Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 11 (2016) pp. 001-010
© MIR Labs, www.mirlabs.net/jias/index.html

Chaff Point Generation by Squares Method Using Composite Representation in Fingerprint Fuzzy Vault

Hachemi Nabil Dellys, Noussaiba Benadjimi, Meriem Romaissa Boubakeur
Laboratoire de Méthodes de Conception de Systèmes (LMCS).
Ecole nationale Supérieure en Informatique (ESI), Oued smar.
Algiers, Algeria.
{h_dellys,an_benadjimi,am_boubakeur}@esi.dz

Layth Sliman
Ecole d'Ingénieur Informatique et technologies du numérique (EFREI)
32 Rue de la république Villejuif, 94800, Paris, France
Layth.sliman@efrei.fr

Fathelalem Ali
Dept of Management and Information Systems.
Meio University, 1220-1 Bimata.
Okinawa, Japan.
ali@meio-u.ac.jp

***Abstract*— In the literature, several abscissae generation methods of chaff points in fingerprint fuzzy vault exist. In this paper, we make an experimental comparison between squares method and threshold methods. The experimental results show that the squares method is far better than methods based on threshold. But minutiae representation in squares method uses 2D representation while threshold methods are represented by composite representation. We proposed to implements squares methods using composite representation and made same experiments which showed less gain of time but more robustness and performance.**

Keywords- Fingeprint; Fuzzy Vault; Chaff-points; Minutiae representation; Abscissae generation.

I. INTRODUCTION

Fuzzy vault is an error tolerate data securing method using public information proposed first by Juels and Sudan [1] in 2002. The unstable nature of the different biometrics, due to the image captures, makes it difficult to obtain a single biometric template. A certain error threshold is tolerated during recognition, but this error threshold does not allow the use of conventional encryption techniques to secure templates. The fuzzy vault method has been applied in biometrics to secure templates, hence the use of new error-tolerant techniques such as Fuzzy Vault. In this paper, we focus in fingerprint fuzzy vault [2, 3, 4, et 5].

Fuzzy vault is composed of two phases: data encoding which consists of encrypting a secret key generated randomly, or imposed by the user, with fingerprint features and chaff points [7, 11, 18, 24 et 26]. The second phase is for data decoding, which consists to reconstruct the secret key from the Vault [19, 23, 29 et 30].

In this paper, the process of chaff points' abscissae generation in fingerprint fuzzy vault is studied. Comparative experiments are performed between threshold methods [26 et 27] and squares method [20 et 29] as implemented in literature. The purpose of this work is to determine which chaff-points generation method is the best to use in terms of the chaff-point number actually generated and response time, and which features representation is the best to use with it. In section 2, we describe the two phases of the fingerprint fuzzy vault process. We present different fingerprint feature representation used in fuzzy vault in section 3. Section 4 describes the process of chaff points generation before carrying out comparison experiments on three most cited in literature (threshold methods and square method) in section 4. Finally, we present our proposal to adapt standard squares method to generate chaff-points with composite representation and make comparison experiments with thresholds methods.

II. FUZZY VAULT APPLIED ON FINGERPRINT

The principle of the fuzzy vault on fingerprint consist of two phases:

1. *Encoding the Vault.* The encoding consists to encrypt a secret key, generated randomly or imposed by the user, with fingerprint characteristics and chaff points.
2. *Decoding the Vault.* Decoding consists to regenerate the key from the vault.

Each phase consists in several stages. In the following, we describe these stages and we give details of the main proposals in the literature that worked on each one.

A. Fuzzy Vault Encode Phase

There are five stages in the encoding phase of fuzzy vault, which are detailed below:

Stage 1: Compute the Characteristic Polynomial. Characteristic polynomials are generated from a secret key (S) [6, 7, 8 et 33] or its hash [9 et 10]. The latter is divided into k parts of the same size, Si, i=1...k, and integrated into the polynomial coefficients. Multiple polynomial generation can also be used when the number of features extracted from fingerprint is small and does not allow the interpolation of a k-degree characteristic polynomial. This method divides the secret key into m sub-secret keys, from which m k-degree polynomials are generated.

Stage 2: Fingerprint Feature Representation. There is a plethora of features that can be extracted from a fingerprint. Only two of these features verify biometrics constraints: minutiae and descriptor representations. After, we summarize the main proposals of these two types of representation**.** [9, 10, 11, 12, 13, 29 et 30].

Stage 3: Chaff Points Modelling. The characteristics extracted from fingerprint are represented by a vector of coordinates. However, fuzzy vault requires their representations in scalar form [14]. Consequently, the modeling by concatenation is used. In this modelling, fingerprint characteristics attribute are concatenated to form the encoding units [7, 12, 14, 15 et 27].

Stage 4: Chaff Points Generation. The Chaff points are represented in two coordinates: abscissa *'c'* and ordinate *'d'*. All Chaff points are represented as ***CHAFF***= $\{(\boldsymbol{ci}, \boldsymbol{di}); \boldsymbol{di} \neq \boldsymbol{P}(\boldsymbol{uci})\}_{\boldsymbol{i=1}}^{\boldsymbol{R}}$, where *uci* are the encoding units obtained from *ci*. [20, 23, 24, 26, 27 et 29] Generation of chaff points require two process: abscissae generation and ordinate generation.

Stage 5: Vault Construction and Storage. In this stage, chaff and authentic points are combined to form a vault. The purpose is to hide all the encoding units by adding chaff points [16].

B. Fuzzy Vault Decoding

This phase includes three main stages which are: the points' alignment, correspondence set determination and polynomial reconstruction.

Stage 1: Points Alignment. There are two techniques to align the fingerprints [11, 17 et 18]**:**

1. *Fingerprint Pre-alignment.* Pre-alignment methods are based on fingerprint information extraction.
   a. *Reference minutiae/point.* This technique denotes a reference minutia against which other minutiae are represented using polar coordinates [13].
   b. *Helper data.* Helper data are a public information, it carries enough information to perform the alignment without revealing any information about the original fingerprint [6, 25, et 26]. Extraction of helper data is based on the principle of *Orientation Field Flow Curves (OFFC)* [18]. The *OFFC* representation is robust against noise generated by fingerprint acquisitions, including the Islands and ridges cuts [6].
2. *Fingerprint Auto-alignment.* Automatic alignment uses invariant representations to translation and rotation of fingerprint [16]. Thus, no pre-alignment process is used [23]. Amongst these techniques are found:
   a. *Geometric hashing.* Alignment by the geometric hash table is considered to be one of the most accurate. But requires a large storage space, and importantly computing time [1, 12, 15 et 33].
   b. *Minutia structure.* These representations which described in the second stage of encoding phase are invariant to translations and rotations [16].

Fingerprint Free-Alignment. This method has the advantage that doesn't use alignment [21]. It uses the local texture around the minutia which is invariant to transformations of fingerprints that occurred during their acquisitions.

Stage 2: Determination of Correspondence Set. The determination of correspondence set consist to identify the authentic points in the vault stored with those extracted from the aligned template in the query. This process creates a set of correspondence points that are authentic in their majority with a legitimate user [11, 17, 18, 32 et 34].

Stage 3: Secret Polynomial Reconstruction. The secret polynomial reconstruction is generated from correspondence set obtained during the first stage. If the polynomial is properly constructed, the secret key can be recovered, and the user is successfully authenticated [8, 10, 22, 32 et 33].

III. Fingerprint feature representaion

There are several features that can be extracted from fingerprint. Two of them check constraints of biometrics, the minutiae representation and representation by descriptors. Here below is a summary of the main proposals in these two types of representation:

a. Minutiae representation as singular points

In this representation, minutiae are described by a set of coordinates, so we find:

i. 2D representation: Minutiae is described by its Cartesian coordinates (x, y). [6].
ii. 3D representation: Minutiae is described by its Cartesian coordinates (x, y) and orientation θ [9 et 20].
iii. 4D representation: Minutiae is described by its Cartesian coordinates (x, y), orientation θ and its type (endpoint / fork). [10 et 21].

This representation is easy to implement, but generates many errors of matching in authentication phase.

b. Minutiae representation as part of a structure:

Minutiae are described by its local or global structure, based on its geometry. So we find:

i. Minutiae representation by local structure: this representation is used to describe minutiae compared to its neighbors. The main proposals in this representation are**:**

♦ The five nearest neighbors structure: this representation describes minutia (*m*) compared with its closest five minutiae (*mi*), in the form of$\{(\boldsymbol{r_i},\boldsymbol{\theta_i})\}_{\boldsymbol{i=1}}^{\boldsymbol{5}}$. The scalars *ri* are the Euclidean distances between *m* and *mi*, and θi, are angles formed between *m* and *mi* [11].

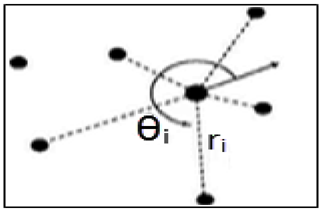

Figure 1.Five nearest neighbors representation.

♦ Vorono ï neighbors: this representation is very similar to the previous one, except that neighbors are determined according to Voronoi diagram [11].

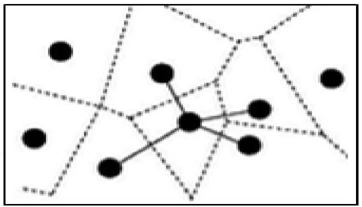

Figure 2. Vorono ïrepresentation.

♦ Composite representation: the composite representation is also a representation of minutiae compared their neighbors [17]. The minutia (*mi*) compared to (*mj*) is described by a 3-tuple ($\boldsymbol{d_{i-j}}$,$\boldsymbol{\varphi_{i-j}}$,$\boldsymbol{\theta_{i-j}}$) where :

▪$\boldsymbol{d_{i-j}}$: is the Euclidian distance between *mi* and *mj* ;

▪$\boldsymbol{\varphi_{i-j}}$: is the difference between orientation angle of *mi* and *mj*;

▪$\boldsymbol{\theta_{i-j}}$, is the counter-clockwise angle between the orientation of *mi* and direction from *mi* to *mj*.

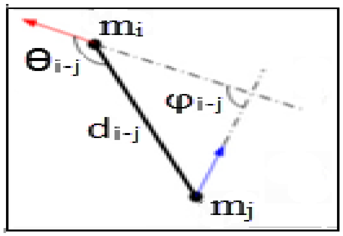

Figure 3. Composite representation

ii. Representation according to the overall structure: this representation of minutiae is based on the fingerprint overall geometry.

The main proposals concerning representation in triangle based [11], where fingerprint template is described as a set of triples of distinct minutiae. Each triplet, (p1, p2, p3), trace a triangle in counterclockwise direction, represented by ((r_{1-2}, θ_{1-2}), (r_{2-3}, θ_{2-3}), (r_{3-1}, θ_{3-1})), where:

- r_{ij} : is the distance between p_i and p_j ;
- θ_{ijis} :the angle of p_i to p_j.

This representation generate less matching errors, but requires more compute.

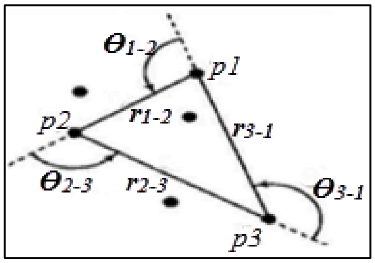

Figure 4. Triangle representation.

c. Descriptors based representation :

This type of representation describes minutiae by a descriptor computed from fingerprint image. The main proposals in this representation are descriptive vector [12] and points around the nucleus [22]. This representation requires fewer compute, but generate more matching errors than the representation as part of a structure.

IV. Chaff points generation

Chaff points are data, similar to the fingerprint minutiae representation, generated pseudo-randomly in order to hide actual minutiae. Chaff-points generation is subject to two constraints [23]:

i. A Chaff point must not be very close to an actual point;
ii. Two chaff points should not be very close. Otherwise, the Chaff points will be easily detected by the attacker.

A. Number of generated Chaff-points.

The greater is number of Chaff-points, the greater is security level. However, a great number of chaff-points affects other aspects, including:

i. Storage memory space of the Vault will be more important [24].
ii. The freedom degree of the Chaff point increases significantly, and will be easily detected by an attacker [25].
iii. System response time increases significantly due to the computing complexity [26].

A compromise between security, storage space and response time of the system is important to fix the number of chaff point must be generated. In several works [24, 9, 20 et 27], the Chaff-points number is determined empirically after several tests. The authors of [28] argue that a good security level of biometric system requires a Chaff-points number greater than authentic point’s number.

B. Chaff points generation methods

In this section, we explain how Chaff-points are generated. Chaff points are represented by two coordinates: abscissa (*c*) and ordinate (*d*). Abscissae and ordinates are generated by different methods. We present the main steps as following:

1. **Chaff points abscissae generation**

The Chaff points abscissae are generated according to two strategies:

i. Generation based on the spacing between points:

This method generates R distinct points randomly according to two strategies [26 et 27]:

a. The first strategy generate points using Euclidean distance. Actual and chaff points are separate at least by threshold (δ) *(Fig.1-a)*.

b. The second strategy takes into account the distribution of the authentic points. Authentic points can be fairly close together and formed a points mass. An attacker can distinguish this mass to find an actual points *(Fig.2 - b)*. To resolve this problem, two threshold are used. The first distance (*δ1*), separate chaff point from others.

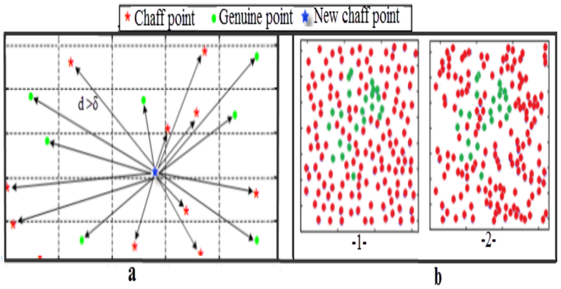

Figure 5. Chaff points abscissae generation with threshold methods. a- one threshold. b- two threshold.

In both strategies, threshold are chosen according to distance average between minutiae and number of chaff points want to generate, the experiments give the best threshold values.

ii. Use of geometric forms:

This strategy is based on geometric constraints to generate Chaff points:

a. Squares based algorithms. This method detailed in [20 and 29] centres each point of the vault by a fixed size square, so the squares will never overlap, but they can be tangent *(Fig.2-a)*.

b. Algorithms based on cell image segmentation. A second method, detailed in [30], divides the image into cells, in the form of fixed-size squares, and considers only one point (Chaff or authentic) per square *(Fig 2-b)*.

2. **Chaff points ordinates generation**

Ordinates generation of Chaff points $(\{di\}_{i=1}^{R},)$ is ensured by one of the following strategies:

i. Ordinates are generated randomly after checking the constraint saying that (c_i, d_i) is not on the secret polynomial [6, 10, 18, 19, 21 et 32].

ii. Ordinates are generated by taking $d_i = P(u_i) + \alpha$, where P is a secret polynomial, α is a real generated randomly and u_i is locking unit obtained from c_i [31].

The ordinates chaff point (d_i) is first determined using one of the methods listed above. Then, misleading polynomials are integrated into the Vault. Finally, Chaff points abscissae (c_i) are re-evaluate by these misleading polynomials [32].

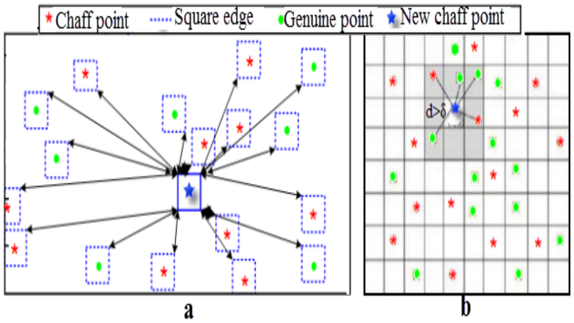

Figure 6. Chaff points abscissae generation. a- squares method. b- cells method.

V. Comparison of existing abscissae chaff-points generation methods

Four methods are proposed to generate abscissae of chaff-points: One threshold method [ref], two thresholds method [ref], cells method [ref] and squares method [ref].

Since Cells method is a particular case of squares methods, we make comparison only between thresholds and squares methods

EXPIRIMENTS

In this section, we make experimental comparison between abscissae generation methods. The strategies based on the spacing between points and those based on squares are compared according to effective number of chaff points really generated and response time. The experiment are launched in DB2_A, FVC2006 database with 1680 fingerprint and average of 38.15 minutiae per fingerprint [ref].

The experiment was run using a software platform based on Java and Matlab, and a computer hardware based on an Intel Core 2 Duo processor and 8 GB of Random memory.

A. *Number of chaff points generated*

The first experiment concerns the chaff points really generated by each methods compared to chaff points want to generate (according storage capacity, security degree, etc.). In Figure 3, we note that the squares curve increase linearly, this is means that squares methods generate effectively the number of chaff points needed to generate.

Otherwise, the one threshold curve increase linearly until around 200 chaff points generates. The curve stabilizes after around 180 chaff point whatever the needed chaff points increased.

The same behavior are noted for the two threshold curve around 350 chaff points. This augmentation of maximum chaff points generated in two threshold methods can be explained by the second threshold which is smallest than the first.

B. *Chaff-points generation time*

The second experiment concerns generation time of chaff points for each methods. In Figure 4, we note that the generation time in squares methods remains insignificant around 0.03 second for 1000 chaff-points generated.

However, the one threshold curve becomes insignificant when number of chaff points generated is lower than 150. When chaff points generated are between 150 and 200, one threshold curve increases exponentially until 200 second, then this curve oscillates between 120s and 210s.

This oscillation is explained by the relatively fixed number of chaff points really generated after 200 chaff-points.

The same behavior is observed for the two threshold curve after 290 chaff points generated, but the oscillation is greater and maximum generation time is noted as 530s. This is explained by the second threshold that must be computed in this case.

As shown in Figure 5, the curves of one threshold and two threshold have the same behavior. First, gain is very small around 1, then, curves increase exponentially by 150 chaff points for one threshold and 290 for two threshold. The maximum is obtained with 220 chaff points for one threshold with gain of 24000, and 400 chaff points for two threshold with gain of 26400. Subsequently, the curves decrease oscillating according curves of Figure 4, because the generation of chaff points stopped after around 220 for one threshold and 350 for two threshold.

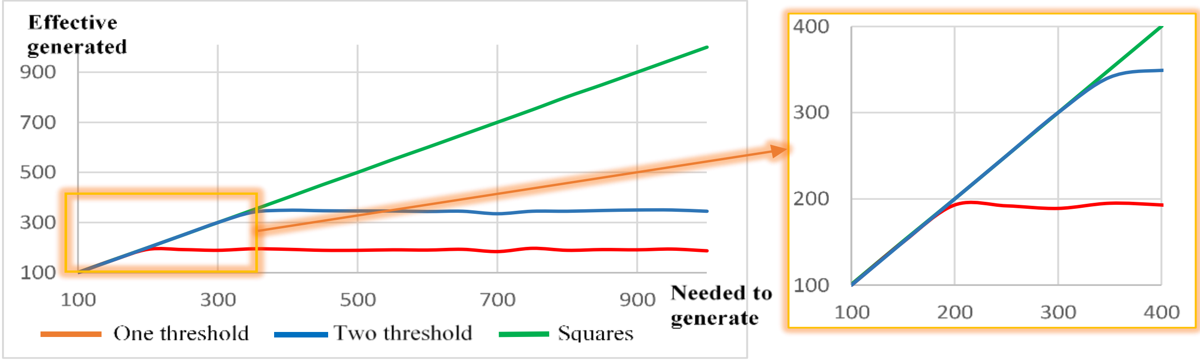

Figure 7. Chaff points really generated compared to chaff points needed to generate.

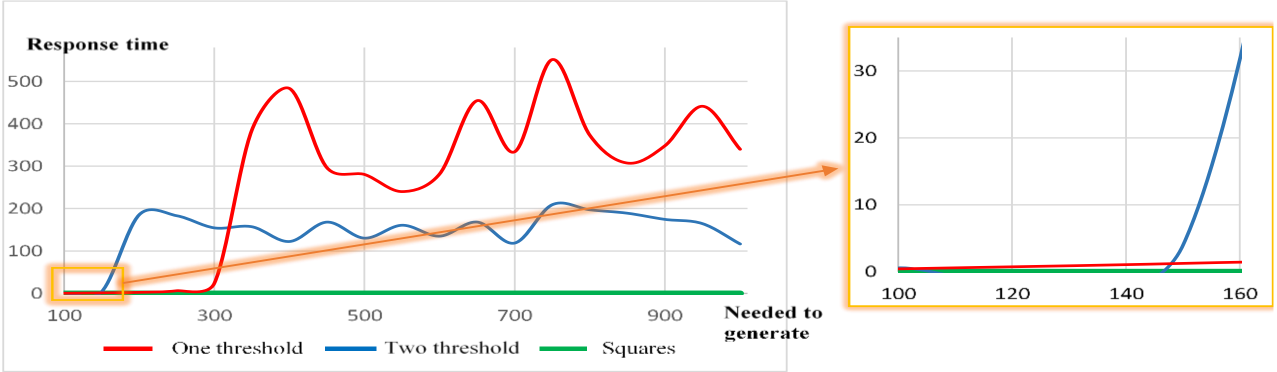

Figure 8. Generation time of chaff points.

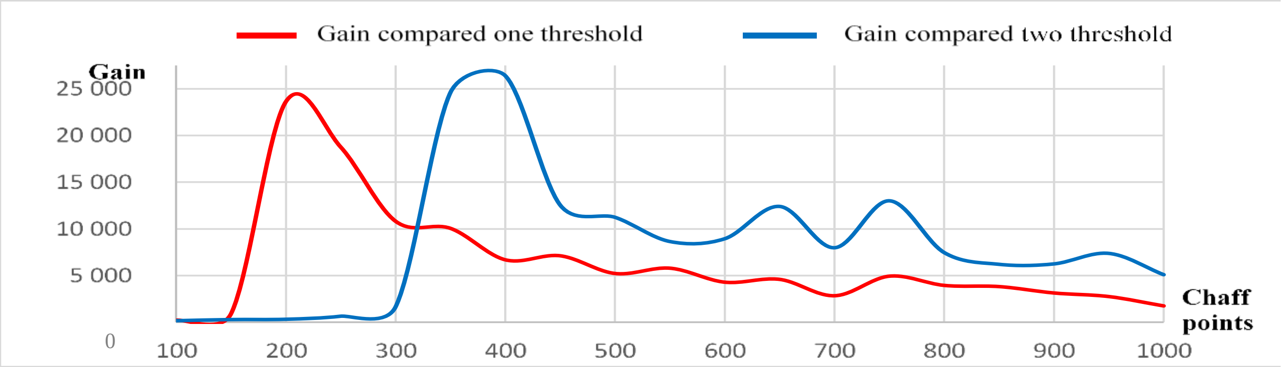

Figure 9. Gain of time between square method using 2D representation and threshold methods.

VI. OUR PROPOSAL

Several methods proposed in literature use singular point representation, but this representation is not robust have many inconveniences, the main of them need alignment Dellys et al [19]. As present by, alignment stages require great response time. In add:

- *Geometric hashing alignment:* Very costly in computing time and storage space;

- Helper data : Lose information details in the process of authentic points distribution and take relatively too much computing time;
- *Reference minutiae/point*: Poor performance of Reference minutiae extraction algorithms makes this method not much effective.

In this paper, we propose to use composite representation and squares method to chaff-points generation. Despite the fact that composite representation increases computing time, robustness and the non-use of points-alignment stage and its high computing time make this representation more interesting than singular point representation. Concerning chaff-points generation, the squares method is better method according comparison made in section IV.

EXPIRIMENTS

We make experiments comparisons with the same environment that section V.

A. Number of chaff points generated

Chaff points actually generated by each method compared to chaff points need to generate is illustrated in Figure 10, we note that the squares curve increase linearly, this is means that squares methods always generate chaff points needed to generate even with composite representation.

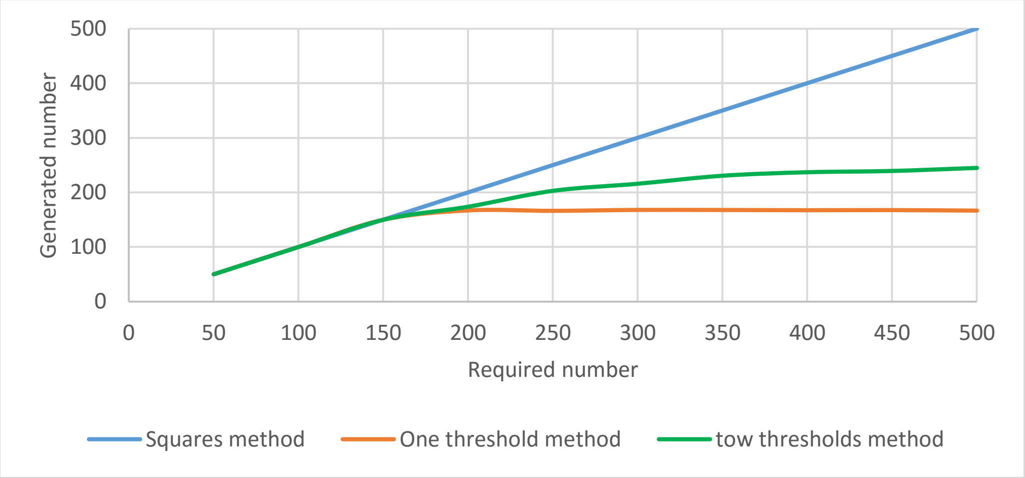

Figure 10.Number of generated Chaff points based on the requested number.

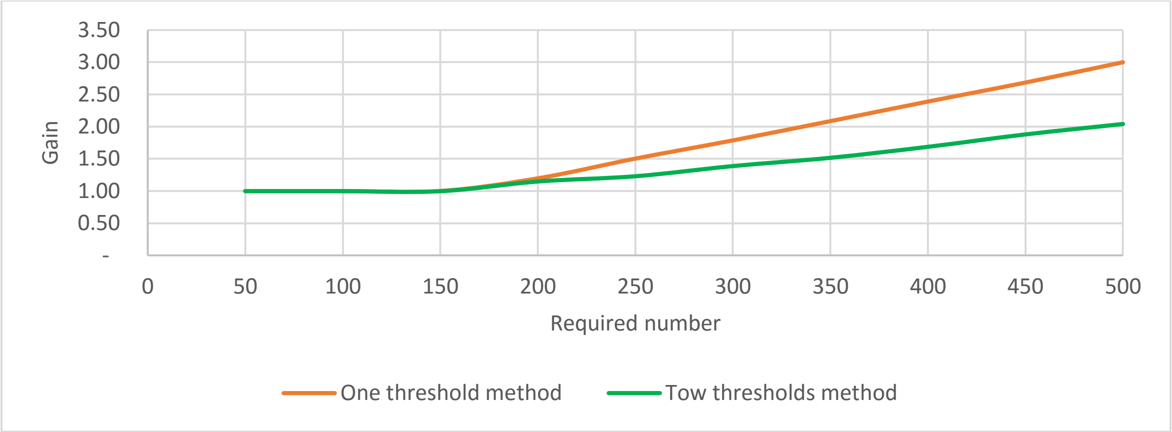

Figure 11.Gain in the generated number.

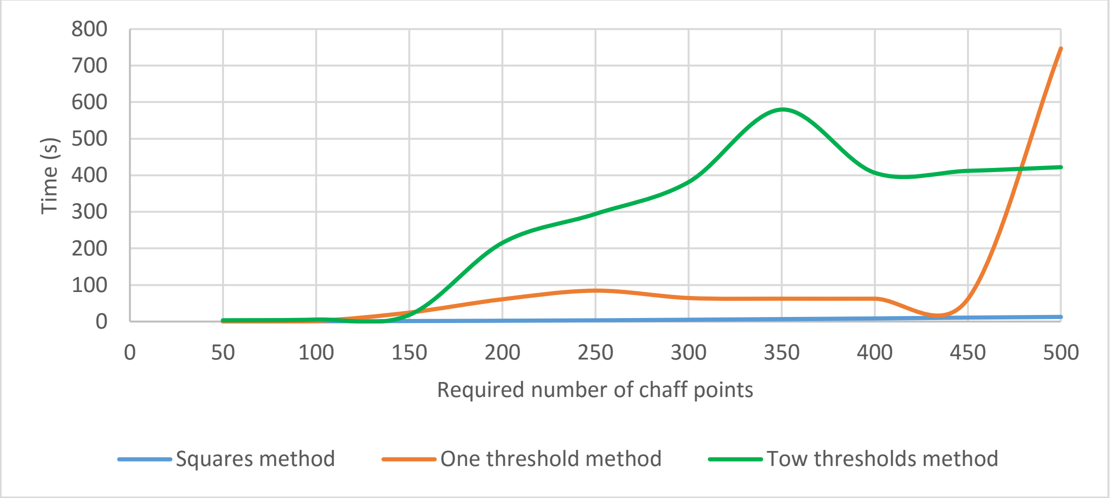

Figure 12.Generation time of Chaff

Otherwise, the one threshold and two thresholds curves increase linearly until a certain value and stabilizes after around 180 chaff-points for one threshold method and around 250 chaff-points for two thresholds, whatever the needed chaff points increased.

Then, as shown in figure 11, number of chaff-points generated using squares method is three times greater than one threshold method and two times greater than two thresholds methods.

B. *Chaff-points generation time*

Figure 12 illustrates generation time of chaff points. We note that the generation time in squares methods using composite representation remains still insignificant around 1 second for 500 chaffs points generated.

However, the one threshold and two thresholds curves become insignificant when the number of chaff points generated is lower than 150. When chaff points generated are between 150 and 450, one threshold curve increases around 100 second, then this curve increase exponentially after 450 chaff-points.

The two thresholds curve increase exponentially after 150 chaff-points generated and maximum generation time is noted as 590s.

As shown in Figure 13, gain of time achieved 119 between squares method and two thresholds method, and achieved 60 between squares method and one threshold method.

VII. CONCLUSION AND PERSPECTIVES

In this paper, we first make an experimental comparison between the abscissae generation of chaff points in fingerprint fuzzy vault with singular point representation. As shown in table 7, squares method generates exactly the needed number of chaff points, while threshold methods stagnates around

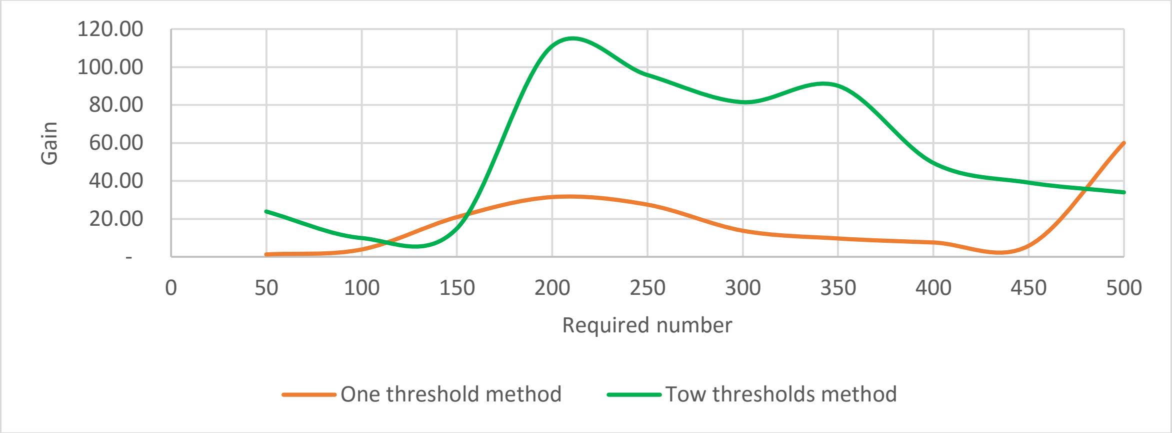

Figure 13.Gain on time.

200 chaff points for one threshold method and 350 for two thresholds method.

As shown in Figure 8, generation time in squares method still insignificant around 0.03s, while in threshold methods generation time increase exponentially after 150 chaff points generated for one threshold and 290 for two thresholds. When a number of chaff points achieved a maximum, the generation of chaff points stopped which causes an oscillation of time generation. The gain of time is very great between squares method and threshold methods with singular point representation, as shown in Figure 9. These comparisons shown that squares methods are the best methods of chaff-points generation when singular point representation is used.

However, singular point representation is not robust and it is vulnerable against attacks. Therefore, we use a composite representation that is more robust. Same experiments are achieved to compare chaff-points generation methods with composite representation. The chaff-points generated still same as needed chaff-points, when one threshold and two thresholds methods stagnates respectively around 180 and 250 chaff-points actually generated. Gain in the chaff-points actually generated is three times greater in squares method than one threshold method and two times greater in squares method than two thresholds method. Response time remains small around 1s when response time for one and two thresholds methods increase exponentially after 150 chaff-points generated. Response time for squares method are greater than squares method using singular points representation, but composite representation not need point alignment stages, which is stage that require most computing time. Gain of time in squares method achieved 60 time compared to one threshold method and 120 time compared two thresholds method.

We conclude that squares method using composite representation are best choice in stages of feature representation and chaff-points generation.

This comparison is a step of several experiments that will help to determine the most efficient method to use in each fuzzy vault stages, in terms of performances, security level, response time and storage.

References

[1] Juels, A. M. Sudan ‘A fuzzy vault scheme’. In Proceedings of the 2002 IEEE International Symposium on Information Theory. 2002.

[2] Nguyen T.H., Wang Y. Ha Y. and Li R., *Performance and security-enhanced fuzzy vault scheme based on ridge features for distorted fingerprints*, IET Biometrics, Volume 4, Issue 1, p.29–39, March 2015.

[3] Brindha V. E. Finger Knuckle Print as Unimodal Fuzzy Vault Implementation, Procedia Computer Science, Volume 47, Pages 205–213. 2015

[4] Tams B., Unlinkable Minutiae-Based Fuzzy Vault for Multiple Fingerprints, IET Biometric, 2015.

[5] Bringer J., Favre M., Pelle C., Saxce H., Fuzzy vault and template-level fusion applied to a binary fingerprint representation, Biometrics Special Interest Group (BIOSIG), 2014.

[6] Uludag, U. and A. Jain. *Securing fingerprint template: Fuzzy vault with helper data. In Computer Vision and Pattern Recognition Workshop*, 2006. CVPRW'06. Conference on. IEEE. 2006.

[7] Lee, S., et al. Analysis of tradeoffs among verification accuracy, memory consumption, and execution time in the GH-based fuzzy fingerprint vault. in Security Technology, 2008. SECTECH'08. International Conference on. IEEE. 2008.

[8] Choi, W., et al., Apparatus and method for polynomial reconstruction in fuzzy vault system, Google Patents, 2012.

[9] Khalil-Hani, M., M.N. Marsono, and R. Bakhteri, *Biometric encryption based on a fuzzy vault scheme with a fast chaff generation algorithm.* Future Generation Computer Systems. **29**(3): p. 800-810, 2013.

[10] Moon, D., et al. Implementation of automatic fuzzy fingerprint vault. in Machine Learning and Cybernetics, 2008 International Conference on. IEEE, 2008.

[11] Jeffers, J. and A. Arakala. Minutiae-based structures for a fuzzy vault. in Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the. IEEE, 2006.

[12] AlTarawneh, M., W. Woo, and S. Dlay. Fuzzy vault crypto biometric key based on fingerprint vector features. in Communication Systems, Networks and Digital Signal Processing, 2008. CNSDSP 2008. 6th International Symposium on. IEEE, 2008.

[13] Harmer, K., et al. Fuzzy Vault Fingerprint Smartcard Implementation Using an Orientation-Based Feature Vector. In BLISS. 2008.

[14] Park, U., S. Pankanti, and A. Jain. *Fingerprint verification using SIFT features*. in *SPIE Defense and Security Symposium*. International Society for Optics and Photonics,2008.

[15] Khachatryan, G.J., Aram; Khasikyan, Hovik, *Alignment-free fuzzy vault scheme for fingerprints.* 2013.

[16] Nagar, A., K. Nandakumar, and A.K. Jain. *Securing fingerprint template: Fuzzy vault with minutiae descriptors*. in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008.

[17] Xi, K. and J. Hu. *Biometric mobile template protection: a composite feature based fingerprint fuzzy vault*. in *Communications, 2009. ICC'09. IEEE International Conference on*. IEEE, 2009.

[18] Nandakumar, K., A.K. Jain, and S. Pankanti, *Fingerprint-based fuzzy vault: Implementation and performance.* Information Forensics and Security, IEEE Transactions on, **2**(4): p. 744-757. 2007.

[19] Dellys H.N., Benadjimi N., Boubakeur M.R., Sliman L., Artabaz S., Benatchba K., Koudil M. *A Critical Comparison of Fingerprint Fuzzy Vault Techniques*, Advance in Visual Informatics, 4th International Visual Informatics Conference, IVIC 2015, Bangi, Malaysia, November 17-19, 2015.

[20] Khalil-Hani, M. and R. Bakhteri. *Securing cryptographic key with fuzzy vault based on a new chaff generation method*. in *High Performance Computing and Simulation (HPCS), 2010 International Conference on*. 2010. IEEE.

[21] Lee, S., et al. *Memory-Efficient Fuzzy Fingerprint Vault based on the Geometric Hashing*. in *Information Security and Assurance, 2008. ISA 2008. International Conference on*. 2008. IEEE.

[22] Uludag, U., Pankanti S., and Jain A.K.. *Fuzzy vault for fingerprints*. in *Audio-and Video-Based Biometric Person Authentication*. Springer. 2005.

[23] Alibeigi, E., M.T. Rizi, and P. Behnamfar. *Pipelined minutiae extraction from fingerprint images*. in *Electrical and Computer Engineering, 2009. CCECE'09. Canadian Conference on*. 2009. IEEE.

[24] Rajkumar, R. and K. Hemachandran, *A secondary fingerprint enhancement and minutiae extraction.* Signal & Image Processing: An International Journal (SIPIJ) Vol. 3., 2009.

[25] Krivokuća, V., W. Abdulla, and A. Swain. *A dissection of fingerprint fuzzy vault schemes*. in *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, ACM. 2012.

[26] Sood, P. and M. Kaur. *Methods of automatic alignment of fingerprint in fuzzy vault: A review*. in *Engineering and Computational Sciences (RAECS), 2014 Recent Advances in*. 2014. IEEE.

[27] Zhou, R., et al. *Adaptive sift-based algorithm for specific fingerprint verification*. in *Hand-Based Biometrics (ICHB), 2011 International Conference on*. IEEE, 2011.

[28] Nandakumar K., Jain A.K. and Pankanti S., *Fingerprint-Based Fuzzy Vault: Implementation and Performance.* 2007.

[29] Nandakumar, K., A. Nagar, and A.K. Jain, *Hardening fingerprint fuzzy vault using password*, in *Advances in biometrics*, Springer. p. 927-937. 2007.

[30] Nguyen T.H. et al. *A fingerprint fuzzy vault scheme using a fast chaff point generation algorithm*. in *Signal Processing, Communication and Computing (ICSPCC), IEEE International Conference on*. IEEE. 2013.

[31] Moon, D., et al. *Fuzzy fingerprint vault using multiple polynomials*. in *Consumer Electronics, 2009. ISCE'09. IEEE 13th International Symposium on*. IEEE. 2009.

[32] Örencik, C., et al., *Improved fuzzy vault scheme for fingerprint verification* 2008.

[33] Juels, A. and M. Sudan, A fuzzy vault scheme. Designs, Codes and Cryptography, 38(2): p. 237-257, 2006.

[34] Jeffers, J. and A. Arakala. *Fingerprint alignment for a minutiae-based fuzzy vault*. in *Biometrics Symposium, 2007*. IEEE. 2007.

BIOGRAPHIES

Hachemi Nabil Dellys: received an engineer degree in computer systems from computer national high schools, Algiers, Algeria, in 2009, and the Magister degree in distributed and mobile computing from the same schools in 2011. He is currently preparing a PhD. Degree in biometric securing at LMCS laboratory. He is also assistant professor in computer national high schools. His research interest is biometric security and pattern recognition.

Layth Sliman: completed his Diploma in Computer Engineering. Then he obtained his masters in Computer Science (Information systems) and his Phd from INSA Lyon, France.

In 2003, he underwent training courses in Development and Implementation program in Computer Software Applications in CMC-TATA, New Delhi, India. In the same year, he also underwent another training course in Information and Communication Technologies in MEIO University and Okinawa International Center, Japan. In 2008, 2009 2010, 2012 and 2013 he did many research stays in Digital Rights Management and image processing in the University of the Ryukyus and Ritsumeikan University - Japan. During the period 2000-2010, he worked as lecturer and assistant professor, did his research and taught Computer Engineering and Information Systems in many universities including INSA, Lyon and the university of the Ryukyus in Japan. Since September 2010 he is associate professor in EFREI, a French engineering school located in Paris.

Fathelalem Ali, is a professor of information engineering, at the Department of Management and Information Sciences, Meio University, Japan, where he joined as faculty in 2000. Professor Ali had obtained a B.Sc. in Mechanical Engineering from the University of Khartoum, in 1988, Master of Engineering in Electrical and Information Engineering, and Ph.D. of Information Engineering in Complex Intelligent Systems, from University of the Ryukyus, Japan, in 1997 and 2000, respectively.

Professor Ali has been working in a multidisciplinary research and education environment and has been actively involved with collaborative research with several universities and institutes in Japan, Malaysia, US, France and Sudan. His research work has been mainly in computational intelligence, and Internet technologies and applications. He had developed ICT related courses and curricula for university students as well as for training and capacity building agencies. His recent research interests are in data science and ICT in education and learning.

Noussaiba Benadjimi, holds an engineer and Master 2 diploma in computer systems from computer national high schools, Algiers, Algeria, in 2015. She is currently a first year PhD student in Database Optimization at the same schools at LCSI laboratory, Her thesis title is "Parallelization of relational operations of type Division". Her research will address key questions in optimization and perfection of relational operations in large databases. She is particularly interested in the effect of parallelism of the flexible relational division.

Meriem Roumaissa Boubakeur, holds an engineer and Master 2 diploma in computer systems from computer national high schools, Algiers, Algeria, in 2015. She is currently a first year PhD student in department of computer science and technology at Chongqing University in China.