Improving Gait Cryptosystem Security Using Gray Code Quantization and Linear Discriminant Analysis

Lam Tran[1,3], Thang Hoang[2], Thuc Nguyen[3], and Deokjai Choi[1]

[1] ECE, Chonnam National University, Gwangju, South Korea
dchoi@jnu.ac.kr
[2] EECS, Oregon State University, Corvallis, Oregon, USA
hoangmin@oregonstate.edu
[3] FIT, Ho Chi Minh University Of Science, Ho Chi Minh, Vietnam
{thlam,ndthuc}@fit.hcmus.edu.vn

Abstract. Gait has been considered as an efficient biometric trait for user authentication. Although there are some studies that address the task of securing gait templates/models in gait-based authentication systems, they do not take into account the low discriminability and high variation of gait data which significantly affects the security and practicality of the proposed systems. In this paper, we focus on addressing the aforementioned deficiencies in inertial-sensor based gait cryptosystem. Specifically, we leverage Linear Discrimination Analysis to enhance the discrimination of gait templates, and Gray code quantization to extract high discriminative and stable binary template. The experimental results on 38 different users showed that our proposed method significantly improve the performance and security of the gait cryptosystem. In particular, we achieved the False Acceptant Rate of $6 \times 10^{-5}\%$ (i.e., 1 fail in 16983 trials) and False Rejection Rate of 9.2% with 148-bit security.

Keywords: gait authentication, biometric cryptosystem, biometric template protection, fuzzy commitment scheme

1 Introduction

Gait has been considered as an efficient modality for recognizing individual via human motion [2]. The growth of microelectromechanical technology has opened a new approach for implementing gait authentication systems (e.g., [3,7,8,11,12,13,25,31,33,34,36]), in which the gait signals are collected by inertial-sensors. This technique permits implicit user authentication and therefore, offers significant usability advantages compared with password or other biometric systems [13] which require the user to perform explicit gesture to be authenticated. Several inertial-sensors based gait authentication schemes have been proposed in the literature (e.g., [3,7,13,25,31,33]). Despite their merits, all these studies rely on traditional pattern recognition approaches, where the extracted gait templates or models are stored locally without confidentiality protection, which might pose security and privacy issues to the user once such raw data are compromised by the attacker (e.g., via malware) [19].

To address the privacy concern of biometric data, several studies leveraging Biometric Cryptosystem (BCS) [28] have been proposed [11,14,15,21,23,27]. One of the most common techniques that has been recently used to protect biometrics templates is Fuzzy Commitment Scheme (FCS) [18], where a binary string is extracted from the biometric templates and then, binded with a cryptographic key encoded by Error Correcting Code (ECC) [22] before being written to the storage (e.g., [11,14,15,27]). Despite the fact that such schemes offer an elegant strategy to protect the privacy of biometric templates, they did not take into account the characteristic of behavioral biometric modalities such as gait, which is well-known to be low discriminative and highly unstable. As described in [20], these issues can significantly degrade the security and performance of the FCS-based system (e.g., key length, False Acceptant Rate (FAR), False Rejection Rate (FRR)), where a low discriminative extracted binary string might result in a high FAR while an unstable one can lead to high FRR and low security. Thus, it is vital to develop a method that can extract high discriminative and stable strings from the gait templates to improve the security and performance of gait cryptosystem.

In this paper, we propose methods to address the aforementioned deficiencies to improve the security and performance of inertial-sensor based gait cryptosystem as follows:

- First, we handle the problem of low discriminability and high variation of gait data by adopting Linear Discriminant Analysis (LDA) [32]. As the traditional LDA is incompatible with FCS (see Section 3.3), we propose a modification of LDA to *(i)* improve the discriminability of gait data from different users, *(ii)* reduce the variation of gait data from the same user, *(iii)* maintain the high feature dimension of gait data to extract a long enough binary string to be used in FCS (Section 3.3).
- Second, we propose Gray code [9] quantization scheme, which can offer strong capability of error toleration, to quantize the gait templates after LDA projection to binary template (Section 3.4).
- Third, we design a method that can determine the reliability of each components in the extracted binary template (Section 3.5). Highly reliable components will be selected to form the final binary string input for FCS.
- Last, we conduct a comprehensive experiment to analyze the efficiency of the proposed techniques and perform security analysis in details to evaluate the security of our system against different attacks. We achieved $6 \times 10^{-5}\%$ FAR (i.e., 1 fail in 16983 trials), 9.2% FRR at 148-bit security. This experimental result indicated that the proposed methods significantly improve not only the security but also the performance of the gait cryptosystem compared with other state-of-the-art works (Section 4).

2 Preliminaries

2.1 Notations

Given a matrix $\mathbf{M}$, $\mathbf{M}[i,j]$ denotes accessing the cell indexing at row i and column j. $|\mathbf{M}|$ denotes the determinant of matrix $\mathbf{M}$. Given two matrices $\mathbf{A}$ and $\mathbf{B}$ having the same number of rows, $\mathbf{C} = [\mathbf{A}\ \mathbf{B}]$ denotes that matrix $\mathbf{C}$ is formed by concatenating $\mathbf{A}$ and $\mathbf{B}$ horizontally. $\mathbf{C} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}$ means $\mathbf{C}$ is formed by vertically concatenating two matrices $\mathbf{A}$ and $\mathbf{B}$ having the same number of columns. Given an $m \times n$ matrix $\mathbf{M}$, we denote the mean vector of $\mathbf{M}$ as $\overline{\mathbf{m}} = (\overline{m}_1, \ldots, \overline{m}_j, \ldots, \overline{m}_n)$ where $\overline{m}_j = \frac{1}{m}\sum_{i=1}^{m}\mathbf{M}[i,j]$. Given an n-dimensional vector $\mathbf{x} = (x_1, \ldots, x_j, \ldots, x_n)$, we denote the mean of $\mathbf{x}$ as $\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j$. $\lceil\cdot\rceil$ is the ceiling operator. $|x|$ means the absolute value of variable x. We denote $\oplus$ as the bitwise XOR operator and $||$ as binary string concatenation operator. $\alpha \gg t$ means logical right shifting α by t bits. $H : \{0,1\}^* \rightarrow \{0,1\}^n$ is a secure cryptographic hash function, where n is the length of hash value.

2.2 The Fuzzy Commitment scheme

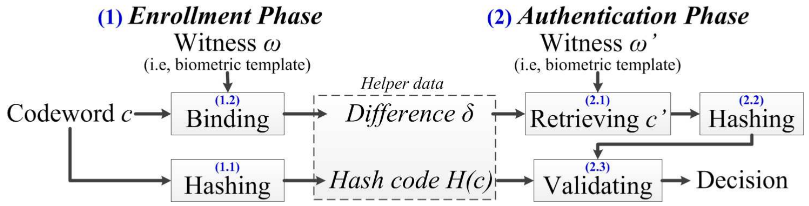


Fig. 1: The Fuzzy Commitment Scheme [18].

Fuzzy Commitment Scheme (FCS) is a generic BCS framework proposed by Juels and Wattenberg [18], which leverages Error Correcting Code (ECC) [22] to handle the variation of biometric data. The key idea of FCS is to express an n-bit witness ω (i.e., biometric template) in term of a codeword $c \in \mathcal{C}$ of length n and an offset $\delta \in \{0,1\}^n$ such that $\omega = c \oplus \delta$ where $\mathcal{C}$ is an error correcting codebook. FCS operates in two phases as sketched in Figure 1.

1. **Enrollment phase**: A codeword $c \in \mathcal{C}$ is selected randomly and its hash value $H(c)$ is calculated (step 1.1). Meanwhile, c is sealed to δ by the biometric template ω (step 1.2). The hash value $H(c)$ and δ are stored as helper data for authentication while c and ω are discarded.
2. **Authentication phase**: Given a biometric template ω', the estimated codeword c' is retrieved using the helper data δ (step 2.1). Then, its hash value $H(c')$ is calculated (step 2.2). Finally, $H(c')$ is matched with $H(c)$ to give the final verification decision (step 2.3).

As discussed in [18], each codeword c in $\mathcal{C}$ has two parts as information part of length k ($k < n$) and redundancy part of length $(n - k)$. The ratio between the amount of two parts in c is a trade-off between security strength and the resilience. The system is more secure when the information part is extended. In contrast, the system provides higher capability of resilience when the redundancy part is lengthened.

2.3 Fisher's Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a data dimensional reduction technique that reserves as much as possible the discrimination information between different classes. Assuming that a training dataset $\mathbf{X}$ includes D classes L_i, each having N_i templates. LDA finds $\mathbf{W}$ to transform $\mathbf{X}$ to $\mathbf{Y}$ as $\mathbf{Y} = \mathbf{W}^T\mathbf{X}$ so that the intra-class variation is minimized and inter-class discrimination is maximized in $\mathbf{Y}$.

Let $\bar{\mathbf{x}}$ be the mean vector of $\mathbf{X}$ and $\bar{\mathbf{x}}_i$ be the mean vector of templates of class L_i. The within-class scatter matrix $\mathbf{S}_w$ and between-class scatter matrix $\mathbf{S}_b$ are calculated by:

$$\mathbf{S}_w = \sum_{i=1}^{D}\sum_{j=1}^{N_i}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^\top, \tag{1}$$

$$\mathbf{S}_b = \sum_{i=1}^{D} N_i(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^\top, \tag{2}$$

where $\mathbf{x}_{ij}$ is the template j of class L_i. The projection matrix $\mathbf{W}$ is the result of the maximization problem using the Fisher's criterion [6] as:

$$J(\mathbf{W}) = \frac{|\mathbf{W}^\top\mathbf{S}_b\mathbf{W}|}{|\mathbf{W}^\top\mathbf{S}_w\mathbf{W}|}. \tag{3}$$

The optimization task of (3) is equivalent to the following generalized eigenvalue problem described in [32] as: $\mathbf{S}_b\mathbf{w}_i = \lambda_i\mathbf{S}_w\mathbf{w}_i$, where $\mathbf{w}_i$ and λ_i ($1 \leq i \leq D - 1$) respectively are the eigenvector and eigenvalue of $\mathbf{S}_w^{-1}\mathbf{S}_b$. When $\mathbf{S}_w$ is nonsingular, the optimal $\mathbf{W}$ is the one whose columns are the eigenvectors corresponding to at most $(D - 1)$ largest eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$.

3 The Proposed Gait Cryptosystem

In this section, we first present the general architecture of our proposed inertial-sensor-based gait authentication cryptosystem. We introduce overall steps of data (pre)processing to extract gait templates collected from the inertial sensor data. Finally, we present the main techniques which adopt LDA and Gray code quantization along with a reliability extraction method to enhance the security and performance of the gait cryptosystem.

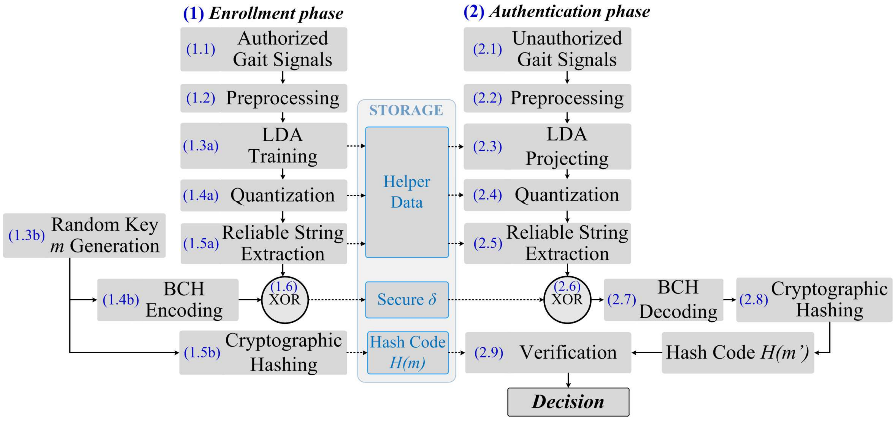


Fig. 2: The architecture of the inertial-sensor based gait cryptosystem.

3.1 Overall of System Architecture

We present in Figure 2 the specification of our proposed inertial-sensor based gait authentication system which follows the Fuzzy Commitment Scheme as follows.

1. **Enrollment**: First, we collect the training gait data (step 1.1) from inertial sensor, and perform data (pre)processing to extract gait templates (step 1.2). We then apply LDA training (step 1.3a) to the extracted gait templates, followed by a Gray code-based quantization (step 1.4a) and reliable string extraction (step 1.5a) to obtain a discriminative and stable binary string. Concurrently, we generate a key m randomly (step 1.3b) and then encode it into a BCH codeword (step 1.4b). Meanwhile, we calculate the hash value of m (denoted as $H(m)$) (step 1.5b). Finally, we bind the binary string with the codeword to get the secure δ (step 1.6). We store the hash value $H(m)$ and δ along with some auxiliary data in steps 1.3a–1.5a as helper data for using in authentication phase.
2. **Authentication**: Given gait data to be verified, we extract a stable binary string by using the stored helper data (steps 2.1–2.5). We retrieve an estimated BCH codeword by binding the new extracted binary string with the stored secure δ (step 2.6). We then decode the estimated codeword to get the secret key m' (step 2.7), and calculate its hash code $H(m')$ (step 2.8). Finally, we match $H(m')$ with $H(m)$ to verify the authenticating user (step 2.9).

Notice that our general framework is inspired and extended from [11]. In this paper, we mainly focus on improving the security and performance of the gait cryptosystem, wherein we introduce two additional steps including LDA and Gray code quantization to enhance the discriminability of gait data. Hence, we present in following sections how to implement such vital steps in details, and refer the readers to [11] for detailed presentations of other (pre)processing steps.

3.2 Data Preprocessing and Feature Extraction

We leverage the methods proposed in [12] for gait data preprocessing. Specifically, we address the disorientation problem using the data additionally collected from orientation sensor, and mitigate the noise in gait signals by adopting the Daubechies orthogonal wavelet with level 2. We represent each sampling of gait signals as $\mathbf{a} = (a_X, a_Y, a_Z)$, where a_X, a_Y, a_Z are acceleration values captured in X, Y, Z dimensions, respectively. Subsequently, we divide the gait data into consecutive of gait-cycle-based segments where each gait cycle is defined as a time period between two times of ground contacting of a same foot while walking. Hence, each gait cycle $\mathbf{C}_i$ contains t acceleration samples as $\mathbf{C}_i = [\mathbf{a}_{i1} \ldots \mathbf{a}_{ij} \ldots \mathbf{a}_{it}]$. We then form the gait pattern by concatenating 4 consecutive gait cycles in a way that two consecutive gait patterns overlap with each other by 2 gait cycles as $\mathbf{P}_i = [\mathbf{C}_{2i-1} ... \mathbf{C}_{2(i+1)}]$. For each $\mathbf{P}_i$, we extract features in both time and frequency domain as described in [12] to form a gait template $\mathbf{x}_i = (x_{i1}, \ldots, x_{ij}, \ldots, x_{iM}) \in \mathbb{R}^M$, where x_{ij} denotes the feature j extracted from pattern $\mathbf{P}_i$, and M is the total number of features being extracted.

3.3 Improving the Discriminability of Gait Data

We observe that gait is more noisy and less discriminative than other biometric traits. Hence, instead of directly using the gait templates for further processing, we adopt LDA to enhance the inter-class discriminability and reduce the intra-class variation.

LDA training: In the enrollment phase, we form a data matrix $\mathbf{G}$ including N gait templates $\mathbf{x}_i$ of the genuine user as $\mathbf{G} = \left[\mathbf{x}_1 \ldots \mathbf{x}_i \ldots \mathbf{x}_N\right]^\top \in \mathbb{R}^{N \times M}$, and the data matrix $\mathbf{I}$ including N' gait templates $\mathbf{x}'_i$ of all other users $\mathbf{I} = \left[\mathbf{x}'_1 \ldots \mathbf{x}'_i \ldots \mathbf{x}'_{N'}\right]^\top \in \mathbb{R}^{N' \times M}$. We form the dataset $\mathbf{M} = \begin{bmatrix}\mathbf{G}\\ \mathbf{I}\end{bmatrix} \in \mathbb{R}^{(N+N') \times M}$. We label the templates in $\mathbf{M}$ with two classes including genuine and impostor. We use $\mathbf{M}$ as the data for LDA training to find the projection matrix for transforming gait templates.

However, the traditional LDA has a dimensional limitation as described in [32] which makes it incompatible to the gait cryptosystem. Specifically, with D as the number of classes, there are $(D-1)$ eigenvectors $\mathbf{w}_i$ of $\mathbf{S}_w^{-1}\mathbf{S}_b$ that have the corresponding eigenvalues λ_i satisfying $\lambda_i > 0$, where $\mathbf{S}_w$ and $\mathbf{S}_b$ are calculated by (1) and (2), respectively. Then, LDA will form a projection matrix $\mathbf{W}$ by using at most $(D-1)$ eigenvectors. Thus, the data dimension after LDA projection will be at most $(D-1)$. In current system, with $D = 2$, the dimension of data after LDA projection is 1 which is insufficient for extracting to a reliable string because the it is required to have the same length with BCH codeword for binding (Figure 2, step 1.6). Therefore, we modify the process of LDA as follows.

First, instead of using $\mathbf{M}$ for LDA training, we separate $\mathbf{M}$ into S submatrices $\mathbf{M}_i$ $(1 \leq i \leq S)$, each having K columns, and apply LDA to each $\mathbf{M}_i$ independently to get a projection matrix $\mathbf{W}_i$. Specifically, we calculate the within-class scatter matrix $\mathbf{S}_w^{(i)}$ and between-class scatter matrix $\mathbf{S}_b^{(i)}$ of each dataset $\mathbf{M}_i$. Then, we factorize the matrix $(\mathbf{S}_w^{(i)})^{-1}\mathbf{S}_b^{(i)}$ to a set of K eigenvectors $\mathbf{w}_l^{(i)}$ and corresponding eigenvalues $\lambda_l^{(i)}$ $(1 \leq l \leq K)$. Second, instead of using at most $(D-1)$ eigenvectors $\mathbf{w}_l^{(i)}$ corresponding to $(D-1)$ largest eigenvalues $\lambda_l^{(i)}$ to form $\mathbf{W}_i$ as described in Section 2.3, we use all eigenvectors $\mathbf{w}_l^{(i)}$ as

$$\mathbf{W}_i = [\mathbf{w}_1^{(i)} \ldots \mathbf{w}_l^{(i)} \ldots \mathbf{w}_K^{(i)}]. \tag{4}$$

$\mathbf{W}_i$ is used to transform gait data in sub-space i in both enrollment and authentication phase. We store all projection matrices $\mathbf{W}_i$ as helper data.

LDA projection: Given S projection matrices $\mathbf{W}_i$, we determine the LDA projection $\mathbf{G}'$ of $\mathbf{G}$ by *(i)* determining $\mathbf{G}_i \in \mathbb{R}^{N \times K}$ for each sub-space $1 \leq i \leq S$; *(ii)* calculating the projection of $\mathbf{G}_i$ as $\mathbf{G}'_i = \mathbf{W}_i^\top \mathbf{G}_i$ for each $\mathbf{G}_i$; *(iii)* forming $\mathbf{G}'$ as:

$$\mathbf{G}' = [\mathbf{G}'_1 \ldots \mathbf{G}'_i \ldots \mathbf{G}'_S] \in \mathbb{R}^{N \times M}. \tag{5}$$

We also transform matrix $\mathbf{I}$ to $\mathbf{I}'$ using $\mathbf{W}_i$ similar to transforming $\mathbf{G}$ as above. Then, we use $\mathbf{G}'$ and $\mathbf{I}'$ for quantization and reliable binay string extraction as will be described in the following sections.

3.4 Gray Code Quantization

In order to reduce the natural variation of gait data, we use N templates in matrix $\mathbf{G}'$ (5) for quantization to construct a binary gait template. We determine $\bar{\mathbf{g}}' \in \mathbb{R}^M$ as the mean vector of matrix $\mathbf{G}'$ and use $\bar{\mathbf{g}}'$ to generate a binary gait template as follows.

First, we normalize each component $\overline{g}'_j$ in $\bar{\mathbf{g}}'$ such that $\overline{g}'_j \in [0,1]$, for $1 \leq j \leq M$. Note that all the min, max values (represented as min, max vectors) extracted in the normalization process will be stored as the helper data. Let $\varPsi$ be a system parameter that specifies the number of bits representing one real-valued component in quantization. Then, we divide the range value [0,1] to $2^{\varPsi}$ continuous subranges which are called as quanta. Hence, the range of each quantum is $\phi = \frac{1}{2^{\varPsi}}$. Consequently, we map each quantum to a unique $\varPsi$-bit string. The normalized value of $\overline{g}'_j$ may variate in two continuous quanta at different times of sampling. So the mapping between the set of quanta and set of $\varPsi$-bit strings should be well-arranged so that any two binary strings corresponding to two continuous quanta differ to each other in one bit. Gray code [9] is a good candidate for this requirement as it is a technique for designing a binary numeral system in which two successive strings have only one different bit. Given a normalized value of $\overline{g}'_j$, the quantum index i_j is defined such that $i_j\phi < \overline{g}'_j \leq (i_j+1)\phi$. Then we calculate the corresponding $\varPsi$-bit string ω_j following Gray code system as [5]:

$$\omega_j = B(\varPsi, i_j) \oplus (B(\varPsi, i_j) \gg 1), \tag{6}$$

where $B(\varPsi, i_j)$ is the representation of i_j in $\varPsi$-bit string.

Finally, from all ω_j, we form the binary gait template $\boldsymbol{\omega}$ which is the quantized template of $\bar{\mathbf{g}}'$ as:

$$\boldsymbol{\omega} = (\omega_1, \ldots, \omega_j, \ldots, \omega_M). \tag{7}$$

3.5 Reliable Binary String Extraction

In this section, we propose a method to extract highly reliable components in the binary gait template. By reliability, we mean the one having low variation in enrolled users' templates and high discriminability between templates of enrolled user and other users.

Given a binary gait template $\boldsymbol{\omega}$, we select R reliable components ω_j to form the reliable string $\omega \in \{0,1\}^n$ which will be used to bind with codeword c. The value of R is determined based on two other predefined parameters including the codeword length n and the number of Gray code quantization bits Ψ as $R = \lceil \frac{n}{\Psi} \rceil$.

We use $\mathbf{I}'$ and $\mathbf{G}'$ in (5) for estimating the reliability of each component. We propose a formula to calculate the reliability φ_j of each component ω_j of as:

$$\varphi_j = \frac{1}{2}\left(1 + \mathsf{erf}\left(\frac{\frac{1}{N'}\sum_{i=1}^{N'}\left|\mathbf{I}'[i,j] - \bar{g}'_j\right|}{\sqrt{2\sigma_j^2}}\right)\right), \tag{8}$$

where erf denotes the Gaussian Error Function [1], $\bar{g}'_j$ is the component j of mean vector $\bar{\mathbf{g}}'$ in Section 3.4, and the variance σ_j^2 of component j is calculated as:

$$\sigma_j^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(\mathbf{G}'[i,j] - \bar{g}'_j\right)^2. \tag{9}$$

In (8), the numerator of expression inside the erf function measures the discriminability of component j between enrolled user and other users. The denominator measures the variation of component j of enrolled user. Let $\mathbf{p} = (p_1, \ldots, p_j, \ldots, p_M) \in \mathbb{N}^M$ be the vector containing the index of components that follows the descending order of reliability, $\varphi_{p_j} \geq \varphi_{p_{j+1}}$. We use first R components of $\mathbf{p}$ to extract the reliable components in $\boldsymbol{\omega}$ to form the final reliable string ω as:

$$\omega = \omega_{p_1} || \omega_{p_2} || \ldots || \omega_{p_R}. \tag{10}$$

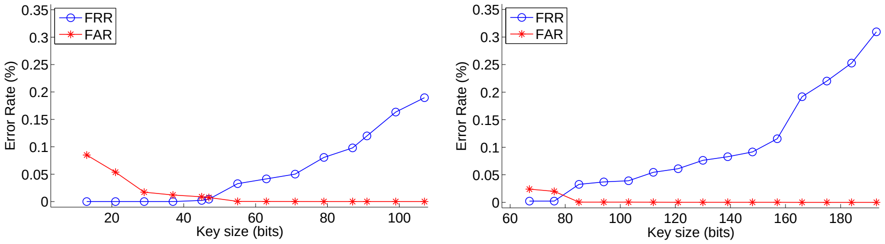
Note that we store the first R components in $\mathbf{p}$ as helper data to extract reliable components in the authentication phase.

4 Experiments

4.1 Configurations and Results

We used the dataset in [12] for the experimental analysis of the proposed system. The dataset contains gait signals of 38 users. We extracted the gait signals to

10224 gait templates using the process in Section 3.2. For the empirical analysis, we built an authentication models for each user. In each model, we considered one user as the genuine and the others are impostors. In the enrollment phase for each user, we formed the matrix $\mathbf{G}$ containing $N = 100$ gait templates of the genuine user and $\mathbf{I}$ containing $N' = 100 \cdot 37 = 3700$ templates of the impostors; and the remaining data is used in authentication phase to verify the built model (12 templates for each time of attempting). In the LDA training step, we divided the original data space into $S = 15$ sub-spaces as explained in Section 3.3. We selected BCH codeword lengths of 255 and 511 bits. We analyzed the system with different values of quantization bit Ψ and key length k to understand the impact of such parameters. We used False Acceptant Error Rate (FAR) and False Rejection Error Rate (FRR) as the standard metrics to evaluate the performance of our proposed system. Finally, we analyzed the security of our system against various attacks.



(a) 255-bit BCH codeword

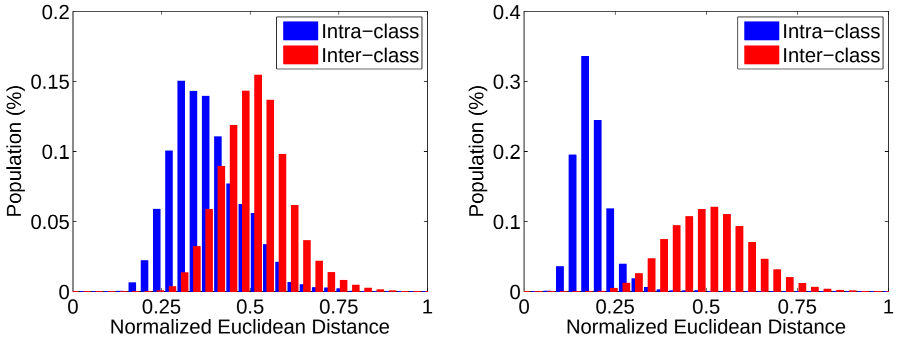
(b) 511-bit BCH codeword

Fig. 3: The FRR and FAR at different key length of codeword 255, 511 and 4-bit Gray code quantization.

With 4-bit Gray code quantization, we have the optimal result. Figure 3 displays the FAR and FRR with different key lengths and BCH codewords. At 255-bit codeword and 87-bit key, the system achieves 0% FAR and 9.8% FRR. With the 511-bit codeword and 148-bit key, the FRR is 9.2% and FAR is 6×10^{-5}% (i.e., 1 fail in 16983 trials). Under different attacks, the security of the system is 87 and 148 bits according to 255-bit and 511-bit codeword, respectively (analyzed in Section 4.4).

4.2 The Impact of LDA Projection

We used Normalized Euclidean distance [35] to analyze the impact of LDA projection on the discriminability of gait template. Figure 4a displays the Normalized Euclidean distance distribution of gait template before LDA projecting. We can see that the overlapping area of the intra-class and inter-class is substantial which reflects the naturally low discrimination of gait data. After applying the modified LDA, the overlapping area reduces significantly as shown in Figure



(a) Before LDA projection

(b) After LDA projection

Fig. 4: The normalized Euclidean distance distribution of gait templates before and after LDA projection.

4b. This contrast illustrates the effectiveness of the modified LDA presented in Section 3.3. The LDA projection step plays an important role since it enhances the data discriminability, and therefore, significantly improves the system performance.

4.3 The Impact of Gray Code Quantization

We used the Normalized Hamming distance [35] to analyze the impact of Gray code quantization. As gait signals are unstable, a specific component of gait template can have different values at each time of sampling. However, if these values still belong to the same quantum, the system will result in the same binary string. The use of Gray code quantization can minimize the error bits when these values fall into different quanta. So, adopting Gray code provides higher capability of error tolerance to enhance the performance.

The number of quantization bits Ψ is a trade-off between the FAR and FRR values of the system. The quantum range ϕ decreases as Ψ increases and vice versa. Given that Ψ is small, (thus ϕ is large), it is likely that the same binary string can be extracted from two different gait templates. As a result, the inter-class and intra-class Hamming distance are decreased as illustrated in Figure 5. This results in the increase of the FAR, and the decrease of the FRR. Figure 6 displays the comparison of Hamming distance distribution between the cases of using 4-bit natural binary code and 4-bit Gray Code quantization. When using Gray code (Figures 6 c, d), the intra-class Hamming distance is much smaller compared with using natural binary code (Figures 6 a, b). Table 1 gives a comparison of 3-bit and 4-bit Gray code quantizations in terms of FRR, FAR at the same codeword length and key length. We can see that when $\Psi = 3$, the FRR is lower while FAR is higher than that of $\Psi = 4$, respectively.

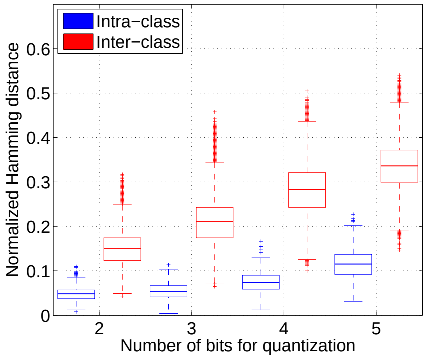


Fig. 5: The Hamming distance distribution when using different values of Ψ for Gray code quantization.

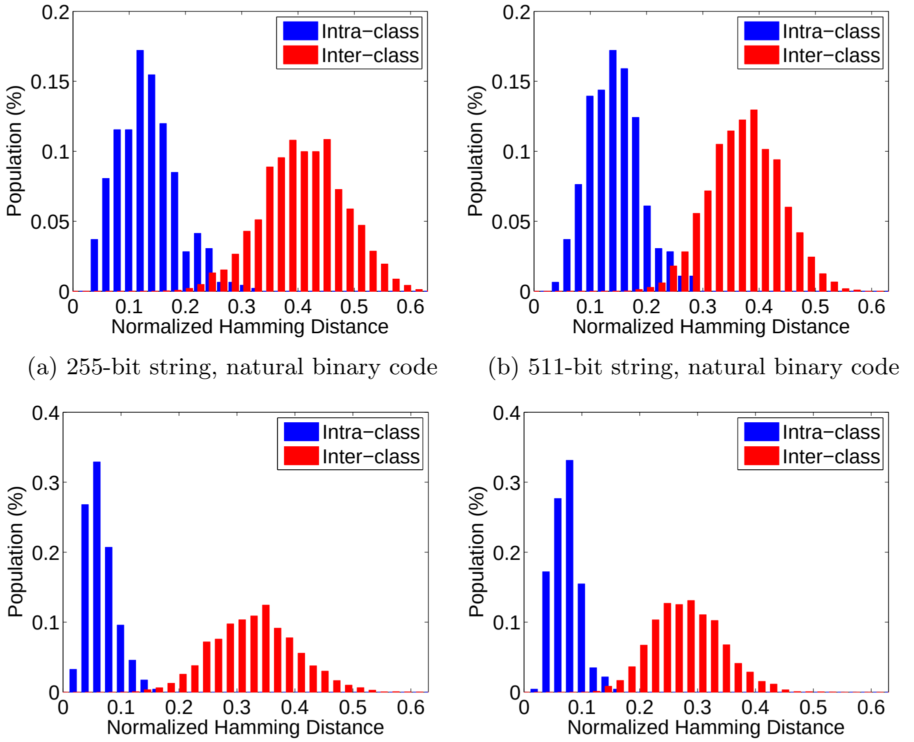
Table 1: The system performance pertaining to codeword length n, key length k and number of quantization bits Ψ

$\Psi = 3$				$\Psi = 4$			
n (bits)	k (bits)	FAR (%)	FRR (%)	n (bits)	k (bits)	FAR (%)	FRR (%)
255	79	0.4	1.9	255	79	6×10^{-5}	8.1
	87	0.3	2.3		87	0	9.8
	91	0.3	2.8		91	0	12.0
511	139	0.65	0.93	511	139	11×10^{-5}	8.3
	148	0.56	1.17		148	6×10^{-5}	9.2
	157	0.38	1.4		157	6×10^{-5}	11.5

4.4 Security Analysis

In this section, we analyze the system security against several statistical attacks. The typical attack is brute force the random key. As the proposed key lengths are 87 and 148 bits for 255-bit and 511-bit codewords, the security strength against key brute force attack are 87 and 148 bits, respectively.

We analyze whether an attacker can exploit information from the helper data including projection matrices $\mathbf{W}_i$, the min, max vectors for normalization, the reliable components index $\mathbf{p}$, secured δ and hash code $H(m)$. The min, max vectors contain statistical information from the whole dataset, and therefore, is not user-specific. Thus, the min, max vectors do not reveal information about genuine user. The reliable component index vector $\mathbf{p}$ only contains the information about the discriminability and stability of gait templates. Such indexes does not reveal information about biometric template, thus it cannot be used to revert to biometric template. With the hash code $H(m)$, the attacker cannot revert to m with a non-negligible probability, given that the cryptographic hash function H is secure.



(a) 255-bit string, natural binary code

(b) 511-bit string, natural binary code

(c) 255-bit string, Gray code

(d) 511-bit string, Gray code

Fig. 6: The Hamming distance distribution of 255-, 511-bit reliable strings when using natural binary code and Gray code with 4-bit quantization.

LDA projection matrices are not user-specific since they only reflect the information about the dataset population. Additionally, the projection matrix is formed by eigenvectors λ of $S_w^{-1}S_b$. From λ, we cannot revert to $S_w^{-1}S_b$ without knowing the corresponding eigenvalues, which are immediately discarded after the LDA training phase. Thus, from the stored eigenvectors, we cannot revert to $S_w^{-1}S_b$ and obtain original biometric templates of the enrolled user.

Using the secure δ, in order to get the key m, the attacker can guess a string ω' that is close enough to ω hidden in δ. The distance strictly depends on the error correcting capability of BCH code and the uncertainty of ω which depends on the quantization method. We use entropy to measure the uncertainty of ω. We calculate the entropy of each bit in ω by the formula in [29] as:

$$H(\omega_i) = -p_i \log_2(p_i) + (1 - p_i)\log_2(1 - p_i), \tag{11}$$

where $p_i = \Pr(\omega_i = 1)$ is the probability of bit i getting value 1 due to quantization. The entropy E of reliable string ω is calculated by summarizing entropy of all components as $E = \sum_{i=1}^{n} H(\omega_i)$. According to the Gray code quantization, the probability of a bit i receiving value 1 is $p_i = 0.5$. Then, the system achieves the entropy E of 250 and 500 for codeword 255 and 511, respectively. The strength of system security against this attack is measured by Sphere-packing bound according to [10] as:

$$C_{SB} \geq \frac{2^E}{\sum_{i=0}^{t} \binom{E}{i}} \simeq \frac{2^E}{\binom{E}{t}}, \tag{12}$$

where t is the error correcting capability. For two proposed key lengths of 255-bit and 511-bit codewords, the error correcting capability t is 26 and 53 bits as in [22], respectively, so the system achieves C_{SB} as 2^{133} and 2^{269}.

Further more, we analyzed the system under statistical attack that is performed based on the distribution of inter-class Hamming distance of extracted reliable string. Specifically, the adversary can extract the reliable string ω' from his own gait signal. Then, with the inter-class Hamming distance as h, he knows that he can guess the string ω of enrolled user by searching for all ω satisfying $d_{H(\omega',\omega)} = h$. Additionally, by utilizing the error correcting capability of BCH code as t, he only needs to search for all ω such that $d_{H(\omega',\omega)} = h - t$ in order to retrieve key m from secure δ. Let $d = h - t$, then the cost of this attack is

$$C_{ST(h)} = \binom{n}{d} = \frac{n!}{d!(n-d)!}. \tag{13}$$

We assume that h follows the Gaussian distribution. We estimate the mean μ_h and variance σ_h of h in Figure 6. Then, we analyze $C_{ST(h)}$ with h at $(\mu_h - 2\sigma_h)$ and $(\mu_h + 2\sigma_h)$ using (13). With 4-bit quantization, the security strength are 108 and 235 bits corresponding to codeword 255 and 511 bits, respectively.

In summary, as the attack on error correcting capability and Hamming distance are more costly than doing brute force on key, the system security is 87 and 148 bits according to 255-bit and 511-bit codewords, respectively.

5 Related Work

Biometric Cryptosystems (BCS) are techniques for securing biometric templates, and also provide approaches to integrate biometrics and existing security solutions (i.e., symmetric cryptography, password-based authentication) by releasing biometric-dependent key [28]. BCS techniques are classified into two main approaches, namely key binding and key generation. In the key binding approach, biometric templates are used to hide/retrieve a pre-specified secret key which can be selected by the user or randomly generated. Fuzzy Commitment [18] and Fuzzy Vault [17] are cryptographic primitives that offer key binding function. On the other hand, the key generation approach directly generates secret key from biometric templates. The cryptographic primitive supporting this approach is Fuzzy Extractor - Secure Sketch [4].

As the concerns of security and privacy have increased tremendously recently, the BCS techniques were widely applied to various biometric traits such as face

[21], iris [27], fingerprint [14,23], speech [15], gait [11] and achieved promising results. Most of studies followed the key binding scheme [11,14,15,27]. For example, the authors in [27] proposed an Adaptive FCS to secure the iris-code and achieved 0% FAR and FRR of 4.92% with 128 bits security. Having to note that, as the great variation of biometric templates in nature, the task of directly generating stable and high-entropy secret key from biometric template is challenging [24,16]. Several studies on key generation on biometric samples have been proposed [21,30].

A number of studies also proposed methods to protect gait templates (e.g., [11,26,34]). In [11], the authors applied FCS to secure inertial sensors based gait signals. In [26], the authors proposed a two-factor authentication scheme named Gait-hashing. They used hash code generated from camera-based gait data and random vectors stored in token for authenticating user, and achieved EER of 10.8%. The authors in study [34] proposed Key-gait which was a scheme for generating shared secret key between two legitimate devices using gait signal captured from wearable sensors, and can generate 128-bit key with 98.3% probability.

6 Conclusion

In this paper, we addressed the problems of inter-class's low discrimination and intra-class's high variance of nature gait data, which have not been received much attention in the privacy-preserving gait authentication community. We proposed a method that applied LDA to increase the discrimination of gait data, and adopted the Gray code quantization to extract a highly stable binary template. Finally, we proposed a strategy to extract a reliable binary string from the stable binary template, which is used as an efficient input for FCS. The achieved results showed that our proposed system enhances not only the security but also the performance of the system, compared with other state-of-the-art works.

7 Acknowledgments

This research was supported by Korea NRF-2017R1D1A1B03035343 project.

References

1. L. Andrews. Special functions of mathematics for engineers. 1992.
2. J. E. Cutting, L. T. Kozlowski, et al. Recognizing friends by their walk: Gait perception without familiarity cues. *Bulletin of the psychonomic society*, 9(5):353–356, 1977.
3. M. Derawi and P. Bours. Gait and activity recognition using commercial phones. *computers & security*, 39:137–144, 2013.
4. Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *International conference on the theory and applications of cryptographic techniques*, pages 523–540. Springer, 2004.

5. R. W. Doran. The gray code. *J. UCS*, 13(11):1573–1597, 2007.
6. R. A. Fisher. The use of multiple measures in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.
7. J. Frank, S. Mannor, J. Pineau, and D. Precup. Time series analysis using geometric template matching. *IEEE transactions on pattern analysis and machine intelligence*, 35(3):740–754, 2013.
8. M. Gadaleta and M. Rossi. Idnet: Smartphone-based gait recognition with convolutional neural networks. *arXiv preprint arXiv:1606.03238*, 2016.
9. F. Gray. Pulse code communication, Mar. 17 1953. US Patent 2,632,058.
10. F. Hao, R. Anderson, and J. Daugman. Combining crypto with biometrics effectively. *IEEE transactions on computers*, 55(9):1081–1088, 2006.
11. T. Hoang, D. Choi, and T. Nguyen. Gait authentication on mobile phone using biometric cryptosystem and fuzzy commitment scheme. *International Journal of Information Security*, 14(6):549–560, 2015.
12. T. Hoang, D. Choi, and T. Nguyen. On the instability of sensor orientation in gait verification on mobile phone. In *2015 12th International Joint Conference on e-Business and Telecommunications (ICETE)*, volume 4, pages 148–159. IEEE, 2015.
13. T. Hoang, V. Vo, T. Nguyen, and D. Choi. Gait identification using accelerometer on mobile phone. In *2012 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 344–348. IEEE, 2012.
14. Y. Imamverdiyev, A. B. J. Teoh, and J. Kim. Biometric cryptosystem based on discretized fingerprint texture descriptors. *Expert Systems with Applications*, 40(5):1888–1901, 2013.
15. K. Inthavisas and D. Lopresti. Speech biometric mapping for key binding cryptosystem. *Biometric Technology for Human Identification VIII (SPIE Defense, Security, and Sensing), Orlando, FL*, pages 80291P–1, 2011.
16. A. K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 2008:113, 2008.
17. A. Juels and M. Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237–257, 2006.
18. A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 28–36. ACM, 1999.
19. H. Kaur and P. Khanna. Biometric template protection using cancelable biometrics and visual cryptography techniques. *Multimedia Tools and Applications*, 75(23):16333–16361, 2016.
20. E. J. Kelkboom, J. Breebaart, I. Buhan, and R. N. Veldhuis. Maximum key size and classification performance of fuzzy commitment for gaussian modeled biometric sources. *IEEE Transactions on information forensics and security*, 7(4):1225–1241, 2012.
21. M.-H. Lim, A. B. J. Teoh, and K.-A. Toh. An efficient dynamic reliability-dependent bit allocation for biometric discretization. *Pattern Recognition Journal*, 45(5):1960–1971, 2012.
22. R. H. Morelos-Zaragoza. *The art of error correcting coding.* John Wiley & Sons, 2006.
23. M. Morse, J. Hartloff, T. Effland, J. Schuler, J. Cordaro, S. Tulyakov, A. Rudra, and V. Govindaraju. Secure fingerprint matching with generic local structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 84–89, 2014.

24. I. Natgunanathan, A. Mehmood, Y. Xiang, G. Beliakov, and J. Yearwood. Protection of privacy in biometric data. *IEEE access*, 4:880–892, 2016.
25. C. Nickel and C. Busch. Classifying accelerometer data via hidden markov models to authenticate people by the way they walk. *IEEE Aerospace and Electronic Systems Magazine*, 28(10):29–35, 2013.
26. C. Ntantogian, S. Malliaros, and C. Xenakis. Gaithashing: a two-factor authentication scheme based on gait features. *Computers & Security Journal*, 52:17–32, 2015.
27. C. Rathgeb and A. Uhl. Adaptive fuzzy commitment scheme based on iris-code error analysis. In *2010 2nd European Workshop on Visual Information Processing (EUVIP)*, pages 41–44. IEEE, 2010.
28. C. Rathgeb and A. Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011(1):3, 2011.
29. C. E. Shannon, W. Weaver, and A. W. Burks. The mathematical theory of communication. 1951.
30. W. Sheng, S. Chen, G. Xiao, J. Mao, and Y. Zheng. A biometric key generation method based on semisupervised data clustering. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(9):1205–1217, 2015.
31. H. Sun and T. Yuao. Curve aligning approach for gait authentication based on a wearable accelerometer. *Physiological measurement*, 33(6):1111, 2012.
32. S. Theodoridis and K. Koutroumbas. Pattern recognition–fourth edition, 2009.
33. N. T. Trung, Y. Makihara, H. Nagahara, R. Sagawa, Y. Mukaigawa, and Y. Yagi. Phase registration in a gallery improving gait authentication. In *2011 International Joint Conference on Biometrics (IJCB)*, pages 1–7. IEEE, 2011.
34. W. Xu, C. Javali, G. Revadigar, C. Luo, N. Bergmann, and W. Hu. Gait-key: A gait-based shared secret key generation protocol for wearable devices. *ACM Transactions on Sensor Networks (TOSN)*, 13(1):6, 2017.
35. Z. Xu and M. Xia. Distance and similarity measures for hesitant fuzzy sets. *Information Sciences*, 181(11):2128–2138, 2011.
36. Y. Zhao and S. Zhou. Wearable device-based gait recognition using angle embedded gait dynamic images and a convolutional neural network. *Sensors*, 17(3):478, 2017.