Turkish Journal of Electrical Engineering & Computer Sciences

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

Research Article

Turk J Elec Eng & Comp Sci
(2018) 26: 89 – 100

doi:10.3906/elk-1610-255

Real-time chaff generation for a biometric fuzzy vault

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

Received: 20.10.2016 • **Accepted/Published Online:** 15.09.2017 • **Final Version:** 26.01.2018

Abstract: Biometric technology is rapidly being adopted in wide variety of security applications. However, the system itself is not completely foolproof and is vulnerable to many attacks. Some of the attacks on the biometric system are very severe, one of which is the attack on template security. In spite of the various template security techniques presented in the literature, none of them is able to provide security, diversity, revocability, and good performance simultaneously to the biometric system. Fuzzy vault is one of the most promising bio-cryptographic techniques to prevent the template data from being misused. To make the fuzzy vault practically realizable in real-life applications especially for large databases, the chaff generation time needs to be reduced to a greater extent. This work focuses on decreasing the chaff generation time to reduce the overall vault creation time. The approach presented in the paper has also been tested on real-time dedicated hardware using fingerprint data acquired in real time by using a fingerprint sensor Verifier 300LC to bridge the gap between the research and real-time application scenarios.

Key words: Biometrics, biometric security, fingerprint recognition, template security, fuzzy vault, chaff points, microcontrollers

1. Introduction

Biometrics was introduced and evolved in the late 1970s and early 1980s. The benefits of using biometrics include enhancement of security, convenience, accountability, reduction in fraud, and delivery of enhanced services [1]. During biometric data acquisition, an unprocessed image or recording of a characteristic, referred to as raw biometric data or as a biometric sample, is provided. Once the biometric data are acquired, biometric templates can be created by a process of feature extraction. Feature extraction is the automated process of locating and encoding distinctive characteristics from biometric data in order to generate a template. It may remove noises and unwanted data and digitize biometric traits [2]. Template security is one of the prime concerns for researchers. If an adversary is able to gain unauthorized access to the template, he/she may be able to generate the original biometric data from it. This regenerated sample may be passed as an input to the biometric system, leading to a security breach [3,4]. Out of the eight major attack points on a biometric system as identified by Ratha et al. [5], attack at the stored template is very severe for the above-stated reason. There are a number of techniques available in the literature to prevent the template from being hacked or misused by malignant users. Fuzzy vault is one of the most reliable cryptographic methods to protect the template from hacking [6].

*Correspondence: manvjeet@pec.ac.in

2. Fuzzy vault scheme

Originally the fuzzy vault was proposed by Juels and Sudan [7]. Later on Clancy et al. [8] improvised the scheme by adding random noise points at some fixed distance apart from genuine points and making the scheme implementable in real scenarios. Because of its robustness this scheme has become very popular in the literature and has been tried and successfully implemented with various biometric traits, e.g. fingerprint, iris, retina, and palm print. The fuzzy vault has two main stages during its implementation: encoding and decoding [9]. The intermediate results obtained during fuzzy vault implementation for encoding and decoding stages are presented in Figure 1. During the process of encoding, generation of chaff points (noise points) takes maximum time [10]. However, as the number of chaff points is increased the security level will also increase. Hence there is a strong tradeoff between the time taken to add number of chaff points and the security level of the system. Generally chaff points to be added are 10 times the genuine minutiae points extracted from the input image at the time of enrolment [11].

The rest of the paper is organized as follows. Section 3 discusses the related work in the area, Section 4 describes the proposed methodology in detail, Section 5 discusses the experimental setup, Section 6 focuses on the results obtained, Section 7 is about performance evaluation of the system, and Section 8 mentions the conclusions drawn.

3. Related work

The basic concept of fuzzy vault was first introduced by Juels and Sudan [7] in 2002 along with the locking and unlocking algorithms of the fuzzy vault. Authors focused on the security of the fuzzy vault, stating that it depends on the number of chaff points added to the genuine minutiae set. The security of the fuzzy vault is directly proportional to the number of added noise points. If the number of chaff points is increased then the overall security of the system will also increase [7]. Clancy et al. [8] later on modified the algorithm so as to make chaff point addition realizable in real life with all its parameters optimized to a greater extent. Authors left a gap as how to pack more chaff points in same space. Nandakumar et al. [12] presented a fully automatic fingerprint fuzzy vault with iterative closest point-based alignment of input and query images. The authors concluded that the fuzzy vault that does not involve chaff point addition should be redesigned. A modified fingerprint fuzzy vault with helper data and autoalignment feature is presented in [13,14]. Hooda and Kaur [15] presented a naive idea using axis distance to reduce chaff point addition time for Clancy's algorithm. However, the threshold equivalence is not established between Euclidean distance as used by Clancy et al. [8] and axis distance is used as distance metric by the authors so as to generalize the concept.

Khalil-Hani and Bakheri [11] proposed real-time implementation of a fuzzy vault for a resource-constrained embedded system. The proposed technique of chaff generation was less vulnerable to brute force attacks as it uniformly divided the chaff generation space from the addition of first chaff point until the last required chaff point to the fuzzy vault set. The proposed technique is based on a circle packing mathematical algorithm and is less compute intensive than Clancy's approach. Khalil-Hani et al. [10] reduced the overall complexity by eliminating the need for complex square and square root computations done during chaff point addition. The authors also proposed a hardware implementation of the fuzzy vault scheme in embedded processors like the Altria Stratix II FPGA development board. The results represent a remarkable time reduction during chaff point addition as compared to Clancy's chaff generation. Nguyen et al. [16,17] proposed an improved chaff generation scheme that further reduced chaff generation time and reduced complexity compared to Khalil-Hani et al. [10,11]. The authors reduced the number of comparisons required to add and qualify a new candidate point

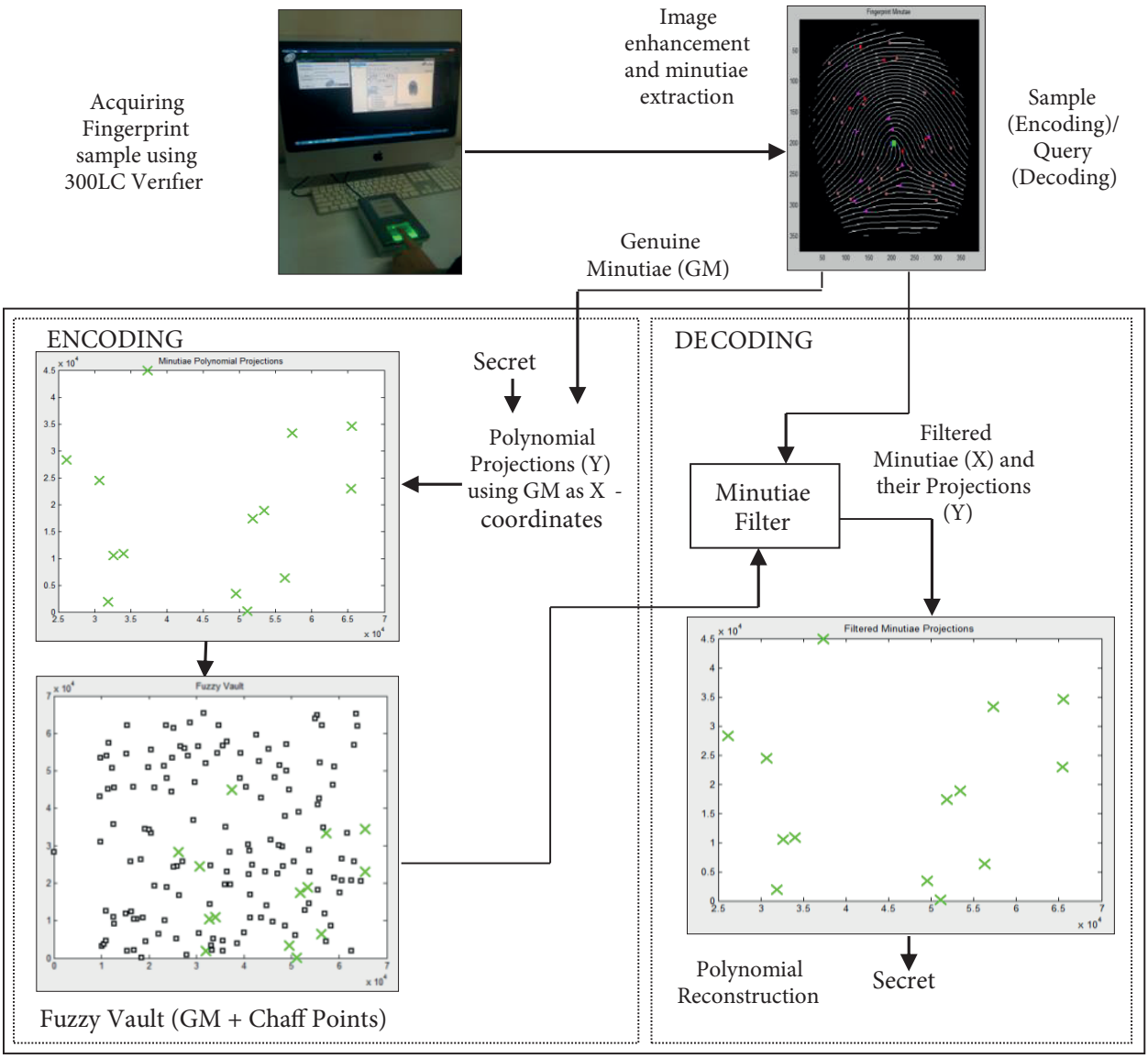


Figure 1. Fuzzy vault encoding and decoding.

as a chaff point. The candidate points in the whole process were also reduced to a greater extent as compared to Khalil-Hani et al. [10,11]. Benhammadi and Bey [18] proposed a hybrid approach of hardening fingerprint vault by combining user generated password fuzzy vault with the transformed minutiae pairwise features to provide diversity and security to fingerprint minutiae templates. The authors have generated 1024 random minutiae pairwise chaff points. The results show that the system takes less time to authenticate the user as compared to Nandakumar et al. [12] with an acceptable level of security. Nguyen et al. [19] proposed a nonrandom chaff point generator in which chaff points are generated using continuous hashing and linear projection. Chaff points are considered as signature for the combination of biometric template and key. Thus any modification with the original vault can be detected and blend substitution attacks can be prevented. Nguyen et al. [20] proposed a fingerprint fuzzy vault for distorted fingerprint images using ridge features information. In order to hide genuine ridge features, the authors have proposed an algorithm for generating chaff ridge features.

Most of the work focuses on reduction in fuzzy vault implementation time, so that the technique can be adopted for real-world applications. The next section discusses the proposed approach in detail, describing the methodology adopted to reduce the chaff addition time, which leads to a reduction in overall fuzzy vault implementation time.

4. Proposed chaff generation methodology

The proposed chaff generation approach modifies the chaff generation process for the base algorithms of Clancy et al. [8] and Nguyen et al. [16,17]. Both the algorithms compute Euclidean distance for distance comparisons between the selected point and a genuine minutiae point. However, the Euclidean distance computations are complex and more time consuming as they require the square and square root computations. The proposed approach makes use of Manhattan distance instead of Euclidean distance, which leads to a significant reduction in the chaff generation time. The modified approach is less compute intensive as it eliminates the need for complex square and square root operations. The approach works well as the absolute distance comparisons are not required for classifying any candidate chaff point as a valid chaff point and for this purpose only relative distance comparisons may work based upon an optimized threshold value. The equivalence established between Euclidean distance and Manhattan distance is described below, where $D_{(Eucl)}$ is the Euclidean distance and $D_{(Man)}$ represents the Manhattan distance and $\Delta x = (x - x_1)$ and $\Delta y = (y - y_1)$ for two points (x, y) and (x_1, y_1),

$$D_{(Eucl)} = \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{1}$$

$$D_{(Man)} = (|\Delta x| + |\Delta y|) \tag{2}$$

$$(\Delta x)^2 + (\Delta y)^2 = (\Delta x)^2 + (\Delta y)^2$$

Add $2|\Delta x \Delta y|$ to R.H.S.

$$(\Delta x)^2 + (\Delta y)^2 \leq (\Delta x)^2 + 2|\Delta x \Delta y| + (\Delta y)^2$$

So,

$$(\Delta x)^2 + (\Delta y)^2 \leq (|\Delta x| + |\Delta y|)^2 \tag{3}$$

From (1), (2), and (3)

$$(D_{(Eucl)})^2 \leq (D_{(Man)})^2 \tag{4}$$

Since the square of a real number is nonnegative

$$(\Delta x)^2 - 2|\Delta x \Delta y| + (\Delta y)^2 = (|\Delta x| - |\Delta y|)^2 \geq 0 \tag{5}$$

Add $(|\Delta x| + |\Delta y|)^2$ to both sides

$$(|\Delta x| - |\Delta y|)^2 + (|\Delta x| + |\Delta y|)^2 \geq (|\Delta x| + |\Delta y|)^2 \tag{6}$$

$$(\Delta x)^2 - 2|\Delta x \Delta y| + (\Delta y)^2 + (\Delta x)^2 + 2|\Delta x \Delta y| + (\Delta y)^2 \geq (|\Delta x| + |\Delta y|)^2$$

$$2 \times ((\Delta x)^2 + (\Delta y)^2) \geq (|\Delta x| + |\Delta y|)^2 \tag{7}$$

From (5), (6), and (7)

$$2 \times (D_{(Eucl)})^2 \geq (D_{(Man)})^2 \tag{8}$$

From (7) and (8) it can be concluded that

$$D_{(Man)} = f(D_{(Eucl)})$$

$$(D_{(Eucl)})^2 \leq (D_{(Man)})^2 \&\& \; 2 \times (D_{(Eucl)})^2 \geq (D_{(Man)})^2 \tag{9}$$

Now assuming $\delta_{(Eucl)}$ as a threshold value in Euclidian distance and $\delta_{(Man)}$ as a threshold in Manhattan distance, the relation between the two threshold values can be established from (9) as

$$(\delta_{(Eucl)})^2 \leq (\delta_{(Man)})^2 \&\& \; 2 \times (\delta_{(Eucl)})^2 \geq (\delta_{(Man)})^2 \tag{10}$$

Thus the condition specifies limits (maxima and minima) on the value of δ and the floor value of average of these maxima and minima may be taken as $\delta_{(Man)}$.

Steps of the proposed approach are as follows:

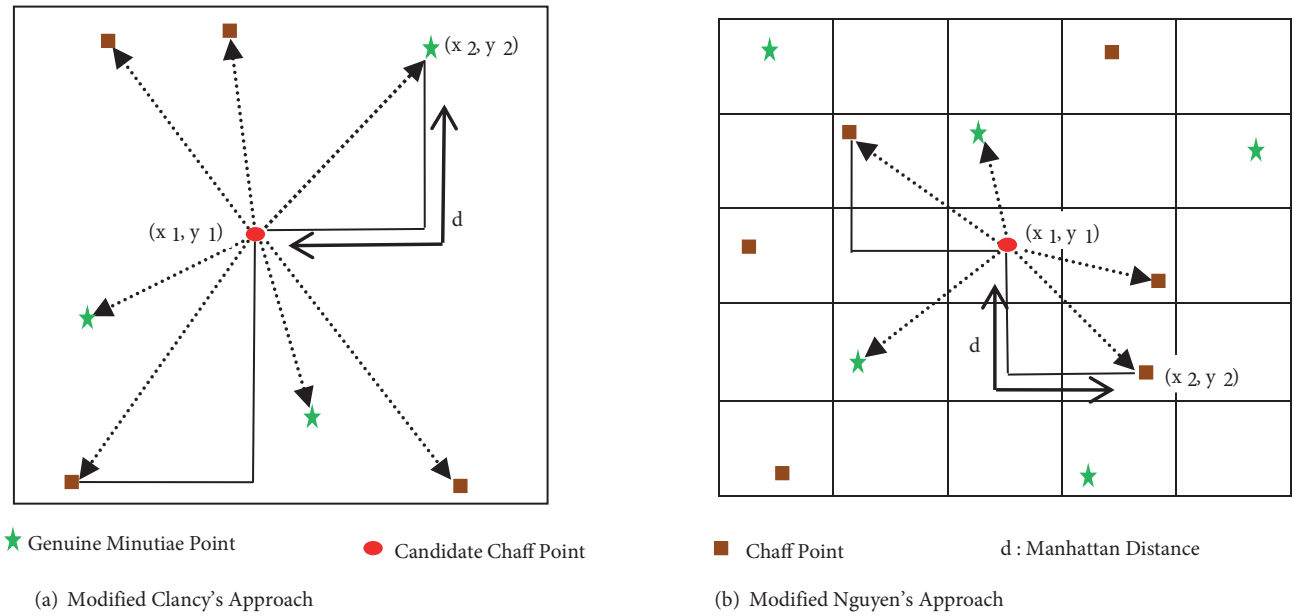
Step 1: Acquire the fingerprint image.

Step 2: Apply the pre-processing steps as binarization, thresholding, thinning etc. on it.

Step 3: Apply crossing number approach [1,9] to extract genuine minutiae points as

GM = $\{m_1, m_2 \ldots m_j\}$, where m_j represents the coordinates (x, y) for a minutiae point.

Step 4: To add chaff points in genuine minutiae set GM, Clancy's [8] and Nguyen's [16,17] approaches of chaff point addition are modified as illustrated in Figure 2 and discussed below in Sections 4.1 and 4.2, respectively.



(a) Modified Clancy's Approach

(b) Modified Nguyen's Approach

Figure 2. Illustration of modified approaches.

4.1. Modified Clancy's approach

Clancy modified the original fuzzy vault as proposed by Jules and Sudan [7], with the justification that the chaff points/random points to be added to the set of genuine points to form a vault should be placed at an appropriate distance 'd' apart from vault members. It makes use of Euclidian distance as a metric for comparing the distance between the new chaff point to be added with the existing list of points (genuine minutiae points and chaff points), which makes the approach more compute intensive. To improve upon it, the proposed modifications in the existing approach are as below:

i. Consider a vault list (VL) = GM

ii. Generate a random point (x, y) in the image. To qualify the chosen point as a chaff point it has to be δ distance apart from all the points in VL. Value of δ in the case of the modified Clancy's approach is computed using Eq. (10).

iii. Compute the Manhattan distance 'd' between the selected point $(\mathrm{x}_1, \mathrm{y}_1)$ and other existing points $(\mathrm{x}_i, \mathrm{y}_i)$ in the vault list.

$$d = |x - x_i| + |y - y_i|$$

iv. If d $\geq$ $\delta_{(Man)}$ for all points in VL

then (x, y) qualifies as a chaff point and add (x, y) to VL

else

generate new random point.

v. Repeat the steps ii, iii, and iv to generate the required number of chaff points.

4.2. Modified Nguyen's approach

Nguyen's approach gave significant time reduction compared with Clancy [8] and Khalil-Hani [11], especially when the number of minutiae points is above 20. However, this approach still makes use of Euclidean distance as a measure. The steps to be followed in the modified approach are described below:

i. Consider a vault list (VL) = GM

ii. Split fingerprint image into equispaced cells of square matrix.

iii. Choose a random cell.

iv. if (random cell contains a genuine minutiae point or chaff point)

discard it and go to step iii

else

consider the cell for candidate chaff point and select a random point $(\mathrm{x}_1, \mathrm{y}_1)$

v. Compute Manhattan distance 'd' among $(\mathrm{x}_1, \mathrm{y}_1)$ and points in its eight adjacent neighborhood cells $(\mathrm{N}_8(\mathrm{p}))$ from the VL.

vi. Using Eq. (10) $\delta_{(Man)}$, a threshold value is computed for modified Nguyen's approach

vii. if d $\geq$ $\delta_{(Man)}$ for all $N_8(p)$ (eight adjacent pixels), where p is (x_1, y_1)

then (x_1, y_1) qualifies as a chaff point and add (x_1, y_1) to VL

else

discard (x_1, y_1) as chaff point

viii. Repeat the steps from iii to vii to generate the required number of chaff points.

The proposed approach works well as by using Manhattan distance instead of Euclidean distance the chaff generation time is reduced significantly especially in worst case comparisons when for each chaff point addition the distance of all eight adjacent cells from the candidate cell needs to be compared. The modified approach is less compute intensive as it eliminates the need for complex square and square root operations. The approach works well as the absolute distance comparisons are not required and only relative distance comparisons may work and the value δ (threshold) is adjusted accordingly as per Eq. (10).

5. Proposed system design and experimental setup

Chaff point addition is a time critical task and so a dedicated hardware-based experimental setup is considered to test the time taken by modified Clancy's and modified Nguyen's approaches and the results are compared with the base algorithms also implemented on the same setup. Figure 3 represents the system block diagram. The system has been tested using the Windows 8 operating system (64-bit). Image acquisition has been done using a Verifier 300LC fingerprint scanner with a resolution of 500 dpi. A Mega Matcher SDK 4.0 (Multimodal Biometric SDK) is used for fingerprint data acquisition. Genuine minutiae points are extracted using MATLAB 7 and are then passed as an input to the dedicated hardware module through a USB port. The hardware module computes chaff points corresponding to the true minutiae and sends them back to the system with an execution time stamp through the same USB port. Since the time computation for chaff point addition is done using dedicated hardware, it is independent of the operating system overheads. An AVR UC3A364-based application board is used as dedicated hardware for chaff generation. It is a 32-bit microcontroller with 64 KB flash memory and can be easily connected to computer via a USB 2.0 cable as shown in Figure 4. The AVR UC3 A3/A4 Series is designed for very high data throughput with Hi-Speed USB device and host, SD/SDIO card, and Multi-Level-Cell (MLC) NAND flash with ECC and SDRAM interfaces. It is designed for cost-sensitive embedded applications that require low power consumption, high code density, and high performance because of which it is well suited for various biometric applications.

6. Results and analysis

The results are computed using the FVC 2002 database and live database considering all those images in which the number of extracted minutiae points is as per the requirement in Table 1. In order to make a fair comparison of the proposed approach with Clancy's approach and Nguyen's approach, their results are recomputed using the same experimental setup. For fuzzy vault implementation the secret key of 128-bit is considered with a polynomial of degree 8. Galois field (GF) (2^{16}) is used as a prime field for polynomial computations. The number of chaff points added is 10 times the genuine minutiae points. The threshold value $\delta_{(Man)}$ is computed as 15 using Eq. (10), which is considered as 12 for Clancy and Nguyen's approaches. Table 1 shows the time

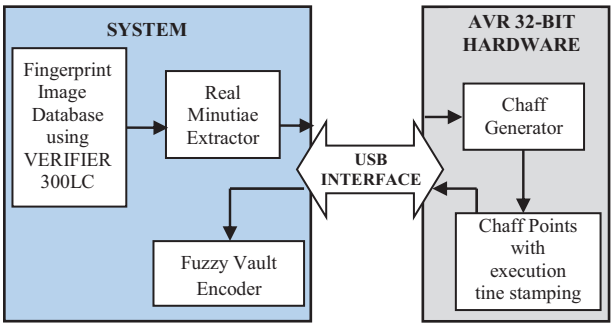


Figure 3. System block diagram.

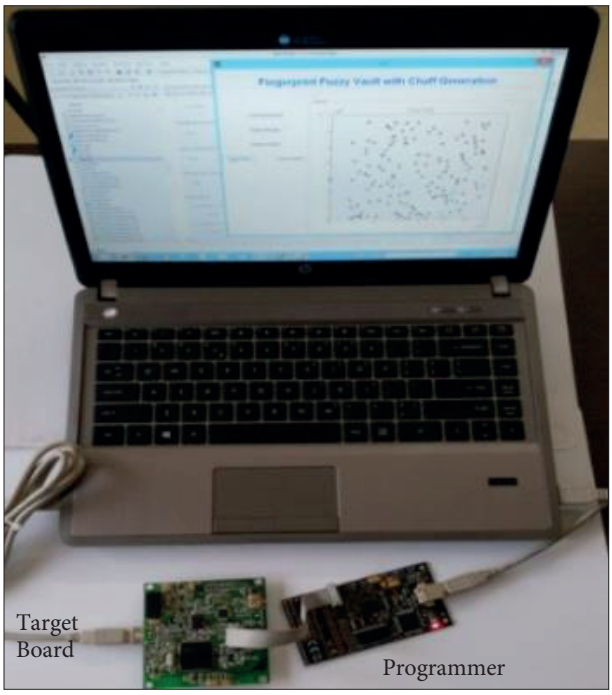


Figure 4. Dedicated hardware.

comparison of Clancy and modified Clancy's approaches for chaff point generation in ms. The average chaff point addition time is computed for both the approaches. The values from five iterations for each case are considered and the average value of time taken to add the required number of chaff points is computed. Table 2 represents computation of average chaff point generation time for Clancy's and modified Clancy's approaches for fingerprint images of four persons (F_1 to F_4) having 13 minutiae points considering 5 iterations per person.

Table 1. Time comparison of Clancy and modified Clancy's approach for chaff point generation.

Sr. no.	No. of genuine minutiae points	No. of chaff points	Average chaff point generation time (ms)	
			Clancy's algorithm	Modified Clancy's algorithm
1	9	90	241.18	3.36
2	10	100	330.7	4.42
3	12	120	511.82	6.1
4	13	130	643.44	7.80
5	14	140	915.74	10.33
6	20	200	3446.78	34.91
7	22	220	5241.15	49.50
8	23	230	8730.25	71.92
9	25	250	14,096.34	113.35

Table 3 shows the time comparison of Nguyen and modified Nguyen's approach for chaff point generation. It is observed that there is reduction in average chaff point generation time for the modified Nguyen chaff generation algorithm, which is even more significant in cases where the number of minutiae points increases beyond 20.

Figure 5a shows the execution time in ms for the generated chaff points in the case of Clancy's and modified Clancy's approaches. It has been observed that there is a significant reduction in chaff generation

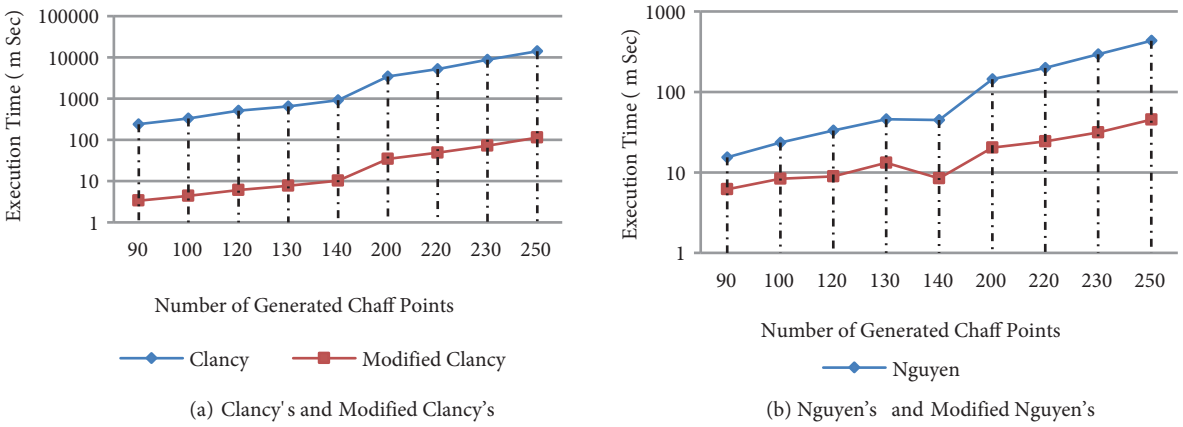
Table 2. Computation of average chaff generation time for generating 130 chaff points.

Fingerprint images for four persons (F_1 to F_4)	Chaff point generation time (ms)							
	Clancy's algorithm				Modified Clancy's algorithm			
	F_1	F_2	F_3	F_4	F_1	F_2	F_3	F_4
1st iteration	691.44	611.24	501.48	585.08	8.64	7.24	7.32	8.04
2nd iteration	756.44	537.76	549.2	740.52	8.84	7.04	6.4	10
3rd iteration	775.92	533.4	716.96	882.56	8.32	7.08	6.56	8.8
4th iteration	697.08	498.52	553.44	695.76	8.68	7.12	7.28	7.4
5th iteration	606.36	520	587.16	828.36	8	7.44	7.24	8.48
Average time (per image)	705.45	540.18	581.65	746.46	8.5	7.18	6.96	8.54
Average time for 130 chaff points	643.44				7.80			

Table 3. Time comparison of Nguyen and modified Nguyen's approach.

Sr. no.	No. of genuine minutiae	No. of chaff points	Average chaff point generation time (ms)	
			Nguyen's algorithm	Modified Nguyen's algorithm
1	9	90	15.41	6.17
2	10	100	23.45	8.34
3	12	120	33.24	8.94
4	13	130	46.04	13.17
5	14	140	44.59	8.45
6	20	200	144.85	20.39
7	22	220	199.8	24.19
8	23	230	293.06	31.31
9	25	250	432.43	45.23

time for the proposed approach. It is due to the fact that every time a new chaff point is to be added it has to be compared with all other existing set of points (genuine minutiae points and chaff points). The existing approach becomes more compute intensive with the Euclidean distance.



(a) Clancy's and Modified Clancy's (b) Nguyen's and Modified Nguyen's

Figure 5. Execution time for chaff generation approaches.

Figure 5b represents the time comparison for generated chaff points in the case of Nguyen's and modified Nguyen's approaches. The chaff generation time is reduced significantly especially when for each chaff point

addition the distance of all eight adjacent cells from the candidate cell needs to be compared and this occurs frequently in cases with greater numbers of minutiae points.

Table 4 presents the comparison of number of candidate chaff points for Clancy's, modified Clancy's, Nguyen's, and modified Nguyen's approaches. The number of candidate chaff points required to add a set number of chaff points is computed for all four approaches, namely Clancy's, modified Clancy's, Nguyen's, and modified Nguyen's. The average value of five iterations for each case is considered.

Table 4. Comparison of number of candidate chaff points for Clancy's, modified Clancy's, Nguyen's, and modified Nguyen's approaches.

Sr. no.	No. of genuine minutiae points	No. of chaff points	Average number of candidate chaff points			
			Clancy's algorithm	Modified Clancy's algorithm	Nguyen's algorithm	Modified Nguyen's algorithm
1	9	90	132	130	185	185
2	10	100	159	160	247	246
3	12	120	208	199	340	314
4	13	130	239	239	430	437
5	14	140	296	289	431	398
6	20	200	710	693	1131	1011
7	22	220	962	902	1484	1204
8	23	230	1425	1211	1979	1557
9	25	250	2080	1733	2738	2269

Figure 6 shows the comparison of number of candidate points required to generate chaff points in the case of Clancy's, modified Clancy's, Nguyen's, and modified Nguyen's approaches. The reduction in candidate chaff points in the proposed approaches can be observed easily in Table 4 and Figure 6, especially when the number of chaff points increases above 20. The reduction can be attributed to threshold equivalence and probability of number of candidate chaff points for required number of minutiae points, which may vary in most cases.

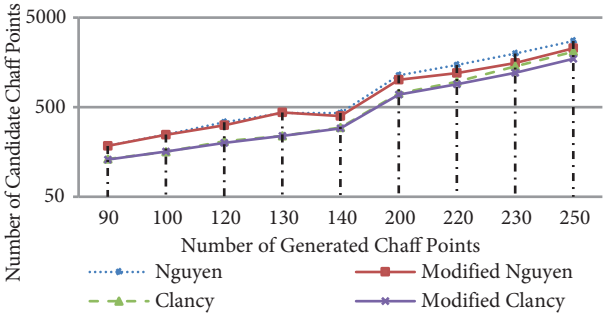


Figure 6. Number of candidate chaff points generated for required chaff in case of Clancy's, modified Clancy's, Nguyen's, and modified Nguyen's.

Table 5 represents the normalized execution time (T) for chaff point generation for Clancy's, modified Clancy's, Nguyen's, and modified Nguyen's approaches. Thus the value of T gives an accurate metric for execution time comparison. It is computed as follows:

$$T = \frac{\text{Chaff point generation time}}{\text{Candidate chaff points}}$$

Table 5. Normalized execution time for chaff point generation for Clancy's, modified Clancy's, Nguyen's, and Nguyen's approaches.

Sr. no.	No. of genuine chaff points	Normalized execution time (T) for chaff point in ms			
		Clancy's approach	Modified Clancy's approach	Nguyen's approach	Modified Nguyen's approach
1	90	1.827	0.026	0.083	0.033
2	100	2.08	0.028	0.095	0.034
3	120	2.461	0.031	0.098	0.028
4	130	2.692	0.033	0.107	0.03
5	140	3.094	0.036	0.103	0.021
6	200	4.855	0.05	0.128	0.02
7	220	5.448	0.055	0.135	0.02
8	230	6.126	0.059	0.148	0.02
9	250	6.777	0.065	0.158	0.02

7. Performance evaluation

The performance of the system is evaluated in terms of false accept rate (FAR) and false reject rate (FRR). The database used for the result analysis consists of fingerprint samples of 100 users with 2 fingerprint impressions of the same finger for each user. FAR is computed considering the first fingerprint impression of each user as the template and all the fingerprint impressions of all other users as query samples. To compute FRR the first fingerprint impression of each user is considered as template and the second one as query. Hence, the total number of genuine attempts to compute the value of FRR is 100 and imposter attempts to compute FAR are 19,800 for 100 users. For modified Clancy's approach the values of FAR and FRR are computed as 2% and 11.6%, respectively. For modified Nguyen's approach the FAR and FRR are calculated as 2% and 11%, respectively. The performance of the proposed algorithms has been increased in terms of chaff generation time. However, the performance of the system in terms of FAR and FRR remains the same as that of the baseline algorithms since the focus of the research work is on reducing the chaff generation time of the existing algorithms majorly in use while implementing fuzzy vault.

8. Conclusion

It has been observed in the experimental results that the time required for chaff generation is reduced to a greater extent in the case of modified Clancy's and modified Nguyen's approaches by using the Manhattan distance as a metric for distance comparison while generating a new chaff point. It has also been observed that the number of candidate chaff points is reduced with the increase in number of minutiae points in case of modified Clancy's and Nguyen's approaches. The approach also reduces the complex computations to simple ones. Since the proposed approach has been evaluated on a dedicated hardware system, it can be deployed in real-time application scenarios.

References

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

[2] Maltoni D, Maio D, Jain AK, Parbhakar S. Handbook of Fingerprint Recognition. 2nd ed. London, UK: Springer, 2003.

[3] Uludag U, Pankanti S, Parbhakar S, Jain AK. Biometric Cryptosystems: Issues and Challenges. In: Proceedings of IEEE 2004; 92: 948-960.

[4] Jain AK, Ross A, Uludag U. Biometric Template Security: Challenges and Solutions. In: Proceedings of 13th European Signal Processing Conference (EUSIPCO); 4 September 2005. Antalya, Turkey: IEEE. pp. 1-4.

[5] Ratha NK, Connell JH, Bolle RM. Enhancing security and privacy in biometrics-based authentication systems. IBM Syst J 2001; 40: 614-634.

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

[7] 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.

[8] 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.

[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] 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.

[11] 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.

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

[13] 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.

[14] Chung Y, Moon D, Lee S, Jung S, Kim T, Ahn D. Automatic alignment of fingerprint features for fuzzy fingerprint Vault. In: International Conference on Information Security and Cryptology; December 2005; Berlin, Germany: Springer. pp. 358-369.

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

[16] 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.

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

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

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

[20] 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.