

# A Geometric Transformation to Protect Minutiae-Based Fingerprint Templates

Yagiz Sutcu<sup>a</sup>, Husrev T. Sencar<sup>b</sup> and Nasir Memon<sup>b</sup>

<sup>a</sup>Polytechnic University, Electrical & Computer Engineering Dept., Brooklyn, NY, USA;

<sup>b</sup>Polytechnic University, Computer & Information Science Dept., Brooklyn, NY, USA

## ABSTRACT

The increasing use of biometrics in different environments presents new challenges. Most importantly, biometric data are irreplaceable. Therefore, storing biometric templates, which is unique to individual user, entails significant security risks. In this paper, we propose a geometric transformation for securing the minutiae based fingerprint templates. The proposed scheme employs a robust one-way transformation that maps geometrical configuration of the minutiae points into a fixed-length code vector. This representation enables efficient alignment and reliable matching. Experiments are conducted by applying the proposed method on a synthetically generated minutiae point sets. Preliminary results show that the proposed scheme provides a simple and effective solution to the template security problem of the minutiae based fingerprint.

**Keywords:** Biometrics, template security, fingerprints, minutiae

## 1. INTRODUCTION

One of the latest technologies for authentication systems is the use of biometric modalities like fingerprints, iris data, face and voice characteristics. It is known that biometric data uniquely represent their owner and have much higher entropy as compared to ordinarily chosen passwords and PINs.<sup>1</sup> In addition, biometric features cannot be stolen, forgotten or duplicated easily. Due to these properties, biometrics based authentication systems are becoming widely used. Despite the inherent qualities, biometrics has its limitations. Most notably, biometric data are irreplaceable, they exhibit considerable variability, and they are subject to imperfect data acquisition.

Typically, biometrics based authentication systems store the copies of the biometric templates in a central database and/or in the portable devices (such as smartcards, tokens, etc.) and employ a matching algorithm to decide if the queried biometric data is legitimate or not. However, widespread deployment of biometrics based authentication systems raise new security concerns. Today, the most important problem facing the use of biometrics is the potential for the compromise of the biometric templates, namely, template security problem. The variability of the biometrics renders solutions based on cryptographic hashes, which are designed to have good diffusion properties, inappropriate for securing the template.

In the context of biometric authentication, one way to deal with this problem is by designing a robust and one-way transformation. Since the error tolerance is essential when biometric data are considered, this transformation should be robust enough to map (if possible) all possible noisy versions of the original biometric data to the same, unique output value. Depending on the application scenario, this requirement can be relaxed and some amount of variation at the output value can be tolerated by some appropriate mechanisms such as quantization. In addition to the robustness property, these transformations should not reveal much information about the original biometric data. That is, the transformations should be non-invertible (one-way) in the sense that, given the output, it should be infeasible to find an input that results in that output.

In this paper, we propose a robust one-way transformation example for securing the minutiae based fingerprint templates. Our scheme basically transforms the geometrical configuration of the minutiae points into a fixed-length feature vector and then uses this feature vector representation efficiently for alignment and matching

---

Further author information: (Send correspondence to Yagiz Sutcu)

Yagiz Sutcu: E-mail: ygzstc@yahoo.com

Husrev T. Sencar: E-mail: taha@isis.poly.edu

Nasir Memon: E-mail: memon@poly.edu

purposes. The rest of the paper is organized as follows. In Section 2, a survey of some of the major works proposed in order to solve the biometric template security problem will be provided. In Section 3, the proposed scheme is introduced and details of the construction are provided. In Section 4, we elaborate on the experimental setup and present/discuss the performance results. Our conclusions and the scope of future work are provided in the last section, Section 5.

## 2. RELATED WORK

Fingerprints are one of the most widely used biometric modality today. Approaches to fingerprint identification/verification problem can be divided into two main categories such as image correlation based techniques and structural matching based methods.<sup>2</sup> The image correlation based techniques apply a global pattern matching algorithm to an enrolled fingerprint and the queried fingerprint captured by the sensor. After the correct alignment of these two images, they are compared for similarity. Generally, correlation based methods require less computation. However they are less robust against image distortions which are very likely due to the nature of fingerprint capturing process. In structural feature matching based methods, a set of minutiae points (ridge endings and bifurcations in the ridge patterns) of the fingerprint is considered as the descriptive feature set, and at the matching stage, the minutiae set extracted from the input fingerprint image are compared with the template stored in a central database and/or a portable device.

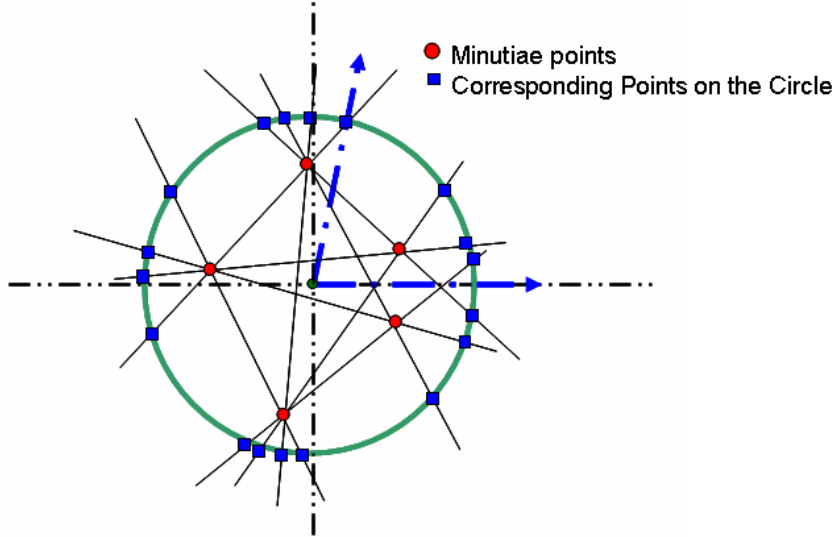
Fuzzy commitment scheme<sup>3</sup> is one of the earliest formal approaches to error tolerance. The basic idea in<sup>3</sup> is that, a secret key is chosen by the user and then encoded using a standard error correcting code (ECC). This encoded secret key is xored with the biometric template to ensure the security of the template and then stored in the database. During verification, the biometric data is xored with the values stored in the database. If the biometric data is close to the one presented at the enrollment stage, the authenticator will be able to correct some of the errors (present in the newly measured biometric data) and secret key will be retrieved correctly and revealed to the user. However, one of the major shortcomings of this scheme is that representations of the biometric data need to be ordered.

Later, to address the problem of unordered feature representations (e.g., the minutiae representation of fingerprints), Juels and Sudan<sup>4</sup> proposed the "fuzzy vault" scheme. The "fuzzy vault" scheme combines the polynomial reconstruction problem with ECC. Although authors provided a detailed analysis of the theoretical bounds on the security provided by the scheme, their construction assumes discrete valued data without any noisy perturbation, thereby making the approach unsuitable for noisy biometric data. Clancy et al. modified the fuzzy vault scheme by incorporating a quantization step (fingerprint vault<sup>5</sup>) considering minutiae representation of fingerprints and they provided optimal operating parameters under various attack scenarios. However, it should be noted that, successful operation of this scheme requires near-perfect pre-alignment of the fingerprints.

Yang and Verbauwhede<sup>6</sup> proposed a rotation and translation invariant feature representation by considering polar coordinate system. Relative positions of minutiae pairs are represented by a six-dimensional feature vectors and a distance-based similarity measure is used to compare fingerprints. Although the provable security of the fuzzy vault scheme, their performance results are poor despite the fact that the database they considered consists of 10 samples per finger from 10 different fingers, forming only 100 fingerprints.

The concept of cancelable biometrics was first introduced by Ratha et al.<sup>7</sup> and recently, a case study for fingerprints<sup>8</sup> is presented. The underlying idea of the approach is to store a transformed version of the template through the use of a fixed non-invertible transformation which is applied directly at the sensor. This approach gives the opportunity to cancel that template and corresponding transformation when the biometric data and/or transformations are compromised. However, in case of any compromise, it is not a trivial job to design another non-invertible transformation which also depends on the type of the biometric data considered.

One method to hide biometric templates and allow robust authentication at the same time is to use recently proposed secure sketch schemes.<sup>9</sup> In such a scheme, a sketch  $P$  is computed from the original template  $X$ , and when another sample  $Y$  of the same biometrics is obtained,  $X$  can be recovered from  $P$  and  $Y$ , if  $Y$  is similar to  $X$  according to some similarity measure. Such a sketch  $P$  is secure if it contains little information about the original biometric template  $X$  of the user. The secure sketch for point sets<sup>10</sup> is probably the first rigorous approach to similarity measures that do not define a metric space. A generic scheme is proposed in<sup>10</sup> for sets



**Figure 1.** Illustration of the proposed transformation

in bounded discrete  $d$ -dimensional space for any  $d$ , where the underlying similarity measure is motivated by fingerprint templates.

In addition to the methods/techniques mentioned above, there are many other approaches which address the template security problem. Ang et al. proposed a method<sup>11</sup> which creates cancelable fingerprint templates. Depending on the user-specific key (which are two parameters defining a line), positions of the extracted minutiae points are reflected and then matching algorithm is employed in this new space. However, they also noted that, sensitive dependence on the key, small key-space and poor performance of the proposed scheme needs further improvement.

Tulyakov et al. proposed a set of symmetric hash functions<sup>12</sup> for minutiae based fingerprint templates. Although their construction makes it possible to estimate the transformation (rotation and translation) parameters which connect two different scan of the same finger, it is not feasible to consider all minutiae points of a fingerprint (which cause an exponential amplification of small errors). For matching purposes, they employed some localized matching.

Teoh et al. proposed a two factor authentication scheme called biohashing.<sup>13</sup> The basic idea of this approach is to create an orthogonal basis using a tokenized random number to project the feature vectors and then thresholding them to obtain a binary hash value. Although high performance is achieved for faces, fingerprints and palmprints, their results are obtained under the assumption that tokenized random numbers cannot be stolen and used by an impostor. More detailed analysis of this weakness is elaborated in<sup>14</sup> and<sup>15</sup>

### 3. PROPOSED SCHEME

In most of the minutiae based approaches, the match between two fingerprints is decided based on a comparison of two sets of minutiae points. Despite the compactness of minutiae based representation, matching algorithms generally require an alignment or minutiae pairing (even triangulation) beforehand. Furthermore, the matching algorithm needs to be robust against global and local distortions. These distortions are the result of imperfect data acquisition process which cause global or individual changes in the orientation or yield a change in the extracted number of minutiae points. This problem is further complicated due to the need to secure the template in the presence of above mentioned sources of distortion.

In our scheme, we treat the set of minutiae points extracted from a fingerprint as a set of two-dimensional vectors denoted as

$$F_i = \{(x_1, y_1)_i, (x_2, y_2)_i, \dots, (x_n, y_n)_i\} \quad (1)$$

where  $n$  is the number of extracted minutiae points and index  $i$  denotes the user. These data points are then mapped onto a circle, which encompasses all the data points, via a one-to-many transformation. Hence, the minutiae points extracted from a fingerprint are represented by a fixed length code vector, called *fingerprint code*. With the proposed scheme, the matching of two fingerprints depends on the distance between the corresponding fingerprint codes. Furthermore, due to nature of the deployed transformation, determining minutiae positions of the fingerprint from a given fingerprint code is extremely difficult. Basic steps of the proposed transformation are as follows:

- *Extraction*: Extract the set of minutiae points from the fingerprint image.
- *Mapping*: Calculate the centroid point of the minutiae set  $F_i$  and draw a circle centered at this centroid point with radius  $R$ . The equation of the circle is:

$$(x - x_{c,i})^2 + (y - y_{c,i})^2 = R^2 \quad (2)$$

where  $(x_{c,i}, y_{c,i})$  is the centroid point of the user  $i$  calculated as

$$x_{c,i} = (\sum_{k=1}^n x_k) / n \quad (3)$$

and

$$y_{c,i} = (\sum_{k=1}^n y_k) / n \quad (4)$$

For every pair of minutiae point in the set  $F_i$ , if the distance between that pair of points is greater than a predefined threshold value,  $T$ , draw a line which passes through these two minutiae points and determine the two points of intersection between the circle and the line.

- *Quantization*: Organize intersection points on the circle into bins according to their position on the circle where each bin is generated by partitioning the circle into arcs of  $\Delta$  degrees.
- *Code generation*: The number of points in each bin will be concatenated together to create the fingerprint code of size  $360/\Delta$ .

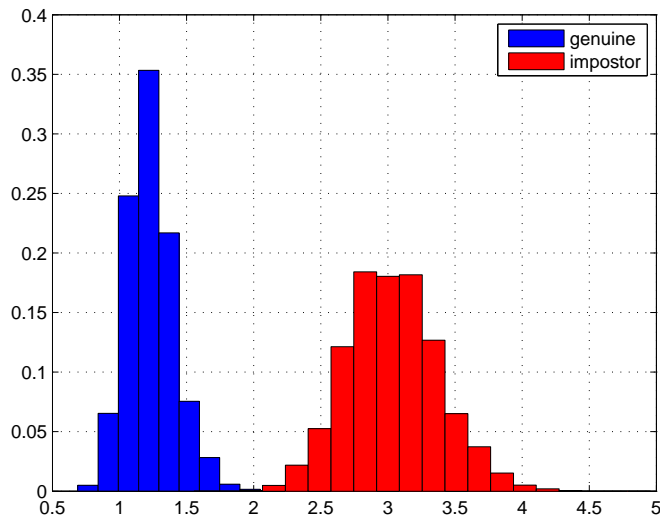
A simple example of this process is illustrated in Figure 1. It can be seen that when  $\Delta$  is set to 90 degrees, there are 4 bins and the fingerprint code is obtained as [4 6 5 3].

Verification then will be realized by comparing the stored fingerprint code against the extracted code from the query fingerprint. It should be noted any problem concerning fingerprint alignment can be easily detected and corrected with the proposed technique since misalignments will cause cyclic shifts on the resultant fingerprint code. In this study, we adopted mean absolute error as a measure of similarity for comparing two fingerprint codes. A query fingerprint is considered as legitimate if the mean absolute error between the fingerprint code extracted from the query fingerprint and the template is smaller than some pre-selected threshold value,  $t$ .

## 4. EXPERIMENTS AND RESULTS

Typically, the performance of biometric systems is measured in terms of false acceptance rate (FAR) and false rejection rate (FRR). FAR measures the ratio of impostors who are incorrectly accepted as legitimate users, while FRR measures the ratio of valid users who are rejected as impostors. Since there are two distributions (legitimate and impostor), ROC curve can be easily obtained by varying the threshold value  $t$  which designates whether an acquired biometric data is accepted as a legitimate one or rejected as an impostor. The rate at which both accept and reject errors are equal is called the equal error rate (EER) and this value is used to evaluate the accuracy of the biometric systems. It is clear that the lower the EER, the more accurate the system will be.

In our experiments, we generated 800 synthetic fingerprints each of which consists of 20 randomly generated minutiae points. The minutiae locations are considered to be two-dimensional vectors (in  $x$  and  $y$  coordinates)



**Figure 2.** Genuine and Impostor Distributions

with values uniformly varying in the range  $[0, 255]$ . Since the noise introduced during finger scanning usually leads to small perturbation of the minutiae point along with removal and/or addition of some minutiae points, we generated two different test datasets to observe the effects of these two different type of distortions separately. For the first test dataset, we generated 20 noisy versions of each and every fingerprint, by moving every minutiae points in x and/or y directions with a maximum displacement of five pixels. That is, all of the minutiae locations can differ by up to 10 pixels for the same fingerprint. For the second dataset, we generated 20 noisy versions of each and every fingerprint, by removing some of the minutiae points from the original fingerprint or by adding some new minutiae points to the original fingerprint. Maximum number of minutiae points to be added or deleted is limited to 2 points. Furthermore, for alignment purposes, we generated another test dataset by randomly rotating every fingerprint up to 30 degrees in both (clockwise and counter-clockwise) directions.

In our simulation, 20 test data for every fingerprint is used to generate  $800 \times 20 = 16000$  genuine authentication attempts and  $800 \times 799 \times 20 = 12784000$  impostor authentication attempts (20 attempts by 799 remaining users for every fingerprint in the database) and this experiment is repeated for both test datasets. In this setup, we fixed the radius of the circle to 150 pixels ( $R = 150$ ) and we tested our method with different levels of quantization,  $\Delta$ , and distance threshold values,  $T$ , which are used to determine acceptable minutiae pairs.

As a measure of closeness between the template and query fingerprint, we used mean absolute error between the template code and the query fingerprint code. By varying the closeness threshold,  $t$ , between two different fingerprint codes, we obtained corresponding ROC curves for different values of  $\Delta$  and  $T$ . (Figure 2 shows an example of normalized genuine and impostor error distributions obtained from the first test dataset with  $\Delta=8$  and  $T=80$ .) Table 1, 2 and 3 show the variation of the EER value (determined from ROC curves) as a function of distance threshold parameter,  $T$ , for different levels of quantization for the three test datasets, respectively.

As can be seen from the Table 1, size of the quantization bins,  $\Delta$ , and distance threshold,  $T$ , determines the performance of the scheme together. For small values of the quantization step, performance of the scheme in terms of EER is worse. As  $\Delta$  increases, EER value becomes smaller and reaches its minimum value at  $\Delta=8$  which means that the whole circle is divided into 45 bins with the size of 8 degrees each. If we increase the bin size further, performance of the scheme starts getting worse again. Also a similar pattern is observed for the distance threshold parameter,  $T$ . For small values of the  $T$ , the performance of the scheme is poor and as  $T$  increases, EER value achieves its minimum value. Similarly, further increase in the value of  $T$  worsens the performance slightly.

**Table 1.** Performance of the proposed scheme for different parameter values for the first test dataset

<i>EER</i>	$T = 0$	$T = 20$	$T = 40$	$T = 60$	$T = 80$	$T = 100$	$T = 120$
$\Delta = 1$	$1.40 \times 10^{-2}$	$1.41 \times 10^{-2}$	$1.24 \times 10^{-2}$	$1.12 \times 10^{-2}$	$1.36 \times 10^{-2}$	$1.84 \times 10^{-2}$	$3.24 \times 10^{-2}$
$\Delta = 2$	$4.93 \times 10^{-3}$	$4.35 \times 10^{-3}$	$5.14 \times 10^{-3}$	$3.52 \times 10^{-3}$	$3.66 \times 10^{-3}$	$3.37 \times 10^{-3}$	$5.33 \times 10^{-3}$
$\Delta = 4$	$9.77 \times 10^{-4}$	$9.15 \times 10^{-4}$	$8.12 \times 10^{-4}$	$6.43 \times 10^{-4}$	$4.01 \times 10^{-4}$	$4.44 \times 10^{-4}$	$5.74 \times 10^{-4}$
$\Delta = 8$	$2.08 \times 10^{-4}$	$1.40 \times 10^{-4}$	$1.22 \times 10^{-4}$	<b><math>0.67 \times 10^{-4}</math></b>	$1.38 \times 10^{-4}$	$1.50 \times 10^{-4}$	$1.87 \times 10^{-4}$
$\Delta = 12$	$2.50 \times 10^{-4}$	$2.50 \times 10^{-4}$	$2.61 \times 10^{-4}$	$2.19 \times 10^{-4}$	$1.88 \times 10^{-4}$	$2.41 \times 10^{-4}$	$2.50 \times 10^{-4}$
$\Delta = 18$	$4.26 \times 10^{-4}$	$4.25 \times 10^{-4}$	$4.62 \times 10^{-4}$	$2.40 \times 10^{-4}$	$2.12 \times 10^{-4}$	$2.28 \times 10^{-4}$	$2.22 \times 10^{-4}$

**Table 2.** Performance of the proposed scheme for different parameter values for the second test dataset

<i>EER</i>	$T = 20$	$T = 40$	$T = 60$	$T = 80$	$T = 100$	$T = 120$	$T = 140$
$\Delta = 1$	$1.13 \times 10^{-1}$	$1.11 \times 10^{-1}$	$1.09 \times 10^{-1}$	$1.09 \times 10^{-1}$	$1.15 \times 10^{-1}$	$1.26 \times 10^{-1}$	$1.66 \times 10^{-1}$
$\Delta = 2$	$5.83 \times 10^{-2}$	$5.88 \times 10^{-2}$	$5.58 \times 10^{-2}$	$5.53 \times 10^{-2}$	$6.28 \times 10^{-2}$	$6.90 \times 10^{-2}$	$8.49 \times 10^{-2}$
$\Delta = 4$	$2.51 \times 10^{-2}$	$2.22 \times 10^{-2}$	$2.18 \times 10^{-2}$	$2.20 \times 10^{-2}$	$2.24 \times 10^{-2}$	$2.80 \times 10^{-2}$	$3.38 \times 10^{-2}$
$\Delta = 8$	$8.93 \times 10^{-3}$	$8.62 \times 10^{-3}$	$9.05 \times 10^{-3}$	$8.97 \times 10^{-3}$	$9.93 \times 10^{-3}$	$1.07 \times 10^{-2}$	$1.42 \times 10^{-2}$
$\Delta = 12$	$1.06 \times 10^{-2}$	$9.51 \times 10^{-3}$	$9.44 \times 10^{-3}$	$9.23 \times 10^{-3}$	$8.82 \times 10^{-3}$	$9.97 \times 10^{-3}$	$1.18 \times 10^{-2}$
$\Delta = 18$	$9.39 \times 10^{-3}$	$9.03 \times 10^{-3}$	$9.36 \times 10^{-3}$	$9.48 \times 10^{-3}$	$8.40 \times 10^{-3}$	<b><math>8.08 \times 10^{-3}</math></b>	$1.01 \times 10^{-2}$
$\Delta = 24$	$1.71 \times 10^{-2}$	$1.70 \times 10^{-2}$	$1.68 \times 10^{-2}$	$1.69 \times 10^{-2}$	$1.62 \times 10^{-2}$	$1.60 \times 10^{-2}$	$1.65 \times 10^{-2}$

The main reason behind observing such a pattern can be explained as follows: Small values of  $\Delta$  and  $T$  decrease the robustness of the scheme in the sense that, if the bin size (for quantization) is set to a small value, the scheme becomes more sensitive to the noise and as a result, the performance gets poorer. Similarly, small distance threshold values also amplify the measurement noise because the intersection points due to line segments connecting relatively close pairs of minutiae points tend to displace more on the circle (as compared to the ones which connect relatively far pairs of minutiae points). Although larger bin size and distance threshold values increase the robustness against the measurement noise, they also cause performance degradation if they are increased too much. Since the entropy of the fingerprint code decreases as the quantization bin size becomes larger, scheme's ability to discriminate different fingerprints reduces as well. Similarly, larger distance threshold values eliminate relatively more minutiae pairs (less number of lines to consider) which in fact means throwing away some useful information that the original fingerprint possesses, thereby decreasing the performance of the proposed scheme.

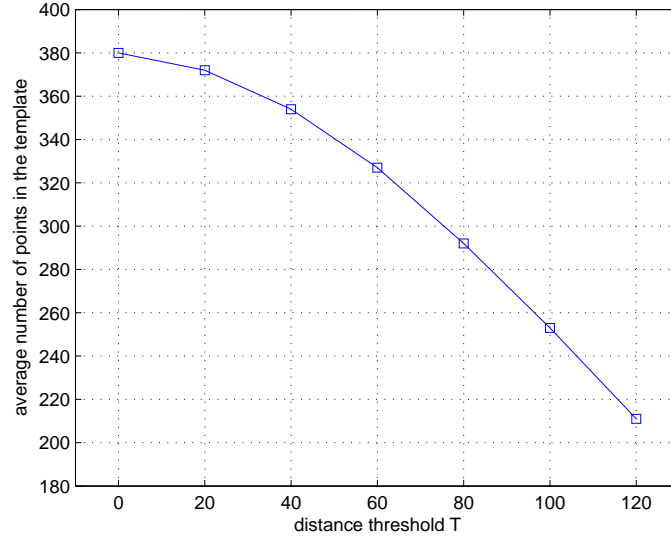
A similar pattern for the variation of the EER value can also be observed for the second and the third test datasets (Table 2 and 3). However, as expected, the overall performance of the scheme (in terms of EER) is worse in these cases, especially in the case where minutiae addition/deletion is considered (Table 2). This is basically due to the fact that, unlike the small perturbation of the minutiae points, addition and removal of new minutiae points yield more error in the fingerprint code. Intuitively, to be able to tolerate such errors, quantization bin size and distance threshold values need to be set to relatively larger values than the first case and our simulation results agree on this intuition as well. Best performance (in terms of EER) for this case is achieved at the point where  $\Delta=18$  and  $T=120$ .

Another important consideration (aside from performance) in biometric based authentication systems is the template security. The proposed scheme provides a simple and efficient solution to this problem. Security of the template relies on the following two facts: (1) the difficulty in inverting the proposed transformations to estimate the original locations of the minutiae points; and (2) the information loss due to binning which further conceals the exact locations of the intersection points on the circle.

To see the first fact more clearly, let the total number of minutiae points extracted from a fingerprint is  $n$ .

**Table 3.** Performance of the proposed scheme for different parameter values for the third test dataset

$EER$	$T = 0$	$T = 20$	$T = 40$	$T = 60$	$T = 80$	$T = 100$	$T = 120$
$\Delta = 1$	$2.48 \times 10^{-2}$	$2.91 \times 10^{-2}$	$2.61 \times 10^{-2}$	$2.55 \times 10^{-2}$	$2.87 \times 10^{-2}$	$3.04 \times 10^{-2}$	$4.67 \times 10^{-2}$
$\Delta = 2$	$6.04 \times 10^{-3}$	$5.83 \times 10^{-3}$	$5.98 \times 10^{-3}$	$4.42 \times 10^{-3}$	$4.23 \times 10^{-3}$	$3.85 \times 10^{-3}$	$6.02 \times 10^{-3}$
$\Delta = 4$	$1.47 \times 10^{-3}$	$1.01 \times 10^{-3}$	$9.23 \times 10^{-4}$	$7.02 \times 10^{-4}$	$4.52 \times 10^{-4}$	$4.79 \times 10^{-4}$	$6.34 \times 10^{-4}$
$\Delta = 8$	$3.65 \times 10^{-4}$	$2.41 \times 10^{-4}$	$1.98 \times 10^{-4}$	$1.32 \times 10^{-4}$	$1.56 \times 10^{-4}$	$1.73 \times 10^{-4}$	$2.26 \times 10^{-4}$
$\Delta = 12$	$3.19 \times 10^{-4}$	$3.16 \times 10^{-4}$	$2.86 \times 10^{-4}$	$2.61 \times 10^{-4}$	<b><math>1.25 \times 10^{-4}</math></b>	$2.68 \times 10^{-4}$	$2.98 \times 10^{-4}$
$\Delta = 18$	$5.35 \times 10^{-4}$	$5.12 \times 10^{-4}$	$4.88 \times 10^{-4}$	$2.42 \times 10^{-4}$	$2.32 \times 10^{-4}$	$2.87 \times 10^{-4}$	$2.82 \times 10^{-4}$



**Figure 3.** Variation of the average number of points in the template for different distance threshold values

Then, the number of points on the circle,  $k$ , satisfies

$$k \leq \binom{n}{2} \times 2 \quad (5)$$

depending on the value of the distance threshold and the equality holds when every pair of minutiae points are considered (in other words, when  $T=0$ ). The variation of the average number of points in the template for different distance threshold values (for our template database) is given in Figure 3. When there is no binning (exact locations of the intersection points are known), the number of possible configuration of intersecting line segments, which can be determined by brute-force search, can be calculated as

$$M = \binom{k}{2} \binom{k-2}{2} \dots \binom{2}{2} = k!/2^{k/2} \quad (6)$$

It should be noted that some of the possible configurations are not reasonable and may be eliminated from the brute-force search space easily. On the other hand, for some other configurations, depending on the intersection pattern of the lines, there will be more than one possible set of locations for the  $n$  minutiae points. (In other words, the lines may intersect with each other at more than  $n$  points and it is not possible to determine which intersection points correspond to the minutiae points). Furthermore when intersection points (on the circle) are quantized into bins, every possible placement of points in a bin increases the number of possibilities, thereby

making brute-force attack much more difficult. Although it is not trivial to measure the entropy loss of the proposed scheme, for large  $k$  and appropriately chosen  $\Delta$ , the number  $M$  in (6) can be made arbitrarily large. However, it should be noted that, the security and entropy loss aspects of the proposed transformation under different attack scenarios other than the brute-force attack require further investigation.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a scheme for protecting minutiae based fingerprint templates. Simply, every pair of extracted minutiae, that are far enough from each other, is represented by two points on a circle centered at the centroid of the minutiae set. These points are then binned together by dividing the circle into arcs, and the resulting bin values are used in generation of the fixed length code. Preliminary results show that the proposed method provides simple yet practical solution to template security problem for minutiae based fingerprint templates.

Our current efforts progress in two main directions. In the first one, we are extending our experiments to consider real fingerprint data to obtain real-life performance bounds of the proposed algorithm. This part will also include the analysis of some alternative geometrical shapes (such as; ellipse) other than circle with some other placement strategies (such as; multiple shapes with multiple placements). In the second one, we are further investigating the security and entropy loss aspects of the proposed scheme under different attack scenarios and more rigorous security analysis will be undertaken.

## REFERENCES

1. L. O’Gorman, “User authenticators: Comparing passwords, tokens and biometrics,” in *Proceedings of the IEEE*, 2 **91**, IEEE, 2003.
2. K. Uchida, “Fingerprint identification,” in *NEC Journal of Advanced Technology*, **2**, 2005.
3. A. Juels and M. Wattenberg, “A fuzzy commitment scheme,” in *Proc. ACM Conf. on Computer and Communications Security*, pp. 28–36, 1999.
4. A. Juels and M. Sudan, “A fuzzy vault scheme,” in *IEEE Intl. Symp. on Information Theory*, 2002.
5. T. Clancy, N. Kiyavash, and D. Lin, “Secure smartcard-based fingerprint authentication,” in *ACM Workshop on Biometric Methods and Applications*, 2003.
6. S. Yang and I. Verbauwhede, “Automatic secure fingerprint verification system based on fuzzy vault scheme,” in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 609–612, 2005.
7. N. Ratha, J. Connell, and R. Bolle, “Enhancing security and privacy in biometrics-based authentication systems,” *IBM Systems Journal* **40**(3), pp. 614–634, 2001.
8. N. Ratha, J. Connell, R. Bolle, and S. Chikkerur, “Cancelable biometrics: A case study in fingerprints,” in *18th International Conference on Pattern Recognition, ICPR 2006*, **4**, 2006.
9. Y. Dodis, L. Reyzin, and A. Smith, “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” in *Eurocrypt, LNCS 3027*, pp. 523–540, Springer-Verlag, 2004.
10. E.-C. Chang and Q. Li, “Hiding secret points amidst chaff,” in *Eurocrypt 2006*, 2006.
11. R. Ang, R. Safavi-Naini, and L. McAven, “Cancelable key-based fingerprint templates,” in *Lecture Notes in Computer Science, LNCS*, 2005.
12. S. Tulyakov, F. Farooq, and V. Govindaraju, “Symmetric hash functions for fingerprint minutiae,” in *Lecture Notes in Computer Science, LNCS*, 2005.
13. A. B. Teoh, D. C. Ngo, and A. Goh, “Biohashing: two factor authentication featuring fingerprint data and tokenized random number,” *Pattern Recognition Letters* **37**, 2004.
14. A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You, “An analysis of biohashing and its variants,” *Pattern Recognition* **39**(7), 2006.
15. K.-H. Cheung, A. Kong, D. Zhang, M. Kamel, and J. You, “Revealing the secret of facehashing,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **3832**(7), 2006.