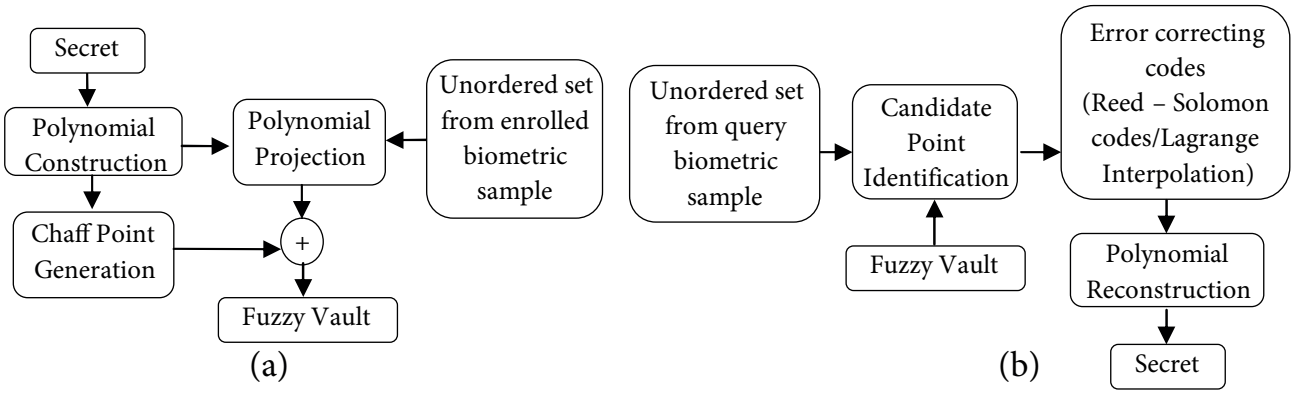

Figure 1. a) Locking fuzzy vault, b) unlocking fuzzy vault.

- Locking/encoding fuzzy vault: An unordered set of points called genuine points from the enrolling biometric sample are extracted. A randomly chosen secret key is divided into $(n+1)$ equal parts and an n-degree polynomial is constructed with its coefficient from the secret key. Considering that only the genuine minutiae points will satisfy the polynomial, a large number of random points called chaff points, which do not lie on that polynomial, are generated. The union of the genuine point and chaff point sets forms the fuzzy vault.

- Unlocking/decoding fuzzy vault: An unordered set of points from the query template is extracted and used to unlock a secret key. A polynomial can be reconstructed if there is substantial overlap between the enrolled and query unordered set of points and thus the secret key can be decoded securely using an error correction scheme (Lagrange interpolation).

The security of the fuzzy vault depends on the intricacy of the polynomial reconstruction and the number of generated chaff points. Generally, the number of chaff points generated is approximately ten times the genuine minutiae points due to the tradeoff between the computational complexity of the brute force attack and storage requirements of the vault [6,7]. In real-time applications, the captured fingerprint image is of good quality when it is acquired in a controlled environment. More than 20 quality minutiae points can be easily extracted from such fingerprint samples. Therefore, the generally needed chaff points are considered as more than 200.

Addition of chaff points during the locking phase of fuzzy vault aims to produce random noise in order to hide the genuine points from the attacker and thus enhance the security of the fuzzy vault. Generating chaff points in a fingerprint-based fuzzy vault is considered as the most computationally intensive task as compared to other processes in fuzzy vault as it consumes 42.5% of the total execution time of the fuzzy vault locking [8]. There is a need to develop a fast chaff generation algorithm. Even a small amount of reduction in execution time can make a significant difference while implementing the fuzzy vault system in real-time scenarios that involve dealing with huge datasets. In this paper, a chaff point generation algorithm for a fingerprint-based fuzzy vault is proposed, which is faster than existing chaff point generation algorithms for generating the required number of chaff points.

The remainder of the paper is organized as follows. Section 2 describes the review of the related work in the concerned field. Section 3 describes the proposed chaff generation algorithm. In Section 4, results and analysis are discussed. Finally, in Section 5, a conclusion is drawn.

2. Related work

Fingerprints are considered as the most suitable modality for fuzzy vault schemes because the minutiae set is an unordered set, which is the basic requirement for fuzzy vault schemes [4].

In the literature, much research has been done on fingerprint fuzzy vaults by various pioneering researchers. Implementation of a fingerprint fuzzy vault with minutiae descriptors containing information about orientation and ridge frequency was presented in [9]. Yang and Verbauwhede [10] proposed an automatic alignment using reference minutiae extracted during the vault encoding and decoding. Fingerprint alignment based on helper data using the iterative closest point algorithm was proposed in [11,12]. Moon et al. [13] proposed a geometric hashing technique to perform alignment in a minutiae-based fingerprint fuzzy vault. An improved fuzzy vault was proposed in [14] by integrating information about the local ridge orientation of minutiae. An alignment-free fingerprint cryptosystem based on a fuzzy vault scheme was proposed in [15] by fusing the transformation invariant local features (minutiae descriptors, minutiae local structure) of fingerprints. A hashed fingerprint fuzzy vault was proposed and implemented in [16] to defy correlation attacks on fuzzy vaults. Benhammadi and Bey [17] proposed a password-hardened fingerprint vault in order to provide diversity and security for fingerprint minutiae templates during storage and comparison. The authors implemented a hybrid approach by combining a user-generated password fuzzy vault with the transformed minutiae pairwise features. A Hadamard-based fingerprint fuzzy vault was proposed by Bansal et al. to provide revocability and diversity [18]. Genuine minutiae points are transformed using Hadamard transformation. Nguyen et al. proposed a fuzzy vault for distorted fingerprint images using the information of ridge features as they are invariant to geometrical transformations [19].

Fingerprint fuzzy vaults use the chaff points to enhance the security by concealing the genuine minutiae points amidst the chaff points. In the literature, several methods for generating and placing chaff points have been proposed by various authors.

Clancy et al. [5] proposed a fingerprint fuzzy vault by using multiple minutiae locations as a locking set. The Euclidean distance method was used to generate chaff points, which should be placed at an acceptable (threshold) distance from the genuine minutiae points. The complexity of generating the chaff points was $O(n^3)$. Clancy's method of generating chaff points was also used by Uludag et al. [9] and Nandakumar et al. [12].

A chaff point placement method was proposed in [20] to prevent some inferences about the location of genuine points. Apart from considering the Euclidean distance between chaff points and genuine points, the authors also considered interchaff distance. In order to reduce the success rate of the attacker, a fake polynomial is embedded in the fuzzy vault along with the secret polynomial.

A fast chaff-generating algorithm was proposed by Khalil-Hani et al. [8,21] based on the algorithm described in [5]. The authors used the mathematical theorem of circle packing. In the fuzzy vault set, a new point is added only if the boundaries of two existing points do not overlap with each other. It uses only addition, subtraction, and comparison operations and hence is less computationally intensive with computational complexity of $O(n^2)$ as compared to [5]. However, it takes more computation time while generating chaff points greater than 200.

A less time-consuming chaff point generation technique was proposed by Nguyen et al. [6,22] for producing more than 200 chaff points. The authors created image cells by splitting the fingerprint image into segments and each image cell was surrounded by eight adjacent image cells. The randomly generated candidate chaff point is considered as the chaff point if no other point lies inside the image cell in which the candidate chaff

point is generated and the distance between candidate chaff points and the points in the adjacent eight image cells is more than the threshold.

A chaff generation algorithm that uses axis difference instead of Euclidean distance was proposed by Hooda and Kaur [23]. It is faster as compared to [5] as simple arithmetic operators like addition and subtraction are used.

Nguyen et al. [24] proposed a modified chaff point generator in which chaff points are generated using continuous hashing and linear projection. Blend substitution attack can be prevented by detecting any modification done to the original vault as chaff points are considered as signatures for the combination of the biometric template and secure key.

3. Proposed chaff generation algorithm

The proposed chaff generation algorithm modifies Nguyen's chaff generation algorithm [6,22] so that it takes a consistent time for generating the required number of chaff points irrespective of the number of genuine minutiae points. Discussed below are the baseline Nguyen algorithm and the proposed chaff generation algorithm.

A fingerprint image is divided into $m \times n$ image cells, where m represents the number of rows and n represents the number of columns. Each image cell is of size $w \times w$. Each image cell would either contain a unique genuine minutiae point or a chaff point or will remain empty. The image cell is addressed as (x_{cell}, y_{cell}), where $x_{cell} \in [0, 1, 2....m]$ and $y_{cell} \in [0, 1, 2....n]$. The pixel inside an image cell is addressed as (x_{order}, y_{order}), where $x_{order} \in [0, 1, 2....w-1]$ and $y_{order} \in [0, 1, 2....w-1]$.

3.1. Nguyen's chaff generation algorithm

Nguyen's approach [6,22] makes use of the three-dimensional user data type ImagecellData to monitor image cells containing minutiae points or chaff points, x-y coordinates of the points, and the orientation θ. A NeighborList set containing minutiae or chaff points in eight adjacent image cells is also formed.

Nguyen's algorithm: chaff point generation

1. Generate a set SMT $\{m_j\}_{j=1}^{r}$ of genuine minutiae $m = (x, y, \theta)$ extracted from a fingerprint where x, y are coordinate values and θ is orientation.
2. Update ImagecellData using SMT to monitor the location of the genuine minutiae points.

   if (image cell contains genuine minutiae or chaff)

   set first element of ImagecellData as 1 and other two elements as the value of x and y coordinate, respectively

   else

   set first element of ImagecellData as 0 and other two elements as NULL
3. Repeat steps from 4 to 8 until ECM (encoded chaff minutiae) set has 's' members.
4. Randomly generate an empty image cell (x_{cell}, y_{cell}).
5. Generate an order (x_{order}, y_{order}) within (x_{cell}, y_{cell}) and calculate actual coordinates (u_x, v_x, θ_x).

6. Obtain and add the points belonging to eight adjacent neighborhoods of image cell (x_{cell}, y_{cell}) to NeighborList.

   if (NeighborList is empty)

   add (u_x, v_x, θ_x) to CM (set of chaff points) and update (x, y) in ImagecellData

   else

   calculate Euclidian distance between (x, y) and points in NeighborList

7. if $(Euclidean\, distance > threshold)$

   add (u_x, v_x, θ_x) to CM and update (x, y) in ImagecellData

8. Encode the chaff points in $\{F = GF(2^{16})\}$ and add the encoded point to the ECM set.

3.2. Proposed chaff generation algorithm

In the proposed chaff generation algorithm two ArrayList sets, namely Used and Free, are maintained. Used ArrayList stores image cells containing genuine minutiae or chaff points while Free ArrayList stores image cells that do not contain any genuine minutiae or chaff points. Initially Used ArrayList will store image cells containing only genuine minutiae points. All other remaining image cells stored in Free ArrayList are shuffled once for generating randomness. In the proposed chaff generation algorithm, a new chaff point is generated by considering a window around the candidate chaff point, which is randomly generated within an image cell selected from Free ArrayList. If no other point (genuine minutiae point or chaff point) lies inside the considered window, then the candidate chaff point is considered as a valid chaff point or otherwise discarded. Chaff points are inserted according to algorithm described below and illustrated in Figure 2.

Proposed algorithm: chaff point generation

1. Repeat steps 2 to 6 until 'n' chaff points are generated.
2. Take image cell IC (x_{cell}, y_{cell}) from Free Arraylist and remove it from there.
3. Randomly generate order $(x_{order} y_{order})$ within IC (x_{cell}, y_{cell}) and calculate its actual coordinates $P(x, y)$ using Eqs. (1) and (2).

$$x = x_{cell} \times w + x_{order} \tag{1}$$

$$y = y_{cell} \times w + y_{order} \tag{2}$$

   $P(x, y)$ *can be considered as a candidate chaff point.*

4. Consider a window W' centered at $P(x, y)$ of same size $(w \times w)$ as that of the image cell and calculate the four boundary points (B_1, B_2, B_3, B_4) using Eqs. (3) to (6).

$$B_1(x, y) = (P(x) - w/2,\ P(y) - w/2) \tag{3}$$

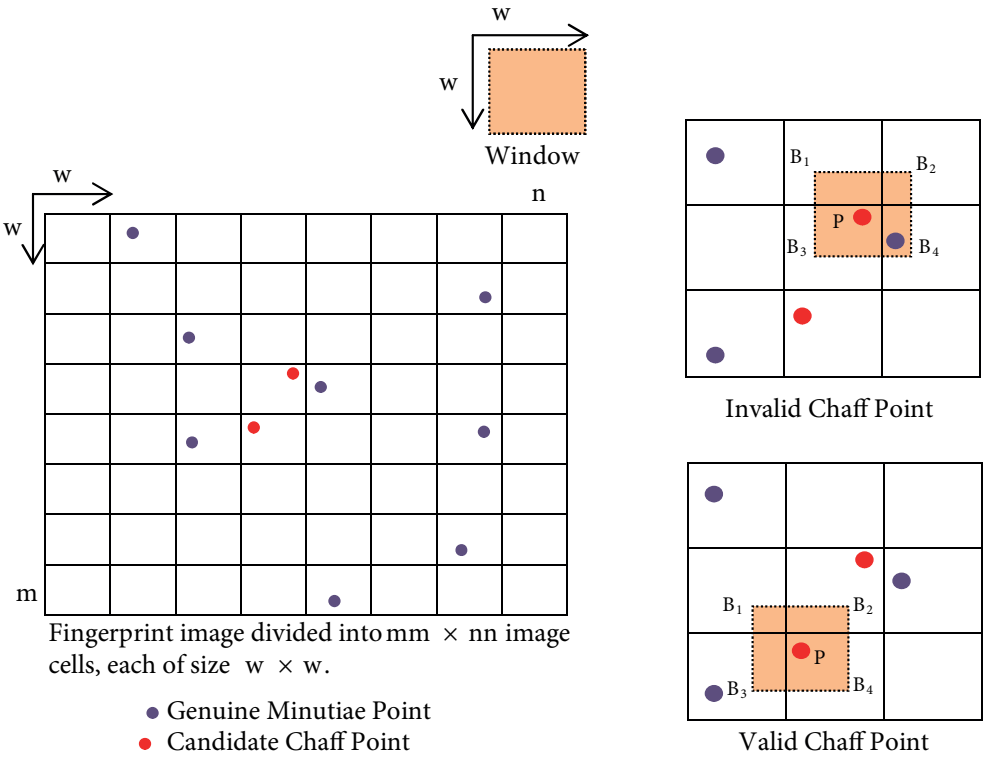

Figure 2. Illustration of proposed chaff point generation.

$$B_2(x,y) = (P(x) - w/2,\ P(y) + w/2) \tag{4}$$

$$B_3(x,y) = (P(x) + w/2, P(y) - w/2) \tag{5}$$

$$B_4(x,y) = (P(x) + w/2,\ P(y) + w/2) \tag{6}$$

5. for $i = 1\,to\,4$

$IC'\left(x'_{cell}, y'_{cell}\right) = B_i(x/w, y/w)$

if $(IC'\left(x'_{cell}, y'_{cell}\right) = IC(x_{cell}, y_{cell}))$

continue

else

if $(IC'\left(x'_{cell}, y'_{cell}\right)$ exists in Used Arraylist)

Obtain $\left(x'_{order}, y'_{order}\right)$ from Used Arraylist

if $(\left(x'_{order}, y'_{order}\right)$ falls inside $W')$

Add $IC(x_{cell}, y_{cell})$ to Free Arraylist

go to step 2

else

continue

else

continue

$P(x, y)$ *is now considered as a valid chaff point.*

6. Add $P(x, y)$ to Used Arraylist.

4. Results and analysis

The performance of the proposed algorithm is analyzed using standard FVC 2002 and a live fingerprint database acquired using the Verifier 300LC fingerprint scanner and Mega Matcher SDK 4.0 at a resolution of 500 dpi. The database consists of fingerprint samples of 100 users with 2 fingerprint impressions of the same finger for each user. The size of the fingerprint samples is 388 $\times$ 374. Size of the image cell and window $W^{'}$ is considered as 9×9. The proposed methodology is stimulated using MATLAB version 2015a running on an HP laptop with the Microsoft Windows 8.1 system having processor configurations as Intel Core i5-5200U CPU @ 2.20 GHz and 64-bit operating system. In order to have a fair comparison, all the chaff generation algorithms have been tested on this same hardware platform.

4.1. Functionality analysis

The number of chaff points added is ten times the genuine minutiae points. The performance of the proposed chaff generation algorithm is compared with the existing chaff point generation approaches of Clancy et al. [5], Khalil-Hani et al. [8,21], Nguyen et al. [6], and Hooda and Kaur [23]. Functionality analysis is performed in terms of time taken and number of candidate chaff points generated for adding the required number of chaff points.

4.1.1. Analysis on the basis of chaff point generation time

The average execution time comparison to generate the required number of chaff points using Clancy's, Khalil's, Nguyen's, Hooda's, and the proposed chaff point generation algorithm is graphically shown in Figure 3.

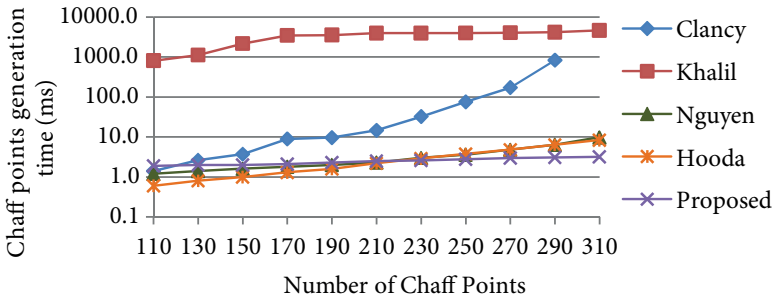

Figure 3. Execution time comparison of generating chaff points.

Clancy et al. [5] make use of the Euclidean distance to validate the chaff points depending on the threshold of minimum distance. It computes the Euclidean distance between a candidate chaff point and all

other existing points in the vault. Clancy's chaff point generation approach takes less time as compared to the proposed chaff point generation algorithm for generating less than 130 chaff points. This is because initially the number of existing points in the vault is lower and therefore the Euclidean distance needs to be computed comparatively fewer times. However, as the number of chaff points needed in the vault increases, time to compute the Euclidean distance with all the existing points also increases linearly. Thus, Clancy's approach takes more time while generating large numbers of chaff points and it fails to generate the required number of chaff points (ten times the genuine minutiae points) for the genuine minutiae points above 30. In the proposed chaff point generation algorithm, instead of computing Euclidean distance, uniform steps (considering a window and comparing candidate points with boundary points) are to be performed for each chaff point generation. This process takes less time in generating large numbers of chaff points while taking more time for generating less than 130 chaff points as compared to Clancy's approach.

The chaff generation approach of Khalil-Hani et al. [8,21] makes use of the circle packing scheme, which takes more time for generating chaff points as compared to the proposed chaff point generation algorithm. This is because in order to generate a chaff point in Khalil-Hani's chaff generation approach, it needs to check that the new point's boundary is not overlapping with any other existing point's boundary. In the proposed chaff generation algorithm, the candidate chaff point is at most compared with only the four boundary points of the window formed by considering the candidate chaff point as the center.

The chaff generation approach of Nguyen et al. [6] splits a fingerprint image into image cells. Chaff points are randomly generated, which must be unique points in an image cell, and the distance between the new point and the existing points in the eight adjacent image cells is greater than or equal to the threshold. Initially while generating chaff points, the probability of randomly choosing a unique point in an image cell is more. Thus, Nguyen's approach takes less time while generating chaff points for less than 21 genuine minutiae points. However, as the required number of chaff point increases, the probability of randomly choosing a unique point in an image cell decreases, causing it to repeatedly choose the image cell already containing the genuine minutiae point or a chaff point. Hence, in Nguyen's algorithm chaff point generation time increases as the required number of chaff points increases. In the proposed algorithm, separate lists are maintained for storing free and used image cells and therefore it takes a consistent time for generating the required number of chaff points irrespective of the number of genuine minutiae points.

Hooda and Kaur [23] use an axis difference to place a chaff point. While generating more than 210 chaff points, it takes more time as compared to the proposed algorithm as it needs to compute the axis difference between the candidate chaff point and all other existing points in the vault. Initially it takes less time as the total number of points in the vault is lower.

4.1.2. Analysis on the basis of number of candidate chaff points

The candidate chaff point is the randomly selected point that can be considered as the valid chaff point only if it meets certain predefined criteria. The total number of candidate chaff points generated using Clancy's, Khalil's, Nguyen's, Hooda's, and the proposed chaff point generation algorithm is graphically shown in Figure 4. There is variation in the number of candidate chaff points because of the different underlying chaff generation algorithms being used. This variation in the number of candidate chaff points leads to the difference in the execution times in spite of the fact that the number of valid chaff points being generated is the same for all approaches. In the proposed algorithm, the total number of candidate chaff points generated is less as compared to the existing chaff point generation methods because a separate list is maintained for storing free image cells in which chaff points can be generated and used image cells that already contain a genuine minutiae point or

the chaff point. Therefore, it eliminates the possibility of choosing the image cell that is already filled with a genuine minutiae point or a chaff point and thus it generates fewer candidate chaff points. For generating 290 chaff points, the number of candidate chaff points generated in the proposed chaff point generation algorithm is 301, which is 216.6, 221.7, 5.1, and 3.1 times less than in Clancy's, Khalil's, Nguyen's, and Hooda's chaff generation approach, respectively.

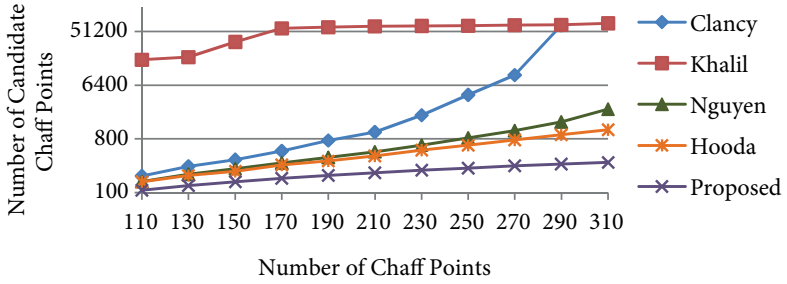

Figure 4. Comparison of number of generated candidate chaff points.

4.2. Performance analysis

The fuzzy vault system using the proposed chaff point generation algorithm is implemented using a key size of 128 bits with an 8th-degree polynomial. The performance of the fuzzy vault scheme using the proposed chaff point generation algorithm is evaluated using the false rejection rate (FRR) and false acceptance rate (FAR). For calculating the FRR, the first fingerprint impression of each user from the created database is considered as the template sample and the second fingerprint impression is considered as the query sample. The total number of genuine attempts is 100. For calculating the FAR, the first fingerprint impression of a particular user and all the other users are considered as the template and query sample, respectively. The total number of imposter attempts is 9900. For the proposed methodology, FRR and FAR are found to be 10% and 0.05%, respectively.

4.3. Security analysis

The proposed chaff point generation algorithm could resist the attack described by Chang et al. [25]. In this attack, an attacker can identify genuine minutiae points by applying brute force attack to measure the free space around the point, known as the 'degree of freedom' of all existing points in the vault, in order to eliminate chaff points. While generating a larger number of chaff points, an attacker may easily identify chaff points that are generated later as they have comparatively smaller degrees of freedom. The proposed approach modifies Nguyen's chaff point generation algorithm; however, it splits the fingerprint image into equally sized segments called image cells and the valid chaff point in an image cell would be unique. Thus, degrees of freedom would not be affected by the order of the generated chaff points and the attacker could not distinguish between genuine minutiae points and chaff points, as was experimentally proven by Nguyen et al. [22].

Security of a fuzzy vault scheme based on the complexity to reconstruct the polynomial is analyzed using the min-entropy method [26]. The min-entropy of the minutiae template MT given vault V under the assumption of uniform minutiae distribution is computed using Eq. (7).

$$H_{\infty}(M^T|V) = -\log_2\left(\left(\frac{r}{n+1}\right) \Big/ \left(\frac{r+s}{n+1}\right)\right) \tag{7}$$

Here, r denotes the number of minutiae points, n denotes the degree of the polynomial, and s denotes the number of chaff points. Consider the values of r, n, and s to be 21, 8, and 210, respectively. The total number of possible combinations is 4.4077×10^{15}, but in order to decode the secret successfully, it needs 293,930 combinations. Hence, for an attacker, it would need 6.6×10^{-11} evaluations to decode the secret. On the basis of this analysis, the security of the proposed vault is approximately 34 bits.

The security of the fuzzy vault can be increased by increasing the degree of the polynomial and key size, making it hard to reconstruct the polynomial as shown in the Table. It can be concluded that with the increase in the degree of the polynomial, the probability of decoding the secret is decreasing and the overall security of the biometric system in terms of entropy is increasing, making the system harder to break by brute force attack. However, there is a strong trade-off between security and complexity in the biometric system. As the security increases, the computational complexity of the system also increases, which will lead to an increase in overall fuzzy vault encoding and decoding time. The choice of degree is strongly application-dependent. However, for implementing a fuzzy vault system in real-time scenarios, the proposed fuzzy vault system with degree 8 achieves adequate security.

Table. Entropy values by varying polynomial degree.

Degree of polynomial	Fuzzy vault key size (in bits)	Total number of possible combinations	Combinations to decode the secret	Probability of decoding the secret	Entropy (in bits)
8	128	4.4077×10^{15}	2.9×10^{5}	6.6×10^{-11}	34
9	144	9.7852×10^{16}	3.5×10^{5}	3.6×10^{-12}	38
10	160	1.9659×10^{18}	3.5×10^{5}	1.8×10^{-13}	42
11	176	3.6042×10^{19}	2.9×10^{5}	8.1×10^{-15}	47
12	192	6.0717×10^{20}	2×10^{5}	3.3×10^{-16}	51
13	208	9.4545×10^{21}	1.1×10^{5}	1.22×10^{-17}	56
14	224	1.3677×10^{23}	5.1×10^{4}	3.9×10^{-19}	61
15	240	1.8464×10^{24}	2×10^{4}	1.1×10^{-20}	66

5. Conclusion

This paper proposes a neoteric method for chaff point generation in fingerprint fuzzy vaults. Results show that the proposed algorithm takes consistent time for generating the required number of chaff points irrespective of the number of genuine minutiae points and generates fewer candidate chaff points as compared to Clancy's, Khalil's, Nguyen's, and Hooda's chaff generation algorithms. The FRR and FAR of the fuzzy vault system are evaluated to be 10% and 0.05%, respectively, which shows that the fingerprint fuzzy vault with proposed chaff generation algorithm achieves acceptable performance.

References

[1] Jain AK, Ross A, Prabhakar S. An introduction to biometric recognition. IEEE T Circ Syst Vid 2004; 14: 4-20.

[2] Ratha NK, Connell JH, Bolle RM. Enhancing security and privacy in biometrics-based authentication systems. IBM Syst J 2001; 40: 614-34.

[3] Jain AK, Nandakumar K, Nagar A. Biometric template security. EURASIP J Adv Sig Pr 2008; 2008: 579416.

[4] Juels A, Sudan M. A fuzzy vault scheme. In: Proceedings of the IEEE 2002 International Symposium on Information Theory; 30 June–5 July 2002. New York, NY, USA: IEEE. p. 408.

[5] Clancy TC, Kiyavash N, Lin DJ. Secure smartcard-based fingerprint authentication. In: Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications; 8 November 2003. New York, NY, USA: ACM. pp. 45-52.

[6] Nguyen TH, Wang Y, Ha Y, Li R. Improved chaff point generation for vault scheme in bio-cryptosystems. IET Biometrics 2013; 2: 48-55.

[7] Nandakumar K, Nagar A, Jain AK. Hardening fingerprint fuzzy vault using password. In: Proceedings of the 2007 International Conference on Advances in Biometrics; 2007. Berlin, Germany: Springer-Verlag. pp. 927-937.

[8] Khalil-Hani M, Bakhteri R. Securing cryptographic key with fuzzy vault based on a new chaff generation method. In: IEEE 2010 International Conference on High Performance Computing and Simulation; 28 June 2010. New York, NY, USA: IEEE. pp. 259-265.

[9] Uludag U, Pankanti S, Jain AK. Fuzzy vault for fingerprints. In: Proceedings of the Fifth International Conference on Audio- and Video-Based Biometric Person Authentication; 20–22 July 2005; Hilton Rye Town, NY, USA. Berlin, Germany: Springer. pp. 310-319.

[10] Yang S, Verbauwhede I. Automatic secure fingerprint verification system based on fuzzy vault scheme. In: IEEE 2005 International Conference on Acoustics, Speech, and Signal Processing; 23 March 2005. New York, NY, USA: IEEE. pp. 609-612.

[11] Uludag U, Jain A. Securing fingerprint template: fuzzy vault with helper data. In: IEEE 2006 Conference on Computer Vision and Pattern Recognition Workshop; 17–22 June 2006; Washington, DC, USA. New York, NY, USA: IEEE. pp. 163-163.

[12] Nandakumar K, Jain AK, Pankanti S. Fingerprint-based fuzzy vault: implementation and performance. IEEE T Inf Foren Sec 2007; 2: 744-757.

[13] Moon D, Lee S, Chung Y, Pan SB, Moon K. Implementation of automatic fuzzy fingerprint vault. In: IEEE 2008 International Conference on Machine Learning and Cybernetics; 12–15 July 2008. New York, NY, USA: IEEE. pp. 3781-3786.

[14] Li P, Yang X, Cao K, Shi P, Tian J. Security-enhanced fuzzy fingerprint vault based on minutiae's local ridge information. In: Proceedings of the Third International Conference on Biometrics; 2–5 June 2009; Alghero, Italy. Berlin, Germany: Springer. pp. 930-939.

[15] Li P, Yang X, Cao K, Tao X, Wang R, Tian J. An alignment-free fingerprint cryptosystem based on fuzzy vault scheme. J Netw Comput Appl 2010; 33: 207-220.

[16] Örencik C, Pedersen TB, Savaş E, Keskinöz M. Securing fuzzy vault schemes through biometric hashing. Turk J Elec Eng & Comp Sci 2010; 18: 515-539.

[17] Benhammadi F, Bey KB. Password hardened fuzzy vault for fingerprint authentication system. Image Vision Comput 2014; 32: 487-496.

[18] Bansal D, Sofat S, Kaur M. Fingerprint fuzzy vault using Hadamard transformation. In: IEEE 2015 International Conference on Advances in Computing, Communications and Informatics; 10–13 August 2015. New York, NY, USA: IEEE. pp. 1830-1834.

[19] Nguyen TH, Wang Y, Ha Y, Li R. Performance and security-enhanced fuzzy vault scheme based on ridge features for distorted fingerprints. IET Biometrics 2015; 4: 29-39.

[20] Örencik C, Pedersen TB, Savaş E, Keskinöz M. Improved fuzzy vault scheme for fingerprint verification. In: SECRYPT 2008 - International Conference on Security and Cryptography; January 2008. pp. 37-43.

[21] Khalil-Hani M, Marsono MN, Bakhteri R. Biometric encryption based on a fuzzy vault scheme with a fast chaff generation algorithm. Future Gener Comp Sy 2013; 29: 800-810.

[22] Nguyen TH, Wang Y, Nguyen TN, Li R. A fingerprint fuzzy vault scheme using a fast chaff point generation algorithm. In: IEEE 2013 International Conference on Signal Processing, Communication and Computing; 5–8 August 2013. New York, NY, USA: IEEE. pp. 1-6.

[23] Hooda R, Kaur M. Novel chaff generation for fingerprint fuzzy vault. British Journal of Mathematics & Computer Science 2015; 10: 1-9.

[24] Nguyen MT, Truong QH, Dang TK. Enhance fuzzy vault security using nonrandom chaff point generator. Inform Process Lett 2016; 116: 53-64.

[25] Chang EC, Shen R, Teo FW. Finding the original point set hidden among chaff. In: Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security; 21–24 March 2006; Taipei, Taiwan. New York, NY, USA: ACM. pp. 182-188.

[26] Nandakumar K. Multibiometric systems: fusion strategies and template security. PhD, Michigan State University, East Lansing, MI, 2008.