Enhanced Multi-line Code for Minutiae-Based Fingerprint Template Protection

Wei Jing Wong\textsuperscript{a}, Andrew B. J. Teoh\textsuperscript{b}, M. L. Dennis Wong\textsuperscript{a,1,}\textsuperscript{*}, Yau Hee Kho\textsuperscript{a}

\textsuperscript{a}Faculty of Engineering, Computing and Science, Swinburne University of Technology (Sarawak Campus), 93350 Kuching, Malaysia.

\textsuperscript{b}Electrical and Electronic Engineering, College of Engineering, Yonsei University, Seoul, South Korea.

Abstract

In this paper, we propose a cancellable fingerprint template technique based on our previous work on multi-line code (MLC) (Wong et al., 2012). The modification and improvement focuses on the change of MLC values and the generation of binary MLC. In addition, an enhanced similarity measure is also proposed to compensate the loss in accuracy for binary MLC, called the dynamically weighted integrated Dice (DWID) similarity. Comprehensive experiments on three FVC datasets are carried out to compare the performance among different settings of MLC. The lowest equal error rate (EER) obtained in the stolen-key scenario is 1.93% for FVC2002 DB1. Besides, analysis on the revocability, non-reversibility and template size of the enhanced MLC have been presented.

Keywords: Cancellable Fingerprint Template, Multi-line Code, Dynamically Weighted Integrated Dice, Stolen-Key Scenario

1. Introduction

Biometric authentication systems offer a great range of advantages over knowledge-based and token-based authentication systems, such as passwords, user IDs, identification cards and...
PINs. Biometric authentication systems are automated authentication systems which make use of distinctive anatomical and behavioural characteristics or identifiers (e.g., fingerprints, hand geometry, face, iris, voice, signature) to identify a person. However, conventional biometric databases store the original biometric features, once compromised, suffer from permanent privacy and security issue. Therefore, biometric template protection schemes are required to shield the original biometric information. Cancellable biometrics is one of the promising candidates of biometric template protection technique. It utilizes a systematic transformation of the derived biometric features to protect the original biometric information. If a cancellable biometric template is compromised, the transformation characteristics can be changed and the user’s biometrics is mapped onto a new template, which replaces the compromised template. The three principle objectives of cancellable biometrics (Maltoni et al., 2009) are:

(i) Non-reversibility: it should be computationally infeasible to recover the original biometric data from the biometric template so that the biometric identifier can never be reconstructed even when the template is stolen.

(ii) Accuracy: the accuracy of fingerprint recognition should not deteriorate after transformation. This determines the overall performance of the system and prevents false authentication.

(iii) Diversity: no same biometric template can be used in various applications. It is also referring to revocability of the biometric template, where new template can be reissued in the event of compromise.

This paper extends our previous work (Wong et al., 2012) on multi-line code (MLC) with the intention to boost the performance of MLC concerning accuracy, security and storage size of the template. In a nutshell, we modify the original MLC in the following aspects:

(i) Change of MLC from number of minutiae in the region of interest to the mean of distances from these minutiae to the reference minutia ($P_r$).

(ii) Conversion of the real number representation to a binary code.

(iii) Application of a new similarity measure in the MLC matching.
2. Related Work

Two main categories of cancellable biometrics include non-invertible transforms and biometric salting (Jain et al., 2008). The former approach applies non-invertible transform function to the biometric data, either in signal level or feature level to make it unable to reconstruct the original biometric data even if template and transformation method are compromised. In the context of cancellable fingerprint, such approach usually implies a loss in accuracy due to the complexity in aligning the fingerprints in the transformed domain. Hence, alignment-free methods have become a trend in the research area. On the other hand, biometric salting applies transforms which are invertible (Rathgeb and Uhl, 2011). In case of user-dependent transforms, a user-specific key has to be presented at each transformation. Even though imposters are very likely to recover the biometric feature from the biometric template, the biometric feature is extracted in such a way that the original biometric pattern cannot be reconstructed.

Both non-invertible transforms and biometric salting can produce either string-based or feature vector-based cancellable template depending on the method used. In a nutshell, biometric salting may maintain the recognition accuracy of biometric authentication systems while non-invertible transforms yield higher security. In this section, we investigate several instances of non-invertible transforms and biometric salting in the context of cancellable fingerprint template.

2.1. Non-invertible Transform

Ang et al. (2005) presented a geometric transformation based on the reflection of minutiae. In this approach, a line passing through the core point is drawn, and the minutiae below the line are reflected while the minutiae above remain. The gradient of the line is determined by a user-specific key ranges from 0 to \(\pi\). Confusion occurs when dealing with fingerprints with no core point (arch) and with more than one core points (whorl). Also, since only one side of the minutiae are reflected, the final template still retains a part of the original minutiae set and thus weakens the security.

Ratha et al. (2007) proposed three kinds of non-invertible transforms including Cartesian transform, polar transform and functional transform. Cartesian transform and polar transform di-
vide the fingerprint space into cells of equal size and re-arrange the minutiae according to the cell they belong to on a many-to-one mapping basis. Functional transform applies a spatial distortion to the fingerprint space using Gaussian kernels so that the position of the minutiae are translated and rotated in the same way. However, Quan et al. (2008) pointed out that these transforms are vulnerable to attacks as most of the transformed minutiae are possible to be reversed to their original locations.

Tulyakov et al. (2005, 2007) used symmetric hash functions to convert the minutiae into hash values. In this algorithm, a minutia is represented by a complex number \( c_i \). For each minutia in the fingerprint, a triplet \( (c_i, c_j, c_k) \) is formed with its two nearest neighbouring minutiae and is hashed using predefined hash functions. A secret key is introduced to seed the choices and order of hash functions for different fingerprints. This work was extended (Kumar et al., 2010) by combining more than one hash functions during implementation to increase the security of the template. Also, \( k \)-plets of minutiae were used instead of triplets, where \( k \) can be more than three. Although it is impossible to reverse the hashed data, a large number of high power hash functions are needed to ensure the revocability of the template, which leads to high complexity.

For aforementioned approaches, matching of fingerprint templates requires pre-alignment of the minutiae. This increases the computational time during fingerprint matching and thus reduces its applicability in real-time authentication systems. One of the solutions of eliminating minutiae alignment is to use invariant features of minutiae in the generation of fingerprint template as proposed by Lee et al. (2007). These features are extracted following the same fashion used by Tico and Kuosmanen (2003). Together with a user-specific PIN, the invariant features are used to parametrize two changing functions which contribute to the transformation of the minutiae, namely the distance-changing function and the orientation-changing function. Another cancellable template utilizing invariant features based on triplets was proposed by Farooq et al. (2007a). The features measured are the length of the three sides, the orientations of the three vertex minutiae and the height of the longest side of a triplet. The template is a binary string of quantized feature values, so it requires less database storage.

Recently, Wang and Hu (2012) proposed an alignment-free cancellable fingerprint template
based on a densely infinite-to-one mapping approach. In this method, the invariant features of every minutiae pair are quantized and converted the generated binary string into a complex vector by using discrete Fourier transform. A randomly generated parametric matrix is then blended with the complex vector to obtain the final template. This approach excels in security, even when the template and the parametric key are stolen.

A novel bit-string representation of fingerprint template was introduced by Lee and Kim (2010), in which each minutia is described by a three dimensional array. The width and height of the three dimensional array is the x-y plane of the fingerprint image, whereas the depth represents the orientation of minutiae. The array is divided into cells of equal size, and the numbers of minutiae in the cells form the final bit-string. It not only eases the matching of templates, but also provides high revocability by using simple permutation. Other string-based cancellable fingerprint templates include minutiae pair representation (Jin et al., 2010), polar-based 3-tuple representation (Jin et al., 2011) and projection-based line vector (Ahmad et al., 2011).

One of the state-of-the-art fingerprint template representations is called the minutiae cylinder-code (MCC) (Cappelli et al., 2010). It forms a cylinder around a minutia and the cylinder is tessellated in the similar manner as presented by Lee and Kim (2010). Instead of counting the number of minutiae in the cells, MCC considers all minutiae around the cell within a certain range. The contribution of each minutia towards the cell value is regulated by its positional and orientation distance from the centre of the cell. Such approach uses the basis of fixed radius-based minutia descriptor without neglecting the area outside but nearby the perimeter. The relaxation approach helps to improve the recognition rate but it increases the storage requirement and computational time. Later, Ferrara et al. (2012) presented the protected MCC (p-MCC) as a template protection scheme to enhance the security of MCC.

2.2. Biometric Salting

Fingerprint salting is also called BioHashing (Teoh et al., 2004, 2008). Other than Finger-Code proposed by Jain et al. (2000), BioHashing pioneers in correlation-based biometric matching schemes. It is a two-factor transform that employs the wavelet Fourier-Mellin transform (WFMT) features of fingerprints and a user-specific tokenized key. User-dependent multi-state
discretization (Teoh et al., 2010) was used in the generation of binary bit-string to improve the performance of BioHashing specifically for stolen-token scenario. As the security of biometric salting lies with the secret key, once both the template and the key are compromised simultaneously, the protected templates become reversible. In addition, Yang et al. (2010) has suggested dynamic random projection to enhance the security of the conventional random projection used by BioHashing. The idea is to construct a random matrix dynamically, depending on the biometric feature vector itself.

By combining the concept of BioHashing (Teoh et al., 2004, 2008) and FingerCode (Jain et al., 2000), Belgeuchi et al. (2010) presented a minutiae-based fingerprint salting scheme. The proposed method extracts the FingerCode for every minutia and applies BioHashing on the FingerCode to produce a protected minutia template, named BioCode.

Another novel fingerprint salting approach was proposed by Takahashi and Hirata (2011). The transformation utilizes chip matching algorithm (Mimura et al., 2001) based on correlation-invariant random filtering (CIRF). It first extracts chip images centred at the minutiae from the fingerprint image and transform these chip images using CIRF to generate the template. The method stresses on the security and privacy of cancellable fingerprint template. The mathematical properties of CIRF were further investigated by Takahashi (2009) to derive a new algorithm for cancellable biometrics that establishes better security without affecting the accuracy.

3. Multi-line Code

In this section, we recapitulate the generation of multi-line code (MLC) (Wong et al., 2012) for the benefits of the readers. MLC is a string-based minutia descriptor used for cancellable fingerprint template. It describes a minutia by inspecting the minutiae distribution along multiple lines of different orientation intersecting at the reference minutia itself. The generation of MLC consists of two steps: MLC formulation and MLC permutation.

3.1. Formulation of MLC

Inspired by Lee’s algorithm (Lee and Kim, 2010) and MCC (Cappelli et al., 2010), the formulation of MLC inspects the fingerprint in three dimensional aspects which include the Cartesian...
plane \((x\) and \(y\) coordinates) and the orientation \((\theta)\). Taking a reference minutia, \(P_r(x_r, y_r, \theta_r)\) as an instance, the procedures to create a MLC based on the reference minutia are as follows:

1) Divide the minutiae into different angular partitions according to the relative angle between \(\theta_r\) and the orientation of the neighbour minutia. As shown in Figure 1b, the depth of the cylinders indicates the range of the angular partitions \((\Delta \varphi)\) and the depth of the entire space covers all the possible values of minutiae orientation, ranging from 0 to \(2\pi\).

2) Construct a straight line of length, \(l\) in the same direction as \(\theta_r\) in every angular dimension and take \(s\) sample points equally distributed along the line with distance, \(d\) in between one another.

3) Based on the location of the sample points marked in step 2, two semi-circle-shaped bit-wise AND masks (with radius \(r\)) are applied on the minutiae map (which indicates the location of all minutiae): one on the left side of the line and another on the right side of the line, to obtain the number of minutiae in the region. This is demonstrated in Figure 1a where different regions of semi-circles are distinguished by their shading.

4) The number of minutiae in every semi-circle is arranged sequentially according to different orientation to form a single-line code.

5) Repeat step 2 to step 4 for lines of different direction with equal angle in between and concatenate all the single-line codes to form a multi-line code. If we are using \(M\) lines to describe the reference minutia, the directions of the lines are: \(\theta_r, \theta_r + \frac{\pi}{M}, \theta_r + \frac{2\pi}{M}, \ldots, \theta_r + \frac{(M-1)\pi}{M}\). Figure 1a shows the graphical illustration of these lines when \(M = 3\). Therefore, the MLC for a minutia can be written as \(L = [a_{11}a_{12}\ldots a_{1n}, a_{21}a_{22}\ldots a_{2n}, \ldots, a_{M1}a_{M2}\ldots a_{Mn}]\), where \(N = 2 \times (\frac{M}{2} + 1) \times \frac{2\pi}{\Delta \varphi}\) is the length of a single-line code. \(a_{mn}\) is assigned an invalid value, -1 if the sample point is located outside the boundaries of the fingerprint image.

6) Repeat step 1 to step 5 for the rest of the minutiae extracted from the fingerprint to generate the template of the fingerprint. The template is an array of multi-line codes, \(T = \{L_1, L_2, \ldots, L_K\}\), where \(K\) is the total number of minutiae.

Since the lines are subject to the reference minutia, the position and direction of the lines change according to the coordinates and orientation of the reference minutia, so as the location of
the circles. Besides, the angular partition a minutia falls in is determined by its angular difference with \( \theta_r \). Hence, MLC is invariant to translation and rotation.

The robustness of MLC against local non-linear distortion depends very much on the distance between two adjacent circles \( d \) and their radius \( r \) which should be adjusted collaboratively so that the circles overlap partially with one another. The idea is to create a buffer region (where two or more circles overlap) to accommodate the close-to-border minutiae which may be excluded from the circles they used to be in due to non-linear distortion. For instance if a close-to-border minutia is slightly shifted, in the case of non-overlapping circles, the minutiae count of two adjacent regions can change from \( \{ \alpha, \beta \} \) to \( \{ \alpha - 1, \beta + 1 \} \) (considering the worst case), introducing an Euclidean distance of 2; whereas for overlapping-circles, the distance caused by such distortion in the worst case is 1 (between \( \{ \alpha, \beta \} \) and \( \{ \alpha - 1, \beta \} \)). However, this only applies to minutiae near certain parts of the circumference. The other parts, especially those orthogonal to the base line are taken care of by increasing the number of lines \( M \) so that the circles may overlap each other in all directions. The value of \( d, r \) and \( M \) used in this paper are 8, 25 and 3 respectively.

3.2. Permutation of MLC

In order to achieve revocability and diversity of the template, we introduce an external factor that offers huge variety of transformations unto the generated multi-line code. In the previous paper (Wong et al., 2012), we simply permute the code based on a user-specific secret key. The secret key is a random number that seeds the permutation order of the MLC. It is important to ensure that no two individuals or two applications of one individual can be assigned the same number.

In addition, permutation of the MLC improves the performance of cancellable fingerprint template. By introducing a unique “personality” to every fingerprint, it greatly reduces the false acceptance rate (FAR) of the system. Figure 2 explains the realization of revocability and accuracy enhancement through MLC permutation.

4. MLC Enhancement Scheme

In this section, we propose two attempts towards enhancing the performance of MLC.
206 4.1. Minutiae Contribution Measure

In this paper, we present a different measure of minutiae contribution in the stage of MLC formulation. Instead of taking the number of minutiae as the code of the region, we now generate the code by computing the mean value of distances between these minutiae and the corresponding sample points. Therefore, the MLC expression is now $L = [\bar{x}_{11}, \bar{x}_{12}, ..., \bar{x}_{1N}; \bar{x}_{21}, \bar{x}_{22}, ..., \bar{x}_{2N}; \ldots; \bar{x}_{M1}, \bar{x}_{M2}, ..., \bar{x}_{MN}]$, where $\bar{x}_{mn}$ denotes the mean distance in each region. Figure 1 gives an example of the line code generated based on the new scheme.
(a) Assume that a fingerprint is first enrolled with key 1. In the case of template database compromise, the original template can be made obsolete and replaced with a new one generated with a different secret key, key 2.

(b) Taking A and B as two multi-line codes generated from two distinct individual fingerprints. Matching A and B might result in a high score due to inter-class similarity. However, by permuting A and B with two unique secret keys, we can increase the distance between the two MLCs, and hence decrease the FAR during fingerprint matching.

Figure 2: Purposes of permuting multi-line code.

According to Farooq et al. (2007b), the accuracy of a biometric system can be increased by adding more information to the protected biometric template. In other words, it is an alteration in the trade-off between accuracy and security. Even though by using the mean distance, the imposter will have no idea about the exact number of minutiae, it may leak the information about the distribution of these minutiae within the region. In this case, the imposter needs less effort to crack the location of the minutiae through brute-force attack. Detailed analysis of the security of MLC is presented in Section 6.4. However, in return we gain accuracy by introducing higher distinctiveness to MLC.

Furthermore, another drawback of using mean distance as the minutiae contribution measure is that it is larger in size. Since the number of minutiae is always a whole number, it can be stored as a short integer, which consumes only two bytes of storage memory. On the contrary, the mean distance contains decimal point and requires four bytes of storage if it is stored in a single-precision floating-point format. This doubles the storage requirement of MLC, and thus leading to MLC binarization (Section 4.2) with the intention to complement the trade-off.
4.2. MLC Binarization

MLC binarization converts the original mean distance value representation into binary codes. This section discusses different approaches of quantizing MLC to obtain the final binary code.

4.2.1. 1-bit Binary Implementation

In 1-bit binary implementation, each number in MLC is converted into a 1-bit binary number (1 or 0). This is a rather simple thresholding process and can be formulated as:

\[
b_\alpha = \begin{cases} 
0 & \text{if } \hat{x} < \tau_b; \\
1 & \text{otherwise.}
\end{cases}
\]  

where \( b \) represents the binarized value of \( \hat{x} \), and \( \tau_b \) represents the threshold value of the conversion.

In addition, since a binary string can only contain logical 1 and logical 0, it is essential to create another binary string, same length as the original real number string to store the invalid bits (indicated by -1 in the real number representation).

4.2.2. k-bit Binary Implementation

In order to ensure that the performance of fingerprint recognition will not deteriorate after binarization, we increase the quantization bit-length to preserve more information about the original real number MLC. Instead of using only 1 bit to represent the code, we can apply \( k \)-bit binarization, where \( k \in \{\mathbb{Z} | k \geq 2\} \). In this paper, we demonstrate the employment of 2-bit binarization. Given that \( r = 25 \), the step size of a 2-bit uniform quantization is:

\[
\Delta_{b-1} = \frac{r}{2^k} = 6.25
\]  

\( (2) \)
where $B$ is the quantization bit-length. The binarized code can be expressed as:

$$b_{\beta-1} = \begin{cases} 
[00] & \text{if } c_{\beta-1} = 0; \\
[01] & \text{if } c_{\beta-1} = 1; \\
[11] & \text{if } c_{\beta-1} = 2; \\
[10] & \text{if } c_{\beta-1} = 3.
\end{cases} \quad (3)$$

where $c_{\beta-1} = \lfloor \frac{x}{\Delta_{\beta-1}} \rfloor$ is the classification rule of quantization. We choose Gray code over natural binary code as the binary sequence of the code because it is designed in such a way that the Hamming distances between two adjacent numbers are equal. However, problem arises when adopting Dice similarity (Dice, 1945) from our previous paper (Wong et al., 2012) in the matching of binary MLC. Since Dice similarity considers only the positive matches, any match involving [00] will result in a zero similarity, which is undesirable. For instance, the similarities between [00] and [01], and [01] and [11] are supposed to be identical, but instead, the calculated Dice similarities of the number pairs are 0 and 0.67 correspondingly. Hence, we propose two solutions to address the issue as described in the followings:

(i) One way to resolve the problem is by applying non-uniform quantization where [00] is only associated with 0 in the real number representation while the numbers range in (0,25) are uniformly distributed to the other three quantization levels. In this scenario, equation 2 is modified into:

$$\Delta_{\beta-2} = \frac{r}{2^B - 1} = \frac{25}{3} \quad (4)$$

and the classification rule is:

$$c_{\beta-2} = \begin{cases} 
0 & \text{if } \hat{x} = 0; \\
\lceil \frac{\hat{x}}{\Delta_{\beta-2}} \rceil & \text{otherwise.}
\end{cases} \quad (5)$$

(ii) Another solution is to use a different similarity measure that is specifically designed for binary biometric template recognition and gives credits to both positive and negative matches. This can solve the zero similarity problem concerning matchings with [00]. Examples of
such similarity or dissimilarity measures include Hamming distance, Faith similarity, Sokal & Michener similarity and AZZOO similarity (Cha et al., 2005). This is further discussed in Section 5.1.

5. Fingerprint Matching

5.1. MLC-to-MLC Matching

Given a MLC $L_e$ taken from the enrolled fingerprint template ($T^E = \{L_1 L_2 ... L_E\}$) and a MLC $L_q$ taken from the query fingerprint template ($T^Q = \{L_1 L_2 ... L_Q\}$), where $E$ and $Q$ is the total number of minutiae in $T^E$ and $T^Q$ respectively. The similarity between $L_e$ and $L_q$ signifies the likelihood of $L_q$ in $T^Q$ being a correspondence to $L_e$ in $T^E$. If either $L_e$ or $L_q$ contains more than 50% of invalid code, the pair is marked unmatchable and is assigned a zero similarity. This applies to all similarity measures discussed in this section. Otherwise, the similarity between two MLCs is calculated as (Wong et al., 2012):

$$S_{\text{Dice}} = \frac{2|L_e \cdot L_q|}{|L_e|^2 + |L_q|^2}$$

(6)

where $\cdot$ denotes a element-wise multiplication and $| \cdot |$ calculates the sum of all elements. The values of $S_{\text{Dice}}$ range from 0 to 1, of which 0 indicates a total mismatch between $L_e$ and $L_q$, whereas 1 indicates a perfect match between them.

In the context of binary MLC, we express the computation of similarity measure in Operational Taxonomic Units (OTUs): $\rho_1 = |L_e \land L_q|$, $\rho_2 = |L_e \land \bar{L}_q|$, $\rho_3 = |L_e \land \bar{L}_q|$, and $\rho_4 = |L_e \land L_q|$, where $\land$ represents a bit-wise AND operation and $L_e$ and $L_q$ are the complements of $L_e$ and $L_q$ respectively. Therefore, equation (6) can be rewritten as:

$$S_{\text{BinDice}} = \frac{2\rho_1}{2\rho_1 + \rho_2 + \rho_3}$$

(7)

As suggested in Section 4.2, we need a more appropriate similarity measure to improve the performance of $k$-bit binary MLCs. AZZOO similarity (Cha et al., 2005) has been proven to outperform other measures in binary feature vector recognition. Unlike Dice similarity which
only gives credit to positive matches ($\rho_1$). AZZOO similarity credits both positive and negative matches ($\rho_4$). The formula of weighted AZZOO (WAZZOO) which gives different weights to $\rho_1$ and $\rho_4$, is given as:

$$S_{WAZZOO} = \sigma_p\rho_1 + \sigma_n\rho_4$$  \hspace{1cm} (8)

For the ease of direct comparison between two chosen similarity measures, WAZZOO similarity is normalized so that the resulting score ranges from 0 to 1. The normalized WAZZOO (NWAZZOO) similarity is represented by:

$$S_{NWAZZOO} = \frac{\sigma_p\rho_1 + \sigma_n\rho_4}{\sigma_p\rho_1 + \rho_2 + \rho_3 + \rho_4}$$  \hspace{1cm} (9)

Furthermore, we propose an enhanced similarity measure integrated from Dice similarity, called the dynamically weighted integrated Dice (DWID) similarity. The formula is derived as:

$$S_{DWID} = \frac{\sigma_m\rho_1}{\sigma_m\rho_1 + \sigma_n(\rho_2 + \rho_3)}$$  \hspace{1cm} (10)

Similar to Dice’s, DWID similarity ignores the negative matches. Nevertheless, DWID similarity weighs $\rho_1$ dynamically so that it is not credited linearly but exponentially. In order to achieve that, we let $\sigma_m = \sigma'_m\rho_1$ while the weight for the non-matches ($\rho_2$ and $\rho_3$) in the normalizing denominator is $\sigma_n = \sigma'_n(\rho_2 + \rho_3)$. Hence equation (10) can be rewritten as:

$$S_{DWID} = \frac{\sigma'_m\rho_1^2}{\sigma'_m\rho_1^2 + \sigma'_n(\rho_2 + \rho_3)^2}$$  \hspace{1cm} (11)

5.2. Global Matching Score

When dealing with two fingerprint templates, each MLC in $T^Q$ is cross-matched with the ones in $T^E$ so that we have a similarity matrix containing similarity scores among all MLCs between two templates. Each element in the similarity matrix is then re-evaluated with the following
criterion to eliminate double-matching:

\[
S(e, q) = \begin{cases} 
S(e, q) & \text{Condition 1;} \\
0 & \text{otherwise.}
\end{cases}
\]  \hspace{1cm} (12)

Condition 1 implies that \(S(e, q) \geq \tau_s\) and \(S(e, q)\) must be the maximum among all values of \(S(e, i)\) (for \(i \in [1, Q]\)) and \(S(i, q)\) (for \(i \in [1, E]\)), where \(\tau_s\) is the lowest similarity value to say that a pair of MLCs are matchable.

To determine the overall similarity between \(T^E\) and \(T^Q\), a global matching score is used to measure the likelihood of \(T^E\) and \(T^Q\) being two instances of the same fingerprint. With the processed similarity matrix, we calculate the matching score with the following formula:

\[
MS = \frac{\sum_{e=1}^{E} \sum_{q=1}^{Q} S(e, q)}{\min(E, Q)}
\]

(13)

6. Experiments and Discussions

6.1. Experiment Setup

The proposed method is evaluated on the following datasets: FVC2002 DB1, DB2 and FVC2004 DB1, DB2. Each dataset contains 100 fingerprints, and each fingerprint has 8 samples with different ways and levels of perturbation. The first impression of every fingerprint is stored as the enrolled template and the remaining seven impressions are used as queries, and thus resulting in 70,000 tests in total, which include 700 genuine tests and 69,300 imposter tests per dataset per setting. There are eight distinct settings being tested as abbreviated below:

- \textit{mlcn-Dice} - MLC representing numbers of minutiae matched using Dice similarity.
- \textit{mlcm-Dice} - MLC representing mean distance values matched using Dice similarity.
- \textit{mlcmb1-Dice} - 1-bit binary MLC representing mean distance values matched using Dice similarity.
• *mlcmb2uq-Dice* - 2-bit uniformly quantized binary MLC representing mean distance values matched using Dice similarity.

• *mlcmb2nq-Dice* - 2-bit non-uniformly quantized binary MLC representing mean distance values matched using Dice similarity.

• *mlcmb2uq-NWAZZOO* - 2-bit uniformly quantized binary MLC representing mean distance values matched using NWAZZOO similarity.

• *mlcmb2uq-DWID* - 2-bit uniformly quantized binary MLC representing mean distance values matched using DWID similarity.

• *mlcmb2nq-DWID* - 2-bit non-uniformly quantized binary MLC representing mean distance values matched using DWID similarity.

Tests are carried out in both genuine-key scenario and stolen-key scenario. In the former scenario, we assume that the secret key is kept secure, so every MLC is permuted with its own unique key (assigned to individual fingerprint) before matching. Whereas in the stolen-key scenario, it is assumed that the secret key is compromised and becomes obsolete, so the MLCs are matched unpermuted.

### 6.2. Accuracy

In genuine-key scenario, the proposed scheme performs ideally where the EERs for all settings are 0%. On the other hand, in the case of stolen key, the EERs declined as expected. Table 1 shows the EERs of the proposed scheme as well as other minutiae distribution-based algorithms. Despite the fact that the performance of MLC is not directly comparable to the other literature due to different minutiae extraction method applied, we use Lee’s method (Lee and Kim, 2010) and p-MCC (Ferrara et al., 2012) as the references in accuracy evaluation. Figure 3 shows the ROC curves of all MLC settings zoomed in to the EER-intercept to provide a graphical comparison of the performances.

The result has proven that the application of both approaches towards enhancing MLC improves the accuracy of the original MLC. Moreover, the solutions suggested in Section 4.2.2 has
Table 1: EERs of the proposed methods and other existing algorithms for stolen-key scenario over different FVC datasets.

<table>
<thead>
<tr>
<th>Method/Setting</th>
<th>FVC2002DB1</th>
<th>FVC2002DB2</th>
<th>FVC2004DB1</th>
<th>FVC2004DB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>mlen-Dice (Wong et al., 2012)</td>
<td>4.69</td>
<td>5.03</td>
<td>10.36</td>
<td>11.05</td>
</tr>
<tr>
<td>mlen-Dice</td>
<td>3.2</td>
<td>3.53</td>
<td>8.26</td>
<td>10.4</td>
</tr>
<tr>
<td>mlenb1-Dice</td>
<td>3.26</td>
<td>4.08</td>
<td>8.88</td>
<td>11.04</td>
</tr>
<tr>
<td>mlenb2uq-Dice</td>
<td>2.37</td>
<td>3.39</td>
<td>7.38</td>
<td>10.28</td>
</tr>
<tr>
<td>mlenb2nq-Dice</td>
<td>2.27</td>
<td>2.85</td>
<td>6.92</td>
<td>9.98</td>
</tr>
<tr>
<td>mlenb2uq-NWAZOO</td>
<td>2.37</td>
<td>4.18</td>
<td>7.45</td>
<td>10.37</td>
</tr>
<tr>
<td>mlenb2uq-DWID</td>
<td>2.05</td>
<td>2.99</td>
<td>7.22</td>
<td>9.75</td>
</tr>
<tr>
<td>mlenb2nq-DWID</td>
<td>1.97</td>
<td>2.54</td>
<td>6.53</td>
<td>9.2</td>
</tr>
<tr>
<td>Lee’s method (Lee and Kim, 2010)</td>
<td>-</td>
<td>-</td>
<td>10.3</td>
<td>9.5</td>
</tr>
<tr>
<td>p-MCC (Ferrara et al., 2012)</td>
<td>1.88</td>
<td>0.99</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

successfully compensated the problem of undesirable similarity among the binary sequences, while the combination of both can reduce the EER to 1.97%, 2.54%, 6.53% and 9.2% for FVC2001DB1, FVC2001DB2, FVC2004DB1 and FVC2004DB2 respectively. However, NWAZOO similarity does not suit MLC matching probably because in binary MLC, the 1’s are sparsely distributed with too many 0’s in between and thus, the outcome is predominated by the negative matches even when it’s weight is small.

6.3. Revocability

We adopt the method described by Lee et al. (2007) to evaluate the revocability or diversity of the proposed cancellable fingerprint template. In this section, we briefly explain the experiments performed under two cases.

6.3.1. Case 1: Original Templates Versus Permutated Templates

In this scenario, we measure the improbability of a permuted template to correlate with the original template of the fingerprint. The experiment procedures are as follows:

1) One out of eight impressions for each fingerprint is used to generate 30 different permuted templates as the enrolled templates.

2) The remaining seven impressions are matched against the 30 enrolled templates.

3) Generate the distribution (hereafter denoted as $D_I$) of matchable MLCs over all fingerprints. A pair of MLCs are said to be matchable if the similarity score satisfies the condition mentioned in equation (12). Then, calculate the separability between $D_I$ and the original genuine
Figure 3: ROC curves of all settings in stolen-key scenario. We can observe that for all datasets, mlcmb2uq-DWID (marked by inverted triangle) has the lowest EER value.

Figure 4 illustrates the distribution of $D_1$ and $OG$ for the highest separability setting (mlcn-Dice) together with two low separability settings (mlcmb1-Dice and mlcmb2uq-NWAZZOO). Both figures in Table 2) and Figure 4 show that the distributions have no correlation or negligible correlation between each other even for the lowest separability setting (mlcmb2uq-NWAZZOO).
6.3.2. Case 2: Permuted Templates Versus Permuted Templates

If an enrolled template is compromised, it is replaced by a new template associated with a different permutation key. In this scenario, we measure the dissimilarity between templates from the same fingerprint permuted with different keys. The experiment is conducted as follows:

1) Similar to previous case, one out of eight impressions for each fingerprint is used as the enrolled templates with five distinct permutation orders.

2) Each of the remaining seven impressions is assigned another ten different keys to produce ten unique permuted templates and matched against the five enrolled templates.

3) Produce the distribution (hereafter denoted as $D_2$) of matchable minutiae over all fingerprints for this scenario. Then, calculate the separability between $D_2$ and $OG$, defined as:

$$\text{Separability}_{D_2} = \frac{|\mu_{D_2} - \mu_{OG}|}{\sqrt{\sigma^2_{D_2} + \sigma^2_{OG}}},$$

(15)

where $\mu_{D_2}$ and $\sigma^2_{D_2}$ are the mean and variance of $D_2$ respectively.

Figure 4 also includes the distribution of $D_2$ for $mlcn$-$Dice$, $mlcmb1$-$Dice$ and $mlcmb2$-$NWAZZOO$. Similar to those in case 1 (Section 6.3.1), the distributions are uncorrelated or hardly correlated with $OG$. Hence, if the length of a real number MLC and binary MLC are 1,476 and 1,476B respectively (correspond to the parameters in Section 3.1), almost all of the 1,476! or 1,476B! permutation orders can be utilized. This applies to all the datasets used.

Revocability increases when the separability increases. Also, the revocability of cancellable fingerprint template is one major factor which determines its diversity. From Table 2, we can observe that for all datasets, $mlcn$-$Dice$ yields the highest diversity while the enhanced MLC weakens in this property.

6.4. Non-reversibility

In this section, we discuss the non-reversibility of the proposed system by evaluating its security against brute-force attack. It is mainly determined by two factors - the strength of the non-invertible transform and the amount of possible permutation keys. As discussed in Section
Table 2: Separabilities of all MLC settings in the format of Separability$_D^1$, Separability$_D^2$ ($\mu_{D^1}, \sigma^2_{D^1}$, $\mu_{D^2}, \sigma^2_{D^2}$).

<table>
<thead>
<tr>
<th>Method/Setting</th>
<th>FVC2002DB1</th>
<th>FVC2002DB2</th>
<th>FVC2004DB1</th>
<th>FVC2004DB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>mlcn-Dice</td>
<td>4.38, 4.38</td>
<td>4.4, 4.4</td>
<td>4.55, 4.55</td>
<td>3.3, 3.3</td>
</tr>
<tr>
<td></td>
<td>[27.38, 78.14]</td>
<td>[33.59, 116.52]</td>
<td>[25.46, 62.58]</td>
<td>[18.76, 64.46]</td>
</tr>
<tr>
<td></td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
</tr>
<tr>
<td></td>
<td>[25.48, 72.75]</td>
<td>[30.67, 124.7]</td>
<td>[22.43, 67.75]</td>
<td>[18.78, 68.02]</td>
</tr>
<tr>
<td></td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
</tr>
<tr>
<td>mlcm1-Dice</td>
<td>3.56, 3.56</td>
<td>3.43, 3.43</td>
<td>3.08, 3.08</td>
<td>2.8, 2.8</td>
</tr>
<tr>
<td></td>
<td>[23.62, 88.05]</td>
<td>[27.74, 130.72]</td>
<td>[18.86, 74.81]</td>
<td>[16.61, 70.55]</td>
</tr>
<tr>
<td></td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
</tr>
<tr>
<td>mlcm2aq-Dice</td>
<td>3.65, 3.65</td>
<td>3.59, 3.59</td>
<td>3.13, 3.13</td>
<td>2.99, 2.99</td>
</tr>
<tr>
<td></td>
<td>[23.93, 85.78]</td>
<td>[28.66, 127.14]</td>
<td>[19.06, 74.25]</td>
<td>[16.44, 60.29]</td>
</tr>
<tr>
<td></td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
</tr>
<tr>
<td>mlcm2aq-DWID</td>
<td>3.75, 3.75</td>
<td>3.72, 3.72</td>
<td>3.39, 3.39</td>
<td>3.05, 3.05</td>
</tr>
<tr>
<td></td>
<td>[24.72, 86.91]</td>
<td>[29.67, 127.54]</td>
<td>[20.54, 73.23]</td>
<td>[21.32, 97.69]</td>
</tr>
<tr>
<td></td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
</tr>
<tr>
<td>mlcm2aq-NWAZOO</td>
<td>2.83, 2.83</td>
<td>3.86, 3.86</td>
<td>3.43, 3.43</td>
<td>2.82, 2.82</td>
</tr>
<tr>
<td></td>
<td>[19.19, 11.11]</td>
<td>[30.31, 122.96]</td>
<td>[20.21, 69.33]</td>
<td>[19.39, 94.94]</td>
</tr>
<tr>
<td></td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
</tr>
<tr>
<td>mlcm2ug-DWID</td>
<td>3.61, 3.61</td>
<td>3.93, 3.93</td>
<td>3.62, 3.62</td>
<td>3.02, 3.02</td>
</tr>
<tr>
<td></td>
<td>[23.81, 86.91]</td>
<td>[30.31, 120.84]</td>
<td>[21.12, 68.19]</td>
<td>[17.82, 69.62]</td>
</tr>
<tr>
<td></td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
</tr>
<tr>
<td>mlcm2ug-DWID</td>
<td>3.72, 3.72</td>
<td>4.4, [31.11, 121.1]</td>
<td>3.67, 3.67</td>
<td>3.04, 3.04</td>
</tr>
<tr>
<td></td>
<td>[24.62, 87.65]</td>
<td>(0, 0), (0.0)</td>
<td>[19.39, 94.9]</td>
<td>[20.03, 87.09]</td>
</tr>
<tr>
<td></td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
<td>(0, 0), (0, 0)</td>
</tr>
</tbody>
</table>

Figure 4: Matchable MLCs distributions for FVC2002DB1. For the three settings shown in the figure, $D^1$ and $D^2$ has all-zero distribution, thus resulting in zero means ($\mu_D$) and zero variances ($\sigma^2_D$) as shown in Table 2. Among the three, OG of mlcn-Dice is distributed furthest from its $D^1$ and $D^2$, resulting in highest separability.

6.3, the attackers require at most 1,476! (or 1,476B!) attempts to reconstruct the original template from the permuted template, which is certainly computational infeasible in real time. Despite that, under the worst case scenario where both the template and the key are compromised and the transformation method is known, the security only relies on the strength of the transformation. It...
refers to the improbability of unveiling the original minutiae set from the unpermuted template.

For the number-of-minutiae version of MLC, provided $\Delta \varphi = \frac{\pi}{3} \approx 1.1$ (from Section 3.1), with precision of 0.1, there are 11 possible orientations in an angular division. Also given $r = 25$ and $\frac{\pi r^2}{2} \approx 982$, we can know that there are approximately 982 pixels in a region (semi-circle). Therefore, a 1 in the MLC may be the result of a minutia being in one of the $11 \times 982 = 10,802$ locations in the semi-cylinder. Even if there are only 40 unique minutiae in the fingerprint, the total number of repeatable minutiae contributing to all the entire template can easily exceed 3,000. Since there is no way to know which ones are repeating, the attacker requires more than $10,802 \times 3,000 \approx 30$ million attempts to uncover the raw minutiae data.

It does not make it easier when the mean distance value is used as MLC. Due to the fact that the calculation of mean value itself is a many-to-one mapping function, the attacker needs to get all the distance values in a region before cracking the location of the minutiae, without any information about the number of minutiae within the region. This creates even more hassles for the attacker. Furthermore, the quantization process in binary MLC generation serves as an additional level of protection as it is another many-to-one mapping function to be cracked. In short, it is impracticable to reverse the transformation in real time.

6.5. Template Size

Given that the length of a MLC is $1,476 \times k$, where $k = 1$ for real number MLC and $k$ is equivalent to the quantization bit-length ($B$) for binary MLC. Taking an integer as a two-byte data and a decimal-point number as a four-byte data, the template size for mlcn and mlcm are approximately 115 KB and 230 KB respectively (assuming that there are 40 minutiae in a fingerprint).

With the implementation of MLC binarization, the original MLC is converted into a 1,476-$B$-bit binary MLC plus a 1,476-bit invalid codes. So when $B = 1$, the template size drops down to 14 KB whereas it is 22 KB for $B = 2$. In general, the template compression rate from mean-distance-value MLC to binary MLC is $32 : (B+1)$. Besides, since MLC contains long repeated 0’s, the storage size can be further reduced using appropriate data compression algorithm.
7. Conclusions

In this paper, a two-staged enhancement method was proposed to ameliorate the performance of MLC considering the recognition accuracy, template size and security. The idea was to acquire the balance point among the four criterions. The results showed that the enhanced MLC excelled in recognition accuracy and template size, but the computational cost and security were compromised to a reasonable extent that the three main objectives of template protection scheme (Section 1) were not violated.

One potential application of MLC (or specifically binary MLC) is biometric cryptography (bio-cryptography), which includes key generation and key binding. However, MLC alone is not robust enough to be used in bio-crypto systems. Error correction coding, fuzzy vault and fuzzy commitment are the possible ways of incorporating MLC in bio-cryptography.

References


