

Received 13 July 2024, accepted 31 July 2024, date of publication 2 August 2024, date of current version 20 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3438076

RESEARCH ARTICLE

On Efficiency of Square-Boundaries Chaff Points Generation With Composite Representation in Fingerprint Fuzzy Vault

HACHEMI NABIL DELLYS¹, LAYTH SLIMAN², (Member, IEEE),
BRENDAN TRAN MORRIS³, (Member, IEEE), AND KARIMA BENATCHBA¹

¹Laboratoire des Méthodes de Conception des Systèmes, École nationale Supérieure d'Informatique, Oued Smar 16309, Algeria

²Efrei Research Laboratory, EFREI Paris, Pantheon Assas University, 94800 Villejuif, France

³Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, NV 89154, USA

Corresponding author: Layth Sliman (layth.sliman@efrei.fr)

ABSTRACT Fingerprint-based biometric systems are widely used because of their advantages against conventional authentication systems based on passwords and tokens. However, a major limitation is that individuals' fingerprint information cannot be easily changed if compromised. The fuzzy vault is a promising technique that secures fingerprint data by generating a set of data from the fingerprint using an injective function, preventing the original fingerprint from being regenerated. Nevertheless, the fingerprint fuzzy vault is computationally intensive and requires substantial memory resources. We propose enhancing the performance of fingerprint fuzzy vaults and reducing resource consumption using a new chaff point generation technique based on square boundaries and composite representation. We conducted integration testing along with detailed benchmarking of the fingerprint fuzzy vault using square-boundary generation against other techniques proposed in the literature for each stage. The experiments demonstrate that our proposal yields relatively better results in terms of False Rejection Rate, False Acceptance Rate, computational time, the number of chaff points generated, and memory usage.

INDEX TERMS Biometric, chaff points generation, composite representation, fingerprint fuzzy-vault.

I. INTRODUCTION

With the spread of new technologies such as the Internet of Things (IoT), Cloud Computing and mobile communication, novel business models like Industry 4.0, smart cities, and distributed supply chains have emerged. In these models, authenticating human activities and actions becomes an essential component of business processes and underlying information systems. This has driven the need for fault-tolerant, flexible, and lightweight biometric authentication schemes. Biometrics utilizes individuals' unique physical features or characteristics to authenticate people. However, biometric authentication techniques suffer from two significant problems. First, unlike passwords or magnetic cards, biometric templates cannot be easily replaced if their data are compromised. Hence, the templates must be secured. Second,

measurements of the same biometric property (fingerprint, iris, etc.) vary according to parameters such as resolution, sensing distance, noise, etc. Consequently, conventional encryption techniques used to secure non-biometric data are unsuitable for biometric templates, as using an error threshold during recognition is essential.

Numerous cryptosystems adapted for biometrics have been proposed to overcome this problem [32], [47], [50]. One of the most promising biometric cryptosystems for securing biometric modalities is the fuzzy vault [48], [49] especially fingerprints is [5], [15], [18], [23]. The fingerprint fuzzy vault process consists of two phases: vault encoding and vault decoding. Vault encoding involves encrypting a secret key with fingerprint features after adding chaff points (fake points with the same data structures as authentic points) to generate the vault. Vault decoding involves extracting the secret key from the vault using fingerprint features. Each phase comprises different stages. The encoding phase contains

The associate editor coordinating the review of this manuscript and approving it for publication was Zahid Akhtar¹.

five stages: computation of the secret polynomial, feature representation, point modeling, chaff point generation, and vault storage. The decoding phase includes three stages: point alignment, determination of the correspondence set, and secret polynomial reconstruction.

Many techniques have been proposed to handle each stage of the fingerprint fuzzy vault. Most works have focused on feature representation, chaff point generation, and point alignment, as these are the most complex and resource-intensive stages. Thus, these stages should be optimized for use in resource-constrained devices such as IoT objects.

In this context, in a preliminary work, we proposed a new chaff point generation technique called square-boundary-based generation (SBG) using composite representation [10]. However, only unit testing was conducted to assess the efficacy of the proposed technique. As mentioned, SBG is intended to be used in combination with other techniques in the eight stages of the fuzzy vault process. Therefore, a complete assessment involving the different stages and phases is necessary. In this work, we conduct integration testing and benchmarking involving our SBG and all possible combinations of the main techniques proposed for the different fuzzy vault stages [2], [3], [6], [33].

In this study, we aim to demonstrate the efficiency of our proposal and provide researchers with relevant information about appropriate strategies to ensure optimal implementation of feature representation, chaff point generation, and point alignment stages when developing new proposals for fuzzy vault processes.

The remainder of the paper is organized as follows: Section II describes the most efficient fingerprint feature representations, the most used chaff point generation techniques, and point alignment techniques. Section III describes our new technique for chaff point generation using square boundaries and composite representation. Section IV outlines our experimental methodology used to compare combinations of the most cited techniques used in fingerprint fuzzy vault stages. Section V presents an experimental comparison of the different technique combinations used in the fingerprint fuzzy vault, including our proposal. Finally, Section VI concludes and discusses the obtained results and the study's significance.

II. FINGERPRINT FUZZY VAULT

The fingerprint fuzzy vault process has five stages in the encoding phase and three stages in the decoding phase. We describe each stage briefly as follows.

A. FUZZY VAULT ENCODING PHASE

First, the fuzzy vault encoding phase is composed of five stages:

1) STAGE 1. COMPUTE THE SECRET POLYNOMIAL

The secret polynomial is generated from a secret key given by the user. Each digit of the secret key is integrated as a coefficient of a secret polynomial. Two strategies are used

to compute a secret polynomial: the first one generates only one polynomial from a secret key, while the second generates multiple polynomials [14], [24].

2) STAGE 2. FINGERPRINT FEATURE REPRESENTATION

Some features are extracted from a fingerprint and represented in an adequate data format to avoid recognition errors. While several features can be extracted from the fingerprint (minutiae, ridge, texture, etc.), minutiae representation is the most widely used [4], [21], [33]. Feature representation is crucial because the other stages depend strictly on the minutiae feature representation [22].

There are three main types of minutiae feature representations: representation by the tuple [20], representation of minutiae by their neighbors (Composite, Voronoi...) [30], [41] and representation by the hash table [20], [51].

a: MINUTIAE REPRESENTATION BY TUPLES

Minutiae representation by tuples is a set of coordinates describing the position of the minutiae. These coordinates comprise only Cartesian coordinates (x, y) in a 2D representation [40]. In 3D representation, an orientation θ is added to produce (x, y, θ) set, representing a minutia's position and orientation [16]. In 4D representation, the minutia type (endpoint or fork) is added to the tuple (x, y, θ , T). 4D representation is the most commonly used in the literature [26].

b: MINUTIAE REPRESENTATION BY ITS NEIGHBORS

This representation describes the minutiae concerning its neighbors. The main approaches for this representation are:

- The five nearest neighbors' representation (Fig. 1-a) describes a minutia (m) by the distances from its five closest neighboring minutiae (m_i) to form a structure noted as $\{(r_i, \theta_i)\}_{i=1}^5$. Where r_i are the Euclidean distances between m and m_i , and θ_i are the angles formed between m and m_i [13].
- Voronoi neighbors: In this representation, a Voronoi diagram [13] selects the neighbors of each minutia instead of using Euclidean distances and angles.
- Composite representation (Fig. 1-b): The minutia (m_i) is compared to its neighbors (m_j) to generate a set of 3-tuples ($d_{ij}, \varphi_{ij}, \theta_{ij}$) where:
 - d_{ij} is the Euclidean distance between the minutia m_i and minutia m_j ;
 - φ_{ij} is the result of subtraction of the orientation angles of m_i and m_j ;
 - θ_{ij} is the anti-clockwise angle between the orientation of m_i and direction from m_i to m_j .

c: MINUTIAE REPRESENTATION BY GEOMETRIC HASH

In this representation, each point in the vault (m) is indexed by all other points (b_i) to form a set of tuples $\{(b_i, m)\}$. The geometric hash table requires a large amount of memory as $k*(k-1)$ sets of points must be generated, where k is the number of points in the vault [20], [51].

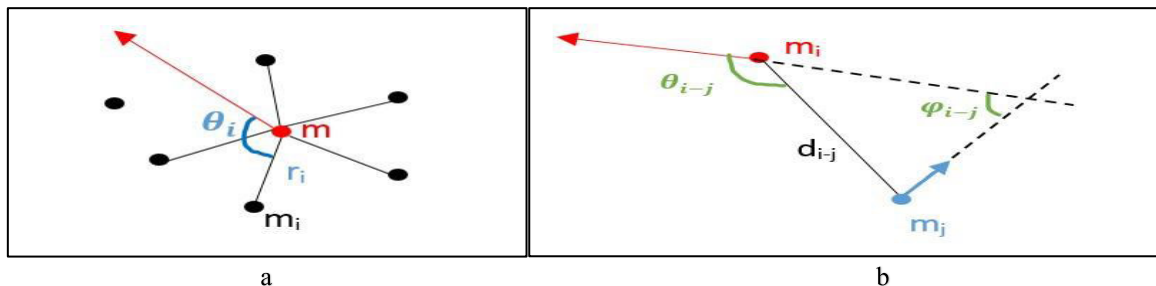


FIGURE 1. Examples of minutiae representation by its neighbours a- five nearest neighbors b- Composite representation.

3) STAGE 3. POINTS MODELING

The majority of feature representation structures employ data represented as a collection of vectors. However, the fuzzy vault process requires a scalar format. Consequently, vector data are concatenated to form encoding units [19]. The techniques most commonly utilized in this stage are sequential concatenation [28] and Galois representation [27].

4) STAGE 4. CHAFF POINTS GENERATION

Chaff points constitute a significant number of artificial points generated to obfuscate the authentic ones [34]. Chaff points possess the same representation and data structures as minutiae features. The objective is to render the identification of authentic points (minutiae), and hence, the acquisition of the secret polynomial, challenging for adversaries. According to [1], chaff point generation is subject to two constraints:

- 1) A chaff point should not be in close proximity to an authentic point (minutiae);
- 2) Chaff points should not be in close proximity to one another.

We employ the terms abscissa (c) and ordinate (d) to determine the feature representation of a chaff point in this stage [29]. We describe them as follows:

i. Chaff points Abscissae generation

The abscissa is a data structure that shares the same representation as authentic points. In the fingerprint fuzzy vault, abscissae are generated from minutiae by different techniques. Our experiments compared the most cited techniques in the literature: one threshold generation technique, two thresholds generation technique, and the square-boundaries-based generation technique. Next, we explain these techniques:

1. One threshold technique

This technique generates chaff points using a Euclidean distance to separate each point in the vault from the other points by a predefined threshold (δ) (Fig. 2) [36].

2. Two thresholds technique

In one threshold technique, all the generated chaff points maintain a minimum threshold from each other. Consequently, the generated set of points exhibits a homogeneous distribution of chaff points in the vault. However, in actual cases, authentic points are often in close proximity to one another, forming a high-density cluster. This enables adversaries to distinguish this mass of points and identify

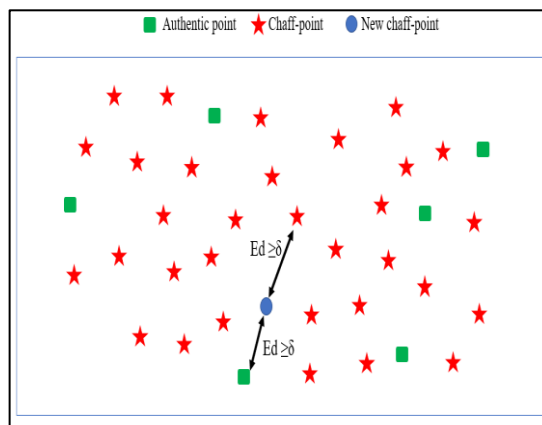


FIGURE 2. Simulation of chaff-points generation with One-threshold method.

authentic points using degree of freedom attacks [43]. To mitigate this issue, two thresholds are employed. The first threshold ($\delta 1$) ensures that a generated chaff point maintains a minimum distance from any authentic point. The second threshold ($\delta 2$) is used to tolerate having smaller distances between chaff points themselves (Fig. 3) [36].

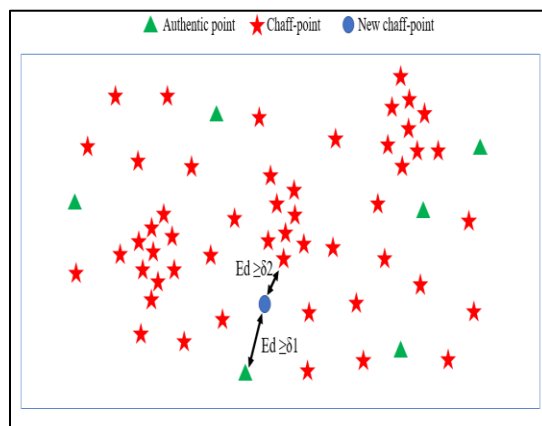


FIGURE 3. Simulation of chaff-points generation with Two-thresholds method.

3. Geometric shape-based technique:

This technique creates a boundary around each point of the vault by a fixed-size geometric shape. Each geometric shape must never overlap with other ones (Fig. 4) [16], [29].

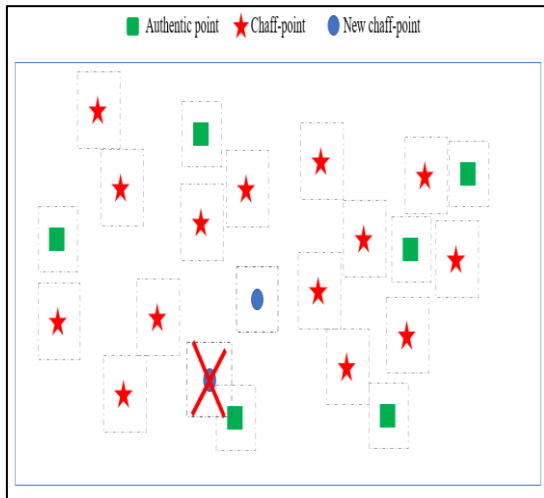


FIGURE 4. Simulation of chaff-points generation with Square-boundaries method.

ii. Chaff points ordinates generation

The ordinate is an evaluation of the distance of chaff points from the secret polynomial. Authentic points' ordinates must reside on the secret polynomial graph. In contrast, chaff points' ordinates must not be in close proximity to the secret polynomial graph. This is employed to avoid ambiguity in the decoding phase.

There are two strategies to generate the ordinates of chaff points: The first involves randomly generating the ordinate while ensuring that a chaff point (c_i, d_i) does not belong to the secret polynomial graph. The second strategy uses the formula $d_i = P(c_i) + \alpha$, where $P(c_i)$ is a point on the secret polynomial graph that corresponds to the abscissa of the authentic point (c_i) , and α is a real scalar generated randomly [26].

5) STAGE 5. VAULT CONSTRUCTION AND STORAGE

Authentic points and chaff points are combined to generate a vault. render the detection of authentic points arduous by obfuscating them amidst a substantial number of generated chaff points [39], [52].

B. FUZZY VAULT DECODING PHASE

The data-decoding phase comprises three stages as detailed below:

1) STAGE 1. POINTS-ALIGNMENT

The misalignment between stored fingerprint templates and newly acquired ones is corrected with the aid of an error threshold to circumvent recognition failure [21], [38]. Point alignment is the first stage in the decoding phase of the fuzzy vault process. This stage attempts to establish an alignment between authentic points in the vault and the extracted minutiae from a fingerprint. Incorrect alignment of certain features will inevitably result in authentication failure. Numerous techniques exist to achieve point alignment. However, the choice of technique is strongly contingent upon the feature representation. This implies that each alignment

technique is adapted to specific feature representations and cannot be utilized with others.

The main strategies employed for points-alignment in fingerprint fuzzy vault are:

i. Fingerprint pre-alignment.

Two main techniques are used. In the first technique, minutiae are represented in relation to a pre-chosen strong minutia used as a Reference point [37]. The second technique uses Helper data, which generally refers to public data that conveys sufficient information to perform the alignment without divulging any useful information about the original fingerprint [36].

ii. Fingerprint auto-alignment.

If the feature representation technique used in stage 2 is invariant to translation and rotation of fingerprint, then all points are relatively aligned during recognition [31]. Thus, no pre-alignment process is implemented [27]. Feature representations that are invariant to translation and rotation include minutiae representation by its neighbors [31] and alignment by the geometric hash table, which is considered one of the most accurate strategies [20].

2) STAGE 2. DETERMINATION OF CORRESPONDENCE SET

This stage implies matching between points extracted from a fingerprint image captured after points-alignment stages and the vault using a threshold error. If numerous correspondences are positive, a secret polynomial is reconstructed [41]. In this stage, the Euclidean distance is the most commonly utilized technique to determine the correspondence set [13].

3) STAGE 3. SECRET POLYNOMIAL RECONSTRUCTION

In this stage, a secret polynomial is regenerated from the matched set obtained in the previous stage. Subsequently, the secret key is extracted from the polynomial coefficients. If the extracted secret key is identical to the inserted one, the user is successfully authenticated [25].

III. SQUARE-BOUNDARIES CHAFF POINTS FOR COMPOSITE REPRESENTATION

In this paper, we propose a new chaff points generation technique based on square-boundaries and composite representation. One issue with typical chaff point generation is that it can be computationally intensive [35]. In preliminary work [10], we demonstrated that composite representation and square-boundaries chaff point generation yield promising results with better false acceptance rate (FAR) and false rejection rate (FRR) performance and computational time compared to other existing techniques. We also showed that the challenge lies in constructing square-boundaries for composite representation, which is not intuitive for representation by tuples.

We define the mathematical representation of our proposal as follow:

- Let Υ be the set of points in the vault.
- Each point $m_i \in \Upsilon$ is represented relatively by each point $m_j \in \Upsilon$ using a composite representation $(d_{ij}, \varphi_{ij}, \theta_{ij})$, where:

Algorithm 1 Algorithm of Generating Chaff-Points Using Composite Representation and Square-Boundaries Technique

```

 $\Gamma \leftarrow \text{initial\_set\_of\_points}()$ 
 $\text{Thr} \leftarrow \text{threshold\_value}$ 
function Is_VALID_CHAFF_POINT ( $m_k, \Gamma, \text{Thr}$ )
  for  $m_i \in \Gamma$  do
    for  $m_j \in \Gamma$  do
      for  $m_l \in \Gamma$  do
         $d_{ik} \leftarrow \text{Euclidean\_distance}(m_k, m_i)$ 
         $d_{jl} \leftarrow \text{Euclidean\_distance}(m_j, m_l)$ 
         $\phi_{ik} \leftarrow \text{Orientation\_angle}(m_k, m_i)$ 
         $\phi_{jl} \leftarrow \text{Orientation\_angle}(m_j, m_l)$ 
        if  $|d_{ik} - d_{jl}| < \text{Thr} / 2$  then
          return False
        end if
        if  $|\phi_{ik} - \phi_{jl}| < \text{Thr} / 2$  then
          return False
        end if
      end
    end
  end
  return True
end
while True do
   $m_k \leftarrow \text{Generate\_new\_point}()$ 
  if Is_Valid_ChAFF_POint ( $m_k, \Gamma, \text{Thr}$ ) then
     $\Gamma \leftarrow \Gamma \cup \{m_k\}$ 
  else
    break
  end if
end while

```

- d_{ij} is the Euclidean distance between points m_i and m_j .
- ϕ_{ij} is an orientation angle between points m_i and m_j .
- θ_{ij} is an anti-clockwise orientation angle between points m_i and m_j .

A square boundary is defined around the composite representation of each point to ensure chaff points are sufficiently distinct. The boundary length is denoted as Thr .

The goal is to maximize the number of chaff points (cardinality of m_i) while ensuring each new chaff point is sufficiently distinct from existing points in terms of both distance and orientation.

Objective:

$$\text{maximize } (\text{card}(m_i)) \quad (1)$$

Under constraints:

$$\forall m_i, m_j, m_k, m_l \in \Upsilon \quad \begin{cases} |d_{ik} - d_{jl}| \geq \frac{\text{Thr}}{2} \\ |\phi_{ik} - \phi_{jl}| \geq \frac{\text{Thr}}{2} \end{cases} \quad (2)$$

where:

- $|d_{ik} - d_{jl}| \geq \frac{\text{Thr}}{2}$ ensure that the Euclidean distance between new chaff points and existing points is sufficiently distinct.
- $|\phi_{ik} - \phi_{jl}| \geq \frac{\text{Thr}}{2}$ ensure that orientations between new chaff points and existing points is sufficiently distinct.

Given these conditions, the boundaries around each point's composite representation will be sufficiently distinct, preventing overlap and ensuring the integrity of the chaff points. It also guarantees that a point " m " is centered in a square boundary.

The Algorithm of generating chaff points is described as follow in Algorithm 1:

Table 1 illustrates an example of the generation of one chaff point using the condition of formula (1) and formula (2). In this example, we use a square boundary with a length $\text{Thr}=10$. Suppose that there is a point m_i represented with composite representation using another point c_1 . To generate a new chaff point p_j physically close to m_i , we can employ another random point in the vault to create a composite representation. If we use the point c_1 , we note that the two conditions of formula (2) are not satisfied, as the differences in distances and anti-clockwise orientations are smaller than 5 ($\text{Thr} \div 2$). If we use the point c_2 , we note that the condition of difference of anti-clockwise orientation is satisfied, but the condition for distances is not. However, if we use the point c_3 , we note that both conditions of formula (2) are satisfied. In this case, the chaff point p_j is accepted and added in the vault, even though physically, the points m_i and p_j are very close.

IV. MATERIALS AND TECHNIQUES

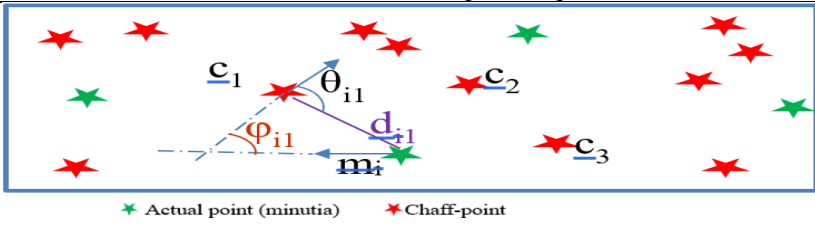
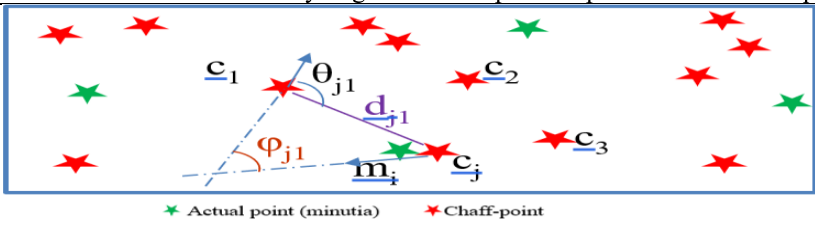
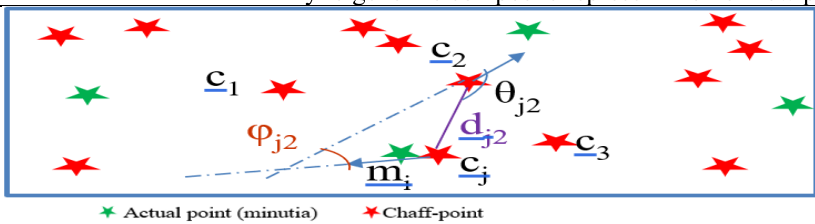
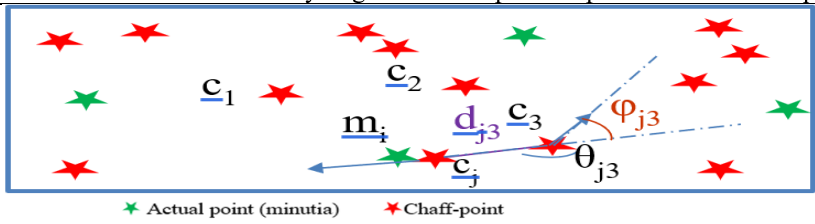
Our proposal is conducted in two phases. In the first phase, we compare our proposal with the thresholds-based chaff points techniques in unit experiments conducted solely in the chaff points generation stage to verify the efficacy of our proposed approach.

In the second phase of our experiments, we conduct a comparison between our proposal and numerous combinations of fingerprint feature representation along with their related points-alignment [44], and chaff points generation techniques [7]. These three stages are considered as the most critical, resource greedy, and extensively studied stages in the fingerprint fuzzy vault process [46], [42].

To demonstrate the efficiency of our proposal, we performed a comparison study with all major combinations of techniques used in the fingerprint fuzzy vault process. The comparison study is divided into three parts:

- The first comparison concerns feature representation and their related points-alignment techniques only. The other stages are fixed. The objective is to show that the results are not only due to feature representation.
- The second comparison study concerns chaff point generation techniques only. The other stages are fixed. The objective is to show that the results are not solely due to chaff point generation technique utilized.

TABLE 1. Example of generation a chaff point using our proposal.

	Figures	Formulas
1	Composite representation of c1 with m _i 	$\varphi_{i1}=65^\circ$ $d_{i1}=14$ Thr=10 $\begin{cases} d_{ik} - d_{jl} \geq Thr \div 2 \\ \varphi_{ik} - \varphi_{jl} \geq Thr \div 2 \end{cases}$
	Try to generate composite representation of new point c _j with c ₁ 	$\varphi_{j1}=62^\circ$ $d_{j1}=15$ Thr=10 $\begin{cases} 14 - 15 \geq 5 \\ 65 - 62 \geq 5 \end{cases}$ Rejected
3	Try to generate composite representation of new point c _j with c ₂ 	$\varphi_{j2}=56^\circ$ $d_{j2}=10$ Thr=10 $\begin{cases} 14 - 10 \geq 5 \\ 65 - 56 \geq 5 \end{cases}$ Rejected
	Try to generate composite representation of new point c _j with c ₃ 	$\varphi_{j3}=53^\circ$ $d_{j3}=6$ Thr=10 $\begin{cases} 14 - 6 \geq 5 \\ 65 - 53 \geq 5 \end{cases}$ Accepted

- The third comparison study concerns the best combinations obtained to demonstrate the efficiency of our proposal.

The selection of techniques employed in the different stages has been made according to our previous theoretical study [9]. In the following, we provide a concise explanation for these choices.

Table 2 summarizes all the techniques utilized or compared in each stage of fuzzy vault in our experiments.

A. IN THE ENCODING PHASE

Stage 1- Secret polynomial generation: One polynomial generation gives the best performance in terms of False Rejection Rate (FRR) and False Acceptance Rate (FAR), and it is the most commonly used technique in literature [42]. Multiple polynomials are efficient only with a small number

of minutiae (<12) [40]. Therefore, we choose to use one secret polynomial in our experiments.

Stage 2- Feature representation: In this stage, we compared the three most commonly used fingerprint feature representations in fuzzy vault process: the 4D representation, hash table representation, and the composite representation (which is the feature representation we used in our proposal). The three feature representations used in experiments are highlighted in red color in Table 2.

Stage 3- Points modeling: The template obtained from feature representation units must be transformed into scalar data so that they can fit to fuzzy vault process. To achieve this, part-by-part concatenation is typically used, except for composite representation, where part-by-part concatenation cannot be used, and it is replaced by Galois representation [27].

TABLE 2. Techniques used (line 1,3,5,7 and 8) and those compared (red line 2, 4 and 6) in our experiments.

	Fuzzy vault stages	Techniques used or compared		
Encoding phase	Determination of secret polynomial	<i>One polynomial</i>		
	Feature representation	<i>4D tuples</i>	<i>Composite</i>	<i>Geometric hash table</i>
	Points modelling	<i>Sequential concatenation or Galois concatenation when Composite representation is employed</i>		
	Chaff points abscissae generation	<i>One threshold</i>	<i>Two thresholds</i>	<i>Square-boundaries</i>
	Chaff points ordinate generation	<i>Random Generation</i>		
Decoding phase	Points Alignment	<i>Reference point alignment</i>	<i>Alignment by geometric hashing</i>	<i>Automatic alignment</i>
	Determination of correspondence	<i>Euclidean distance</i>		
	Secret polynomial reconstruction	<i>Brute force</i>		

Stage 4- Chaff points generation: This stage consists of two parts: abscissae generation and ordinate generation. For abscissae generation we compare between the popular one-threshold and two-threshold, with our proposed square-boundaries generation technique. We employ random ordinate generation as it is the most commonly used technique in the literature due to its performance. The three chaff points generation techniques used in experiments are highlighted in red color in Table 2.

Stage 5- Vault construction and storage: Authentic points, extracted from fingerprint minutiae and formatted with desired feature representation, are combined with chaff points to construct the vault. This stage is performed simultaneously with the chaff point generation stage.

B. IN THE DECODING PHASE

Stage 6 - Points-alignment: In this stage, we compare combinations of feature representations and alignment techniques. The combinations include reference point alignment with 4D tuples, automatic alignment for composite representation, and geometric hashing alignment for hash table representation. The three points-alignment techniques used in experiments are highlighted in red color in Table 2.

Stage 7- Determination of correspondence set: We selected Euclidian distance as it the most commonly used technique to determine the correspondence set [12].

Stage 8 - Secret polynomial reconstruction: The brute force technique is the most commonly used technique to reconstruct the secret polynomial, and it is the only technique that can reconstruct the polynomial in all cases [25]. Consequently, we choose this technique to obtain a fair comparison between all combinations of techniques in other stages.

C. IMPLEMENTATION DETAILS

Our experiments were conducted on two databases (1800 fingerprints): the DB2_A, FVC2006 database [45], with 140 individuals and 12 fingerprint instances per individual, and the DB2_B, FVC2006 database, with 10 individuals and 12 fingerprint instances per individual. The experiments were performed with the help of a software platform based on Java and Matlab languages. This software runs on a computer hardware with an Intel Core 2 Duo processor (3.2GHz, 4mo cache memory) and 8 GB of random-access memory.

The experiments were launched on nine combinations of techniques in fingerprint fuzzy vault process, especially in feature representation, chaff points generation and points alignment stages as summarized in Table 3.

V. EXPERIMENTAL EVALUATION

In this section, we conduct different experiments on the most well-known techniques for feature representation, chaff point generation, and point alignment stages in the fingerprint fuzzy vault by comparing the nine combinations shown in Table 3. First, we conduct unit tests in the chaff point generation stage to demonstrate the efficiency of the square-boundaries technique against threshold techniques. Then, we conduct experiments on combinations of techniques used in different stages three by three to show how the combination of techniques can impact the fingerprint fuzzy vault process.

A. CHAFF POINTS UNIT TESTS

In this section, we compare the performances of chaff point generation techniques as a unit test to determine the average number of chaff points effectively generated and the average computational time for each of the one threshold technique, two thresholds technique, and square-boundaries technique.

Table 4 presents the results of the experiments conducted to determine the number of chaff points generated by each technique applied to 1800 fingerprints from the DB2_A and DB2_B FVC2006 databases. We note that our proposal can fully generate the desired number of chaff points, unlike the one and two thresholds-based techniques. The one threshold- technique ceased generating new chaff points after reaching approximately 200 chaff points. Similarly, the two thresholds technique stopped at around of 350 chaff points generated (Fig. 5-a). Chaff point generation is halted when the saturation threshold is reached, and very small chaff points are generated in the vault after that (between 3 and 7 in our experiments).

Additionally, the average computational time to generate chaff point remains insignificant for our square-boundaries-based method, at approximately 0.03 seconds per fingerprint for 1000 chaff points (Fig. 5-b). However, the computational time for one threshold-based technique increases exponentially with an average of 182 seconds per fingerprint.

The two-thresholds technique exhibits similar behavior, with an average of 231 seconds per fingerprint. The high computation time of the one-threshold and two-thresholds

TABLE 3. Summary of combinations of techniques used in the experiments. (in red, the stage focused in comparison for each combo).

Combo	Techniques used in each stage
Combo 1: 4D Tuple feature representation + reference point alignment	One polynomial+ 4D Tuple representation + Sequential concatenation+ One threshold+ Random Generation+ Reference point alignment + Euclidiene distance+ brute force
Combo 2: Geometric hash table representation + Alignment by geometric hashing	One polynomial+ Geometric hash table + Sequential concatenation+ One threshold+ Random Generation+ Alignment by geometric hashing + Euclidiene distance+ brute force
Combo 3: composite feature representation + Automatic alignment	One polynomial+ Composite representation + Galois concatenation+ One threshold+ Random Generation+ Automatic alignment + Euclidiene distance+ brute force
Combo 4: One threshold feature representation + reference point alignment	One polynomial+4D tuples+ Sequential concatenation+ One threshold + Random Generation+ Reference point alignment + Euclidiene distance+ brute force
Combo 5: Two thresholds feature representation + reference point alignment	One polynomial+4D tuples+ Sequential concatenation+ Two thresholds + Random Generation+ Reference point alignment + Euclidiene distance+ brute force
Combo 6: Square boundaries feature representation + reference point alignment	One polynomial+4D tuples+ Sequential concatenation+ Square-boundaries + Random Generation+ Reference point alignment + Euclidiene distance+ brute force
Combo 7: 4D Tuple feature representation + Square boundaries chaff points generation + reference point alignment	One polynomial+ 4D Tuple representation + Sequential concatenation+ Square-boundaries + Random Generation+ Reference point alignment + Euclidiene distance+ brute force
Combo 8: Geometric hash features representation + two thresholds chaff points generation + Alignment by geometric hashing	One polynomial+ Geometric hash table + Sequential concatenation+ Two thresholds + Random Generation+ Alignment by geometric hashing + Euclidiene distance+ brute force
Combo 9: Composite representation +square boundaries chaff points generation +automatic alignment.	One polynomial+ Composite representation + Galois concatenation+ Square-boundaries + Random Generation+ Automatic alignment + Euclidiene distance+ brute force

TABLE 4. Unit experiments results.

Feature representation	Desired number of chaff points	The average number of chaff points effectively generated	Average Computational time (per fingerprint)
One threshold	1000	≈197	182 s
Two thresholds	1000	≈343	231 s
Square-Boundaries	1000	1000	0,03 s

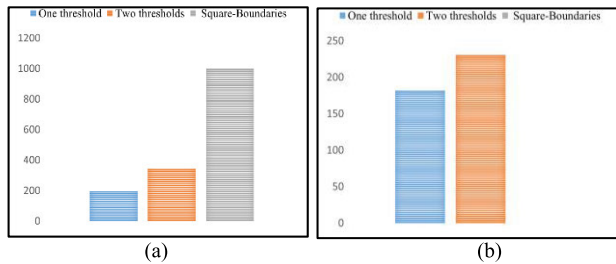


FIGURE 5. (a) Number of chaff-points generated. (b). Average Computational time (per fingerprint).

based techniques can be attributed to the complexity of generating new chaff points as the number of chaff points approaches the saturation threshold. This issue does not exist arise the square-boundaries technique, where the saturation threshold is not reached (up to 1000 chaff points) as it allows for the physical generation of two close points with different composite representations.

The obtained results from unit experiments in only the chaff points generation stage show that composite representation adapted to square-boundaries chaff points generation gives the best performances against Tuple representation and thresholds-based chaff points generation. To prove our proposal’s efficiency, we conduct a further experimental study of all stages of fuzzy vault process using combinations of the most typical techniques in feature representation and chaff points generation.

B. THE EFFICIENCY OF FEATURES REPRESENTATION AND POINTS-ALIGNMENT TECHNIQUES

In this section, we conduct a comparative experiment on the most well-known feature representations in the fingerprint fuzzy vault by comparing three combinations (**Combo 1:** 4D features representation + reference point alignment, **Combo 2:** Geometric hash feature representation and points alignment, and **Combo 3:** Composite features representation + automatic points alignment), as shown in Table 3. Each of these combinations employs a specific type of feature representation and its related points alignment techniques. Combo 1 utilizes 4D Tuple representation and reference point alignment [17]. Combo 2 employs geometric hash table representation and alignment by geometric hashing [20]. Finally, Combo 3 utilizes composite representation, enabling an automatic point alignment [36]. In the chaff points abscissae generation stage, the One threshold generation technique is employed because it can be used with all feature representations. Only 200 chaff points are generated to avoid

TABLE 5. Performances, computing time and memory usage of the first experiment.

	Chaff points generated	FRR	FAR	Avg. Computation	Avg. memory
Combo 1: 4D-Reference Point	200	69.7	0.7	0.31 s	2.14 kb
Combo 2: Geometric hash	1000	7	3	698.77 s	287 kb
Combo 3: Composite-Automatic	1000	18.4	13	0.62 s	1.65 kb

excessive computing time (see section V.A). The objective of this experiment is to compare the performances of feature representations regardless of the chaff point generation techniques employed, except when composite representation is utilized because it is impossible to concatenate tuples. In this case, Galois concatenation is employed to construct a set of 16-bit scalars in the point modeling stage. The comparison is conducted according to the performance of the false rejection rate (FRR), false acceptance rate (FAR), computational time, and the amount of memory usage.

As shown in Table 5, the performance in the hash table representation (Combo 2: Geometric Hash) is 7% for FRR and 3% for FAR. Tuple representation with alignment by reference point yielded an FRR of 69.7% and an FAR of 0.8%. The composite representation with automatic alignment combination (Combo 3) yielded an FRR of 18.4% and an FAR of 13%. As we can observe, Combo 1: 4D + Reference Point, which employs tuple representation along with alignment by reference point, yielded a very high FRR (69.7%) and a low FAR value (0.8%), which is a consequence of a large number of accepted individuals due to errors in point alignment. We also note that geometric hashing requires an average computation time of 698.77 seconds and 287 KB of average memory for one template. In contrast, 4D-tuple representation with alignment by reference point requires only an average time of 0.31 seconds and 2.14 KB of memory. The composite representation with automatic alignment requires an average time of 0.62 seconds and only 1.65 KB of memory.

As shown in (Fig. 6), The geometric hash table exhibits the best alignment in terms of FAR and FRR performances. However, the computational time required to generate the hash table and achieve alignment is significantly higher (1,127 times greater than that required for Combo 3 and 23,292 times greater than that for Combo 1). It necessitates a more significant amount of memory (287 KB) than other combinations. Even when Combo 2, which employs geometric hash table, is utilized to recognize one individual,

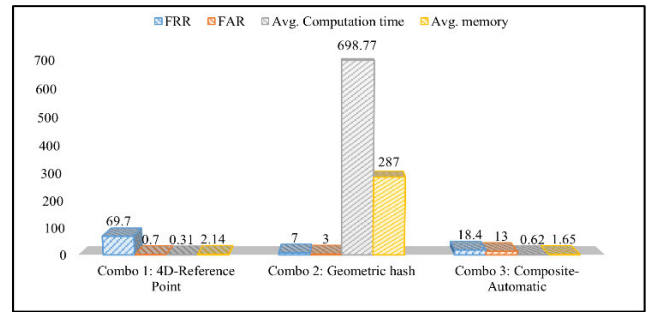


FIGURE 6. Performances, computing time and memory usage of the first experiment.

it takes up to 11 minutes to obtain results. This limits the use of this representation in resource-constrained devices.

In this experiment, we observed that the FRR and FAR performances are relatively strong in the fuzzy vault when the geometric hash table is employed for feature representation. However, the performance significantly declines when tuple representation and reference point alignment techniques are utilized. Combo 1 appears to be the optimal combination in term of average generation, However, as previously mentioned, we halted chaff points generation at 200 for fair comparison in this stage due of reaching the saturation threshold, as seen in the next experiment. Additionally, the FRR is notably high. As a result, Combo 3, which utilizes composite representation with automatic alignment, offers a favorable balance between computational time and reasonable memory usage with acceptable performance that can be further optimized.

C. THE EFFICIENCY OF ABSCISSAE CHAFF POINTS GENERATION TECHNIQUES

In this section, we study the efficiency of chaff points generation techniques through experimental comparisons between the most well-known techniques in the literature, which are: One threshold generation, Two thresholds generation, and Square-boundaries based generation [17], [44] in the entire fingerprint fuzzy vault encryption process. The comparison is conducted according to the number of chaff points effectively generated, computational time, and the memory usage. The objective of this experiment is to compare the performances of chaff point generation techniques regardless of the feature representation employed. Thus, the other stages are fixed: One polynomial generation, sequential concatenation, random generation, Euclidian distance, and brute force reconstruction are used respectively in their related stages. Since tuple representation is the only type of representation that can be used with all chaff point generation techniques [8], [17], it is utilized for fair comparison. Tuple representation requires the use of a reference point alignment in the point alignment stage [11]. In this experiment, we compare the performances of three combinations (Combo 4, Combo 5, and Combo 6) described in Table 3, according to their chaff points generation stage where combo 4 employs One

TABLE 6. Performances, computing time and memory usage of the second experiment.

	Desired number of chaff points	The average number of chaff points effectively generated	Estimation of Saturation threshold	Average Computational time before saturation threshold	Average Computational time after saturation threshold	Average memory
Combo 4: One threshold+reference point	1000	≈153	150	0.31 s	373 s	2.14 kb
Combo 5: Two thresholds+reference point	1000	≈297	290	0.42 s	188 s	3.22 kb
Combo 6: Square+reference point	1000	1000	Not reached	0.62 s	-	3.8 kb

thresholds generation of chaff points, combo 5 employs Two thresholds chaff points generation, and combo 6 employs Square-boundaries based chaff points generation.

As shown in Table 6, When One threshold technique (combo 4) and the Two thresholds technique (combo 5) are employed, the number of generated points does not exceed an average of 153 and 297 chaff points, respectively. These numbers indicate that the saturation thresholds (maximum number of chaff points that each technique can generate) are reached in each combination. In contrast, when the Square-boundaries technique is employed (combo 6), the number of generated chaff points reaches the desired level, and the saturation threshold was not reached before 1000 chaff points were generated.

As shown in (Fig. 7), the average time required to generate a new chaff point once the saturation threshold is reached became very high (373 s per chaff point for One-threshold technique and 188 s per chaff point for Two thresholds technique).

(Fig. 8) illustrates the evolution of computational time of combo 6, which used the Squares technique, remains consistently low (average of 0.62 seconds for 1,000 generated chaff points). In contrast, the One threshold and Two thresholds techniques initially require relatively short computational time before reaching their saturation threshold (150 and 290 chaff points generated with the One threshold technique and the Two thresholds techniques, respectively). However, the computation time increases exponentially accompanied by oscillatory behavior in generation time with very few numbers of new chaff points generated (between 3 and 7).

D. THE EFFICIENCY OF THE SQUARE-BOUNDARIES TECHNIQUE USING THE COMPOSITE REPRESENTATION

The previous experiments show that composite representation with automatic alignment provides the best compromise between FAR/FRR performance, computational times, and memory usage. The results also show that the square-boundaries technique yields the best results in chaff point abscissa generation. Nevertheless, we employ tuple representation in this experiment to ensure a fair comparison.

Consequently, we undertook to utilize the square-boundaries technique adapted to composite representation. In our experiments, we aim to test new combinations

TABLE 7. Performances, computing time and memory usage of the third experiment.

	FRR	FAR	Computational time Average	Average memory usage
Combo 7: 4D+square+reference point	62.2	0.8	0.42 s	4.29 kb
Combo 8: Geometric hash+ two thresholds	5.8	2.7	707.94 s	591 kb
Combo 9: Composite+square+automatic	15.3	8.6	0.83 s	2.68 kb

of techniques to study the efficiency of the chaff points generation based on squares-boundaries and composite representation. Therefore, we employ three combinations of the best ones obtained in the previous two experiments, as described in Table 3. Combo 7 utilizes 4D tuple representation with reference point alignment and a square-boundaries chaff point generation. Combo 8 employs the geometric hashing representation with their related points-alignment technique. Two thresholds chaff points generation is utilized in this combination because there is not yet an existing technique to generates a square-boundaries around a geometric hash table representation of any point. Finally, Combo 9 is our proposal, which employs composite representation with an automatic alignment and a square-boundaries chaff points generation.

Table 7 shows the FRR and FAR performance, average computational time, and average memory of the three combinations compared in the third experiment. The FAR and FRR are better using square-boundaries chaff points generation than those obtained in the second experiment where we employed One threshold generation. This is because the square-boundaries technique allows for a more robust representation of points and better reproducibility. However, we note that the average computational time is increased. This is due to the additional computational time caused by the calculation of the square-boundaries in combo 7 and 9, and the calculation of the second threshold in combo 8.

As shown in (Fig. 9), Combo 8, which utilizes Geometric hash representation, remains superior in terms of FRR and FAR. However, the average computational time and memory usage per fingerprint is respectively 835 times and 221 times higher than Combo 9. This technique may not be suitable for

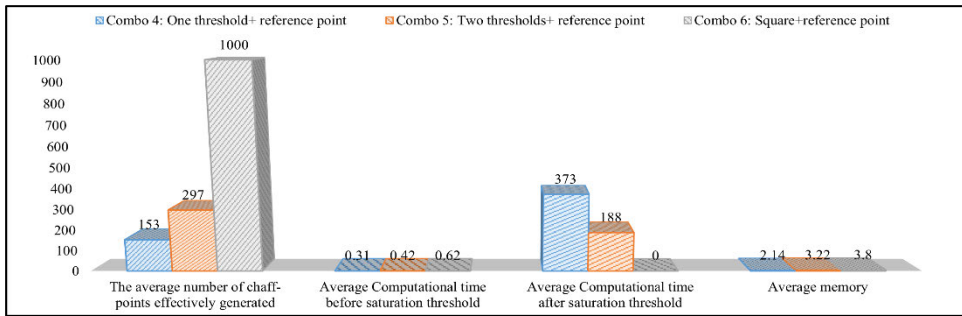


FIGURE 7. Performances, computing time and memory usage of the second experiment.

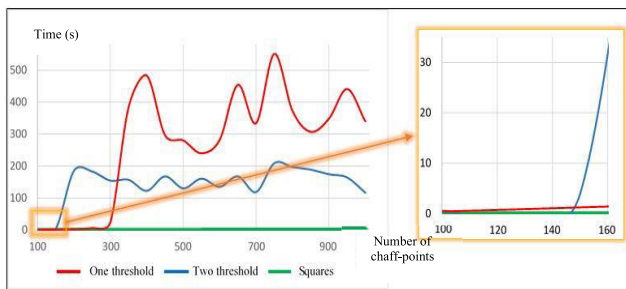


FIGURE 8. The evolution of time per number of chaff points generated.

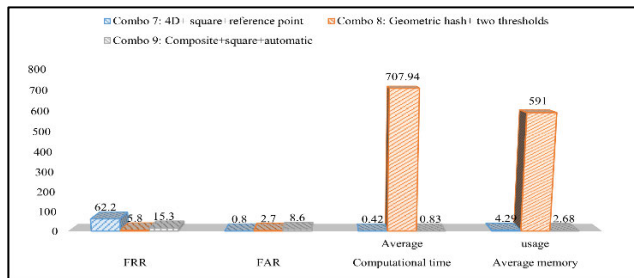


FIGURE 9. Performances, computing time and memory usage of the third experiment.

applications in real-time systems and resource-constrained devices.

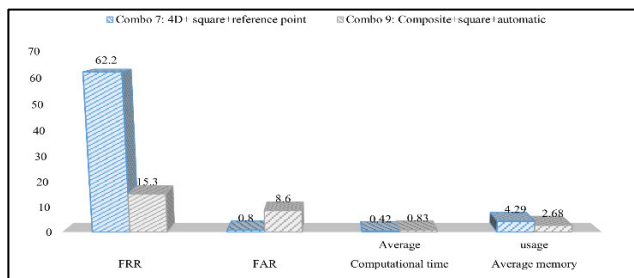


FIGURE 10. Third experiment performances of combo 7 and combo 9 only.

In contrast, (Fig. 10) reveals that Combo 7 is twice as fast as Combo 9, but it exhibits poor FRR performance and consumes twice as much memory as combo 9.

Finally, we note that employs combo 9 which utilizes composite representation with automatic alignment and Square-boundaries for chaff points generation, results in the best compromise between performance, computational time, and memory usage. This is particularly advantageous for applications in real-time systems and resource-constrained devices.

VI. RESULTS ANALYSIS AND STATISTICAL SIGNIFICANCE

In the last section, we conducted three experiments with nine combinations representing a combination of techniques used in the three most critical stages of fingerprint fuzzy vault combination. The objective is to study the efficiency of our proposal against others feature representation, chaff points generation, and points-alignment techniques. Table 8 summarizes the obtained results of all implemented combination.

In the first experiment (Combo 1, Combo 2, Combo 3), we studied the efficiency of the most popular techniques in the features representation and points alignment stages. The obtained results show that the best compromise between FRR/FAR performance, computing time, and memory usage was offered by the combo 3 (composite representation/automatic alignment).

In the second experiment (Combo 4, Combo 5, and Combo 6), we studied the efficiency of the most used chaff points generation techniques. The objective was to generate a maximum number of chaff points with minimal time expenditure and a small memory footprint while retaining good performance.

threshold before the desired 1000 points are generated. According to this experiment, we conclude that the square-boundaries techniques yields the best results by reaching the desired number of chaff points in the shortest computing time and reasonable memory usage.

In the third experiment (Combo 7, Combo 8 and, Combo 9), we compare our proposal (combo 9: composite+square) against the best techniques from the previous two experiments. Combo 7 employs 4D tuple representation with reference point alignment and square-boundaries chaff points generation. Combo 8 employs geometric hash representation with geometric hash points-alignment and Two-thresholds technique to generate chaff points. The results demonstrates

TABLE 8. Summary of results obtained in the ten combinations.

	Combo 1: 4D-Reference Point	Combo 2: Geometric hash	Combo 3: Composite+automatic	Combo 4: One threshold+reference point	Combo 5: Two thresholds+reference point	Combo 6: Square+reference point	Combo 7: 4D+ square+reference point	Combo 8: Geometric hash+ two thresholds	Combo 9: Composite+ square+automatic
Number of Chaff points (%)	20	100	100	20	35	100	20	35	100
Complexity	$O(n^3)$	$O(n^{n-1})$	$O(n^2)$	$O(n^3)$	$O(n^3)$	$O(n^2)$	$O(n^3)$	$O(n^{n-1})$	$O(n^2)$
FRR	69.7	7	18.4	69.7	68.2	69.4	62.2	5.8	15.3
FAR	0.7	3	13	0.7	0.7	0.8	0.8	2.7	8.6
Avg Time/after saturation (s)	0.31	698.77	0.62	0.31/373	0.42/188	0.62	0.42	707.94	0.83
Average memory (kb)	2.14	287	1.65	2.14	3.22	3.8	4.29	591	2.68

TABLE 9. ANOVA test for the ten combinations.

	Sum of squares	df	squares average	F	p-value
Between groups	1216746089	9	135194009,9	85162,9452	0
Within groups	24383609,42	15360	1587,474572	-	-
Total	1241129699	15369	-	-	-

that the best compromise between FRR/FAR performance, computational time, and memory usage is provided by Combo 9.

In this combination, acceptable chaff point generation performance is achieved (FRR = 8.6 and FAR = 15.3), along with good computational time (average of 0.83 seconds) and reasonable memory usage (average of 2.68 kb).

Since the first experiment is conducted on a different number of chaff points compared to those of the second and the third experiments, we carried out a statistical significance test to validate it. The data analysis of the results of the nine combinations yields the ANOVA table (Table 9).

We note that p-value is extremely close to zero. Therefore, we can say that the experiments are statistically significant. Besides, we perform a Student's t-test between each pair

of combinations. The obtained p-values are all between 0 and 0.03. Since all p-values are under 0.05, this confirms that the results obtained by the ANOVA test, as well as the experiments conducted in this work, are statistically significant.

VII. CONCLUSION

A novel fingerprint fuzzy vault chaff points generation technique using composite representation and squares-boundaries-based generation is proposed in this paper. First, a unit test was conducted in chaff points generation stage. The obtained results demonstrates that our proposal yields superior results compared to other techniques. Second, comparative experiments in a full fingerprint fuzzy vault process were conducted using nine combinations of the most popular feature representation techniques, chaff point generation, and point alignment stages. The objective was to compare our proposal and other popular techniques in literature to determine the optimal combination for resource-constrained devices considering performance, computational requirements, and memory usage. The experiments indicated that to achieve a high level of security, the use of geometric hash table representation is recommended (FRR≈5.8 and FAR≈2.7). Nevertheless, this combination necessitates a relatively prolonged computational time (≈707s) and substantial memory usage (≈591Kb) per fingerprint. Hence,

it is not well-suited for real-time environments or resource-constrained devices.

The results also demonstrate that our proposal, employing composite representation and the square-boundary technique in chaff point generation, is more efficient than the other combinations and provides the best compromise between FRR/FAR performance ($FRR \approx 15.3$ and $FAR \approx 8.6$), computational time and memory usage. This is particularly advantageous since fingerprint recognition systems are generally operated using resource-constrained devices and in real-time environments.

As future work, we aim at testing our findings to implement seamless and lightweight authentication mechanisms in business scenarios, especially those related to Industry 4.0.

ACKNOWLEDGMENT

The authors would like to thank the Fingerprint Verification Competition (FVC) for providing the FVC 2006 Database. The FVC 2006 Database was accessed from the following link: <https://bias.csr.unibo.it/fvc2006/default.asp> (link consulted date: 25/04/2024).

REFERENCES

- [1] E. Alibeigi, M. T. Rizi, and P. Behnamfar, "Pipelined minutiae extraction from fingerprint images," in *Proc. Can. Conf. Electr. Comput. Eng.*, May 2009, pp. 239–242, doi: [10.1109/CCECE.2009.5090128](https://doi.org/10.1109/CCECE.2009.5090128).
- [2] A. I. Arrahmah, Y. S. Gondokaryono, and K.-H. Rhee, "Fast non-random chaff point generator for fuzzy vault biometric cryptosystems," in *Proc. 6th Int. Conf. Syst. Eng. Technol. (ICSET)*, Oct. 2016, pp. 199–204, doi: [10.1109/ICSEngT.2016.7849650](https://doi.org/10.1109/ICSEngT.2016.7849650).
- [3] V. S. Baghel, S. Prakash, and I. Agrawal, "An enhanced fuzzy vault to secure the fingerprint templates," *Multimedia Tools Appl.*, vol. 80, nos. 21–23, pp. 33055–33073, Sep. 2021, doi: [10.1007/s11042-021-11325-w](https://doi.org/10.1007/s11042-021-11325-w).
- [4] K. Bobkowska, K. Nagaty, and M. Przyborski, "Incorporating iris, fingerprint and face biometric for fraud prevention in e-passports using fuzzy vault," *IET Image Process.*, vol. 13, no. 13, pp. 2516–2528, Nov. 2019, doi: [10.1049/iet-ipr.2019.0072](https://doi.org/10.1049/iet-ipr.2019.0072).
- [5] M. Gorski and W. Wodo, "Analysis of biometric-based cryptographic key exchange protocols—BAKE and BRAKE," *Cryptography*, vol. 8, no. 2, p. 14, Apr. 2024, doi: [10.3390/cryptography8020014](https://doi.org/10.3390/cryptography8020014).
- [6] D. Chitra and V. Sujitha, "Security analysis of prealigned fingerprint template using fuzzy vault scheme," *Cluster Comput.*, vol. 22, no. 5, pp. 12817–12825, Sep. 2019, doi: [10.1007/s10586-018-1762-6](https://doi.org/10.1007/s10586-018-1762-6).
- [7] T. K. Dang, M. T. Nguyen, and Q. H. Truong, "Chaff point generation mechanism for improving fuzzy vault security," *IET Biometrics*, vol. 5, no. 2, pp. 147–153, Jun. 2016.
- [8] A. F. De Abiega-L'Eglise, G. Gallegos-Garcia, M. Nakano-Miyatake, M. Rosas Otero, and V. Azpeitia Hernández, "A new fuzzy vault based biometric system robust to brute-force attack," *Computación Y Sistemas*, vol. 26, no. 3, pp. 1–24, Sep. 2022.
- [9] H. N. Dellys, N. Benadjimi, M. R. Boubakeur, L. Sliman, K. Benatchba, S. Artabaz, and M. Koudil, "A critical comparison of fingerprint fuzzy vault techniques," in *Proc. Adv. Visual Inform. Ser.*, vol. 9429, 2015, pp. 178–188.
- [10] H. N. Dellys, N. Benadjimi, M. R. Boubakeur, L. Sliman, and F. Ali, "Chaff point generation by squares technique using composite representation in fingerprint fuzzy vault," *J. Inf. Assurance Secur.*, vol. 11, pp. 1–10, Sep. 2016.
- [11] K. Harmer, W. Sheng, G. Howells, M. Fairhurst, and F. Deravi, "Fuzzy vault fingerprint smartcard implementation using an orientation-based feature vector," in *Proc. Bio-inspired, Learn. Intell. Syst. Secur.*, Aug. 2008, pp. 1–26, doi: [10.1109/bliss.2008.20](https://doi.org/10.1109/bliss.2008.20).
- [12] R. Hooda and S. Gupta, "Fingerprint fuzzy vault: A review," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, pp. 1–18, Sep. 2013.
- [13] J. Jeffers and A. Arakala, "Minutiae-based structures for a fuzzy vault," in *Proc. Biometrics Symp., Special Session Res. Biometric Consortium Conf.*, Sep. 2006, pp. 1–28, doi: [10.1109/BCC.2006.4341622](https://doi.org/10.1109/BCC.2006.4341622).
- [14] J. Jeffers and A. Arakala, "Fingerprint alignment for a minutiae-based fuzzy vault," in *Proc. Biometrics Symp.*, Sep. 2007, pp. 1–27, doi: [10.1109/BCC.2007.4430546](https://doi.org/10.1109/BCC.2007.4430546).
- [15] R. B. Joshi, "Cryptographic fuzzy vault with image processing," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 5, pp. 434–437, Jul. 2016, doi: [10.17148/IJARCC.2016.51108](https://doi.org/10.17148/IJARCC.2016.51108).
- [16] M. Khalil-Hani and R. Bakhteri, "Securing cryptographic key with fuzzy vault based on a new chaff generation method," in *Proc. Int. Conf. High Perform. Comput. Simul.*, Jun. 2010, pp. 1–29, doi: [10.1109/hpcs.2010.5547122](https://doi.org/10.1109/hpcs.2010.5547122).
- [17] M. Khalil-Hani, M. N. Marsono, and R. Bakhteri, "Biometric encryption based on a fuzzy vault scheme with a fast chaff generation algorithm," *Future Gener. Comput. Syst.*, vol. 29, no. 3, pp. 800–810, Mar. 2013, doi: [10.1016/j.future.2012.02.002](https://doi.org/10.1016/j.future.2012.02.002).
- [18] P. Kaur and N. Kumar, "Biometric cryptosystem with deep learning: A new frontier in security," in *Proc. Int. Conf. Adv. Power, Signal, Inf. Technol. (APSIT)*, Jun. 2023, pp. 739–744, doi: [10.1109/apsit58554.2023.10201772](https://doi.org/10.1109/apsit58554.2023.10201772).
- [19] G. Khachatryan, A. Jivanyan, and H. Khasikyan, "Alignment-free fuzzy vault scheme for fingerprints," in *Proc. 9th Int. Conf. Comput. Sci. Inf. Technol. Revised Sel. Papers*, Sep. 2013, pp. 1–6, doi: [10.1109/CSITech-nol.2013.6710354](https://doi.org/10.1109/CSITech-nol.2013.6710354).
- [20] S. Lee, D. Moon, and H. Choi, "Memory-efficient fuzzy fingerprint vault based on the geometric hashing," *Inf. Secur. Assurance*, vol. 1, pp. 1–19, Sep. 2008.
- [21] C. Li and J. Hu, "A security-enhanced alignment-free fuzzy vault-based fingerprint cryptosystem using pair-polar minutiae structures," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 3, pp. 543–555, Mar. 2016, doi: [10.1109/TIFS.2015.2505630](https://doi.org/10.1109/TIFS.2015.2505630).
- [22] X. Li, N. Ding, H. Lu, D. Gu, S. Wang, B. Xu, Y. Yuan, and S. Yan, "A modified fuzzy fingerprint vault based on pair-polar minutiae structures," in *Proc. Int. Conf. Inf. Secur. Cryptol.*, 2017, pp. 482–499.
- [23] S. Sharma, A. Saini, and S. Chaudhury, "Multimodal biometric user authentication using improved decentralized fuzzy vault scheme based on blockchain network," *J. Inf. Secur. Appl.*, vol. 82, May 2024, Art. no. 103740, doi: [10.1016/j.jisa.2024.103740](https://doi.org/10.1016/j.jisa.2024.103740).
- [24] R. Mehmood and A. Selwal, "Polynomial based fuzzy vault technique for template security in fingerprint biometrics," *Int. Arab J. Inf. Technol.*, vol. 17, no. 6, pp. 926–934, Nov. 2020.
- [25] D. Moon, S. Lee, Y. Chung, S. Bum Pan, and K. Moon, "Implementation of automatic fuzzy fingerprint vault," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Jul. 2008, pp. 1–16, doi: [10.1109/icmlc.2008.4621063](https://doi.org/10.1109/icmlc.2008.4621063).
- [26] D. Moon, W.-Y. Choi, K. Moon, and Y. Chung, "Fuzzy fingerprint vault using multiple polynomials," in *Proc. IEEE 13th Int. Symp. Consum. Electron.*, May 2009, pp. 290–293, doi: [10.1109/ISCE.2009.5156914](https://doi.org/10.1109/ISCE.2009.5156914).
- [27] A. Nagar, K. Nandakumar, and A. K. Jain, "Securing fingerprint template: Fuzzy vault with minutiae descriptors," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–29, doi: [10.1109/icpr.2008.4761459](https://doi.org/10.1109/icpr.2008.4761459).
- [28] K. Nandakumar, A. Nagar, and A. K. Jain, "Hardening fingerprint fuzzy vault using password," in *Advances in Biometrics*. Cham, Switzerland: Springer, 2007, pp. 927–937.
- [29] T. H. Nguyen, Y. Wang, T. N. Nguyen, and R. Li, "A fingerprint fuzzy vault scheme using a fast chaff point generation algorithm," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Aug. 2013, pp. 1–6, doi: [10.1109/ICSPCC.2013.6664061](https://doi.org/10.1109/ICSPCC.2013.6664061).
- [30] T. H. Nguyen, Y. Wang, Y. Ha, and R. Li, "Performance and security-enhanced fuzzy vault scheme based on ridge features for distorted fingerprints," *IET Biometrics*, vol. 4, no. 1, pp. 29–39, Mar. 2015, doi: [10.1049/iet-bmt.2014.0026](https://doi.org/10.1049/iet-bmt.2014.0026).
- [31] S. Patil, M. Hendre, and A. Abhyankar, "Alignment free fingerprint template security using minutia neighbors," in *Proc. IEEE Int. Carnahan Conf. Secur. Technol.*, Jul. 2023, pp. 1–4.
- [32] S. Sapkal and R. R. Deshmukh, "Biometric template protection with fuzzy vault and fuzzy commitment," in *Proc. 2nd Int. Conf. Inf. Commun. Technol. Competitive Strategies*, Mar. 2016, pp. 1–16, doi: [10.1145/2905055.2905118](https://doi.org/10.1145/2905055.2905118).
- [33] P. Kaur and N. Kumar, "SIFTBCS: Scale invariant feature transform based fuzzy vault scheme in biometric cryptosystem," *Multimedia Tools Appl.*, vol. 83, no. 10, pp. 28635–28656, Sep. 2023.

- [34] N. Singla, M. Kaur, and S. Sofat, "Secure fingerprint fuzzy vault including novel chaff point generation method," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 1–10, doi: [10.1109/ccaa.2017.8229959](https://doi.org/10.1109/ccaa.2017.8229959).
- [35] Z. Song, D. Chen, J. He, W. Yuan, and P. Chen, "A fuzzy vault scheme with multi-secret sharing," in *Proc. 3rd Int. Conf. Comput. Commun. Netw. Secur.*, Oct. 2022, pp. 1–11, doi: [10.1117/12.2659114](https://doi.org/10.1117/12.2659114).
- [36] P. Sood and M. Kaur, "Methods of automatic alignment of fingerprint in fuzzy vault: A review," in *Proc. IEEE Recent Adv. Eng. Comput. Sci. (RAECS)*, Chandigarh, India, 2014, pp. 1–4, doi: [10.1109/RAECS.2014.6799559](https://doi.org/10.1109/RAECS.2014.6799559).
- [37] B. Tams, "Absolute fingerprint pre-alignment in minutiae-based cryptosystems," in *Proc. Int. Conf. BIOSIG Special Interest Group (BIOSIG)*, Sep. 2013, pp. 1–12.
- [38] B. Tams, J. Merkle, C. Rathgeb, J. Wagner, U. Korte, and C. Busch, "Improved fuzzy vault scheme for alignment-free fingerprint features," in *Proc. Int. Conf. Biometrics Special Interest*, 2015, pp. 1–28, doi: [10.1109/BIOSIG.2015.7314608](https://doi.org/10.1109/BIOSIG.2015.7314608).
- [39] N. Tantubay and J. Bharti, "An efficient biocrypto-system using least square polynomial curve fitting with interpolation based new chaff-points generation method," *Adv. Electr. Comput. Eng.*, vol. 21, no. 3, pp. 21–30, 2021, doi: [10.4316/aeece.2021.03003](https://doi.org/10.4316/aeece.2021.03003).
- [40] U. Uludag and A. Jain, "Securing fingerprint template: Fuzzy vault with helper data," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2006, pp. 1–23, doi: [10.1109/cvprw.2006.185](https://doi.org/10.1109/cvprw.2006.185).
- [41] K. Xi and J. Hu, "Biometric mobile template protection: A composite feature based fingerprint fuzzy vault," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–5, doi: [10.1109/ICC.2009.5198785](https://doi.org/10.1109/ICC.2009.5198785).
- [42] L. You, L. Yang, W. Yu, and Z. Wu, "A cancelable fuzzy vault algorithm based on transformed fingerprint features," *Chin. J. Electron.*, vol. 26, no. 2, pp. 236–243, Mar. 2017, doi: [10.1049/cje.2017.01.009](https://doi.org/10.1049/cje.2017.01.009).
- [43] R. Zhou, S. Sin, D. Li, T. Isshiki, and H. Kunieda, "Adaptive SIFT-based algorithm for specific fingerprint verification," in *Proc. Int. Conf. Hand-Based Biometrics*, Nov. 2011, pp. 1–6, doi: [10.1109/ICHB.2011.6094354](https://doi.org/10.1109/ICHB.2011.6094354).
- [44] J. Zouari and M. Hamdi, "Enhanced fingerprint fuzzy vault based on distortion invariant minutiae structures," in *Proc. 7th Int. Conf. Sci. Electron., Technol. Inf. Telecommun. (SETIT)*, Dec. 2016, pp. 491–495, doi: [10.1109/SETIT.2016.7939920](https://doi.org/10.1109/SETIT.2016.7939920).
- [45] R. Cappelli, M. Ferrara, A. Franco, and D. Maltoni, "Fingerprint verification competition 2006," *Biometric Technol. Today*, vol. 15, nos. 7–8, pp. 7–9, Jul. 2007.
- [46] M. I. Hashem and K. Alibraheemi, "Literature survey: Biometric cryptosystems based on fingerprint processing techniques," in *Proc. Int. Conf. Data Sci. Intell. Comput. (ICDSIC)*, Nov. 2022, pp. 198–201, doi: [10.1109/ICDSIC56987.2022.10076184](https://doi.org/10.1109/ICDSIC56987.2022.10076184).
- [47] A. Sedik, A. A. El-Latif, M. El-Affendi, and H. Mostafa, "A cancelable biometric system based on deep style transfer and symmetry check for double-phase user authentication," *Symmetry*, vol. 15, no. 7, p. 1426, Jul. 2023, doi: [10.3390/sym15071426](https://doi.org/10.3390/sym15071426).
- [48] C. Rathgeb, B. Tams, J. Merkle, V. Nesterowicz, U. Korte, and M. Neu, "Multi-biometric fuzzy vault based on face and fingerprints," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Ljubljana, Slovenia, Sep. 2023, pp. 1–10, doi: [10.1109/ijcb57857.2023.10448963](https://doi.org/10.1109/ijcb57857.2023.10448963).
- [49] J. C. Bernal-Romero, J. M. Ramirez-Cortes, J. D. J. Rangel-Magdaleno, P. Gomez-Gil, H. Peregrina-Barreto, and I. Cruz-Vega, "A review on protection and cancelable techniques in biometric systems," *IEEE Access*, vol. 11, pp. 8531–8568, 2023, doi: [10.1109/ACCESS.2023.3239387](https://doi.org/10.1109/ACCESS.2023.3239387).
- [50] S. M. Abdullahi, S. Sun, B. Wang, N. Wei, and H. Wang, "Biometric template attacks and recent protection mechanisms: A survey," *Inf. Fusion*, vol. 103, Mar. 2024, Art. no. 102144, doi: [10.1016/j.inffus.2023.102144](https://doi.org/10.1016/j.inffus.2023.102144).
- [51] M. B. Akanbi, R. G. Jimoh, A. L. Imoize, J. B. Awotunde, and S. B. Abdulrahman, "Enhanced template protection algorithms based on fuzzy vault and cuckoo hashing for fingerprint biometrics," *Fusion, Pract. Appl.*, vol. 10, no. 2, pp. 8–24, 2023, doi: [10.54216/fpa.100201](https://doi.org/10.54216/fpa.100201).
- [52] J. Yang, S. Chen, and Y. Cao, "PUF-based key storage scheme using fuzzy vault," *Sensors*, vol. 23, no. 7, p. 3476, 2023, doi: [10.3390/s23073476](https://doi.org/10.3390/s23073476).



HACHEMI NABIL DELLYS received the Ph.D. degree in computer science, in 2019, with a focus on distributed and mobile computing. He has been an Associate Professor with the Higher National School of Computer Science (ESI), Algeria, since 2012. He has been a Researcher and an Educator with the Superior Military School of Material (ESM), since 2019. With a robust professional background, he doubles as a Consultant and a Trainer in computer science, offering expertise to socio-economic enterprises and governmental ministries. He has made significant contributions to scientific research in computer security and biometrics. He has authored numerous papers published in reputable international journals and conferences and actively served on conference committees and as a reviewer.



LAYTH SLIMAN (Member, IEEE) received the degree in computer engineering, the master's degree in computer science, and the Ph.D. degree from INSA Lyon, France, in collaboration with the University of the Ryukyus, Japan, and the Doctor of Science degree from Paris-Saclay University (UEVE). He has conducted research and taught computer engineering and information systems at many universities, including INSA Lyon; the University of the Ryukyus; Beijing University of Technology; and the Institute of Visual Informatics, Malaysia. Since September 2010, he has been a Research Professor with EFREI. He is a Research Fellow with many prestigious research institutes in Japan, France, USA, and Malaysia. He is a member of the National Accreditation Board of Engineering Higher Institutes, France. He has chaired and/or organized more than ten international conferences. He has been the Managing Editor of *Applied Soft Computing*, since 2020.



BRENDAN TRAN MORRIS (Member, IEEE) received the Ph.D. degree from the University of California at San Diego, in 2010. He is currently an Associate Professor of electrical and computer engineering and the Founding Director of the Real-Time Intelligent Systems Laboratory, University of Nevada, Las Vegas, Las Vegas, NV, USA. His research interests include computationally efficient systems that utilize computer vision techniques for activity analysis, situational awareness, and scene understanding.



KARIMA BENATCHBA received the Ph.D. degree in computer science from École nationale Supérieure d'Informatique, Algeria, in 2005. She is currently a Professor with École nationale Supérieure d'Informatique. Her research interests include combinatorial optimization, metaheuristics, biomimetic methods, and biometrics.

...