Speech Biometric Mapping for Key Binding Cryptosystem

Inthavisas K. and Lopresti D.

The Department of Computer Science & Engineering, Lehigh University,
Bethlehem, PA 18015, USA
Emails: {kei206, dal9}@lehigh.edu

ABSTRACT

We propose a new scheme to transform speech biometric measurements (feature vector) to a binary string which can be combined with a pseudo-random key for a cryptographic purpose. We utilize Dynamic Time Warping (DTW) in our scheme. The challenge of using DTW in a cryptosystem is that a template must be useful to create a warping function, while it must not be usable for an attacker to derive the cryptographic key. In this work, we propose a hardened template to address these problems. We evaluate our scheme with two speech datasets and compare with DTW, VQ, and GMM speaker verifications. The experimental results show that the performance of the proposed scheme outperforms VQ and GMM. It is slightly degraded when compared to the DTW speaker verification. The EERs against attackers utilizing the hardened template are 0% both datasets.

Keywords: Cryptography, Secure template, Speaker verification

1. INTRODUCTION

For more a decade, biometrics as cryptography has been an interesting area because of the inability of humans to remember strong cryptographic key.[2, 20] The traditional approach uses a password to release a cryptographic key, but it is easy to guess using dictionary attacks.[6] Hence, users have to select unusual keys for their passwords that are easy to forget. To address these problems, biometrics are used to combine or generate a cryptographic key to apply to applications such as file encryption and user authentication for two reasons. First, it is hard to get past the biometrics compare to a common eight character password. Second, biometrics are human characteristics, so they cannot be forgotten.

When comparing two sequences of the speech biometric, the main problem is that the duration of the same biometric provided by the same user at a different time changes with non-linear expansion and contraction. The solution to this problem is to use DTW to set up a non-linear mapping of one signal to another by minimizing the distance between two signals.[27] To utilize DTW, we need a template as a *keying signal* to set up a warping function for incoming inputs. This information must be stored in a secure fashion (e.g., on a token such as a tamper resistant smart card, and making a strong template that cannot be transformed to matching features). Our design is focused on the strong template; thus a template protection security is a crucial issue for utilizing DTW. An ideal biometric template protection scheme should possess four properties.[17] 1) Diversity: Different templates must be used for different applications. 2) Revocability: A compromised template can be canceled and re-issued. 3) Security: It must be computationally hard to invert the secure template to the original template. 4) Performance: The system using the secure template should not degrade the recognition performance. Speech biometric satisfies the first two properties as the users can easily change their biometric samples. The remaining properties are the critical issues that we will focus on.

The template protection approaches that are proposed in the literature can be classified into two categories:[13] *feature transformation* and *biometric cryptosystem.* For the first approach, a one-way function is typically applied to a template and it is computationally hard to invert the transformed template. The scheme utilizing a one-way function is called *non-invertible transforms.* The difficulty to design transformation functions that satisfy both the discriminability and non-invertibility is the main drawback of this approach.[13] Another scheme for this approach is called *salting.* The salting scheme uses an invertible transformation function that is parameterized by a random key or a password to transform a template. This scheme also suffers from transformed features. Furthermore, if a random key or a password is compromised, it can be used to recover the biometric template. For the second approach, the public information that does not significantly reveal the biometric template is stored. This information is referred to as *helper data*. During the matching process, the helper data and the biometric are used to derive a cryptographic key. The system that directly uses the helper data and the biometric to generate the cryptographic key is called a *key generation cryptosystem*. If the biometric is used to extract the cryptographic key from the helper data, the system is called a *key binding or key regeneration cryptosystem*. The system that uses more than one scheme will be referred to as *hybrid schemes*.

We protect the DTW template using the idea similar to the non-invertible transformation scheme. The *Hardening* algorithm (Section 3.1) is proposed to perturb the original template by removing some frequency-domain features from the template. Finally, the rest of features will be transformed to a time-domain template that refers to as a *hardened template*. This template will be used as a keying signal in DTW process. The Discrete Fourier Transform (DFT) and the inverse DFT (IDFT) will be used to create a stored or hardened template. More precisely, the following is a definition of a hardened template. Given a DFT vector (full template) $X = \{x_i,\ i = 1, \ldots, F\}$, *a hardened template* $\mathcal{H}_T$ is an IDFT of a hardened vector $\mathcal{H} = \{X | \exists x_i = 0\}$ such that the hardened template must be useful to create a warping function, while it must not be usable for an attacker to derive the cryptographic key.

The next step is to regenerate a cryptographic key. The key binding approach is used to protect the key. We refer this template (key binding) to as a lock data $\mathcal{L}$ or *a binary template*.

The other problem is the correlation among features. Hao reported that "an iris code usually has a run length of 8 consecutive '1's or '0's.[10]" For speech (e.g., Monrose et al.'s scheme[21]), we cannot specify the exact length of repetition. It depends on the number of phonemes in a pass-phrase and the idiosyncrasy of each user when he/she utters the pass-phrase. However, the consecutive '1's or '0's will lower the randomness of the key. We address this problem by proposing the *Mapping* algorithm (Section 3.2) using *a multi-threshold template* $\mathcal{T}$ that are determined from pseudo-random bits (Section 3.3). Hence, the algorithm can generate a binary string that an observer cannot predict.

2. RELATED WORK

A number of researchers proposed biometric cryptosystems. Soutar et al. proposed a first practical approach using the interaction of a biometric image with a secure block of data.[29] They used a filter function which is derived from an image in such a way that the filter function could produce a consistent key while it cannot be used to retrieve information about the image or key. Monrose et al.[19] proposed a behavioral biometric key generation based on a keystroke biometric. They use dynamic features (duration of keystrokes and latencies between keystrokes) to strengthen a user's password. This scheme makes the system more secure by adding 15 bits of entropy to the password for 15 dynamic features.[31] In their later works,[20–22] they applied this scheme to voices. The algorithm to generate cryptographic keys from voices is mainly based on the speaker verification and identification technologies such as digital signal processing, feature extraction, and the vector quantization technique. Consequently, their system was eventually able to generate cryptographic keys up to 60 bits from voice features. Garcia-Perera et al.[9] proposed a way to generate cryptographic keys based on speech recognition. The phoneme of the user's pass-phrase was trained and mapped to binary by using a Support Vector Machine (SVM) classifier. However, their scheme can generate a short length of key; the bit length is equal to the number of phoneme in the pass-phrase. Hao and Chan[11] proposed a way to generate biometric keys from hand-written signatures. The DTW template was protected by utilizing static features as the DTW template so that the template did not reveal the key that was generated from dynamic features. Their approach achieved 40 bits of entropy with FAR 1.2%. At a later time, Hao et al.[12] proposed the combining of a biometric and cryptography with a two factors scheme: a biometric and a token. They stored a lock data (encoded keys combine with a biometric) in a smart card which can be unlocked and decrypted at later time by a user biometric. The template was hidden by following the fuzzy commitment scheme.[14] They were able to generate 140 bits from iris codes with 44 bits of entropy.

Ratha et al.[25] proposed cancelable fingerprint templates. The non-invertible transformation functions were used to transform fingerprint features (minutiae) position so that the matcher can still be applied in feature domain. The result showed that there was a trade-off between discriminability and non-invertibility. In this proposal, three transformation functions were proposed: cartesian, polar, and functional. The Cartesian transformation yielded the best security. However, the performance was relatively poor. In addition, Nagar et al.[23]

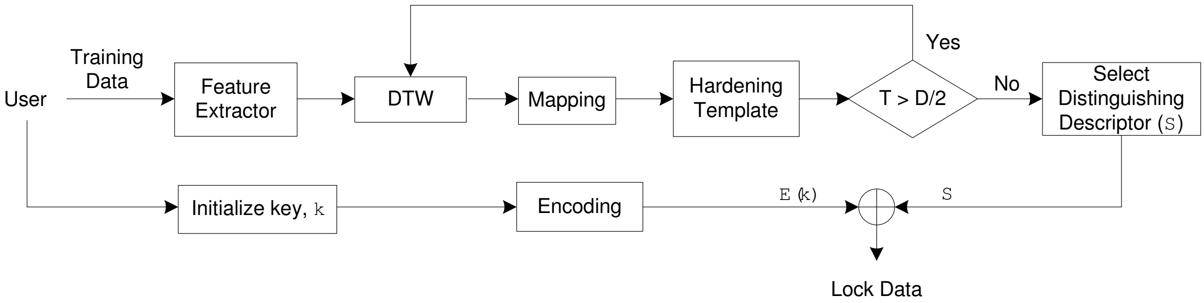

Figure 1. Biometric key binding in training phase.

have shown that Ratha et al.'s scheme was vulnerable to intrusion attack because it was relatively easy to obtain a pre-image of the transformed template.

Maiorana et al.[16] protected a DTW template by transforming the template in a non-invertible way for signature authentication. Even though the original template cannot be recovered, the system left the transformed template which could be used in matching process. Teoh and Chong[30] proposed secure speech template protection in speaker verification system. The speech template was hidden by the random subspace projection process. The problem is the same as Maiorana et al.'s system in case that the random key is compromised.

In this work, we construct a cryptographic key regeneration similar to others described above, but we focus on how to reliably, securely, and randomly (in the context of cryptography) generate a binary string from biometrics. The DTW will make our scheme more reliable while the hardened template maintains security. Finally, a multi-thresholds scheme will help our scheme generate a binary string unpredictably to maximize the entropy of the template. The following sections describe our scheme to generate a cryptographic key and discuss the evaluation of the scheme. Feature extraction is the first process to derive voice features. This process involves speech processing described in Section 3. Next, we describe the scheme to generate a cryptographic key from the extracted features. First, we describe the hardening scheme (Section 3.1), and then we describe the mapping scheme (Section 3.2). Lastly, we describe the scheme to generate multi-thresholds (Section 3.3). In Section 3.4, we describe the biometric key retrieval process. In Section 4, we describe the datasets used for evaluation. In Section 5, we describe the experiments to evaluate the performance of our scheme and discuss the results.

3. BIOMETRIC KEY BINDING (BKB)

Our design can be overviewed as two phases: training and verification. The biometric key binding is in the training phase indicated in Figure 1. Users provide their training pass-phrases that are repeated *l+1* times to the system. Feature extraction is the first process to derive feature vectors and Discrete Fourier Transform (DFT) features. This process involves digital signal processing detailed as the following. We use a low-pass digital filter with a cut-off at 4 kHz to strip the higher frequencies from the signal. The next step is pre-emphasis, which is the process to raise the Signal to Noise Ratio. The signal is pre-emphasized by passing the signal to a first order digital filter $H(z) = 1\text{-}\alpha z^{-1}$, where we set $\alpha = 0.98$. Framing is the next step. The signal is framed into the short time analysis interval. Each frame is multiplied by a window function to reduce abrupt changes at the start and the end of each frame. These frames have to be overlapped properly. The length of each frame is usually around 30 msecs; this length would yield good results for speech processing with 10 msecs overlap.[7] For the sampling rate of 8 kHz, we use 240 samples per frame that are shifted every 80 samples. The system is initialized by using one of the training utterances as the keying signal which is stored as 121 DFT features of m frames. Then the system performs DTW to the rest of training utterances. The feature vectors of each utterance (m frames) will be mapped, a frame per bit, to a binary string of length m called *a set of feature descriptors*. Lastly, l sets of feature descriptors are used to define *distinguishing features*: features of length D that the user can reliably generate. The binary string of distinguishing features derived from the training utterances is called *distinguishing descriptors*. The mapping and defining the distinguishing features procedure are detailed in Section 3.2.

We initialized the template by using a full set of DFT features. However, we are not able to use the full template as attackers can utilize it to derive the cryptographic key. Hence, the template has to be perturbed which is what we call *hardening the template* and we refer the result to as a *hardened DTW template.* We set the goal of hardening the template by the following statement: the attacker utilizing a hardened template should not be better than *the simplest attack* where the attacker randomly guesses the distinguishing descriptors.

Specifically, let the total number of bit derived from the hardened template that corresponds to the distinguishing descriptors be *T*; the system should yield T as less than or equal to *D/2.* The motivation is due to the hardening goal. For a simplest attack, any random bits are equally likely to be 1 or 0. Hence, the expected proportion of agreeing bits between the simplest attack and the template (distinguishing descriptors) is 0.5 or $D/2$.

If this condition hold, $T \leq$ *D/2*, the template will not help the attackers as they just using a simplest attack is easier (better). For this reason, if T is greater than *D/2*, the template will be hardened (see Section 3.1). After each step in hardening the template, the new hardened DTW template will be the keying signal of the training pass-phrases and the process will be re-started until the condition is met. Finally, the IDFT of the latest hardened DTW template is stored as a hardened template $\mathcal{H}_T$ and 2^n-1 distinguishing descriptors, where $n = 3, 4, \ldots$, will be selected based on feature variation to form a binary string *S.*

Once the hardened template is set, a pseudo-random key k is generated and then encoded properly denoted by *E(k).* In our case, we use BCH code.[15] The encoding code *E(k)* has to tolerate error within Hamming distance (H), a maximum number of bit differences between the distinguishing descriptors and the feature descriptors of a legitimate user. For the next step, the S and the encoding code *E(k)* will be hidden using an XOR operation and then stored as a lock data denoted by $\mathcal{L}$. Only the user with feature descriptors S' that is sufficiently similar to the S within Hamming distance ($|S - S'| \leq H$) can unlock the $\mathcal{L}$ and correctly decode the key. We refer to the fuzzy commitment scheme[14] for more detail.

3.1 Hardening template

As described earlier, the DFT features should be used to create a template to be a keying signal. The template is m frames of 121 DFT features each. We need to store a hardened template in order to set the time alignment to the input signal using DTW technique. This template should not be used to derive the key. The straightforward way is to enumerate over m frames of the original template then choose a set of optimal features that yield T $\leq$ *D/2*, but the computational time is not possible. Hence, the optimal search algorithm should be employed. We choose a Sequential Backward Search (SBS) that is a top down search procedure starting from the full set of features and remove one feature per step until the condition is met.[24] By using SBS, it is easy to terminate the program under the assumption we described earlier.

To start, a user presents l+1 training pass-phrases to the system. Then, the sets of DFT features, $\beta_1, \ldots, \beta_{l+1}$, are extracted from the pass-phrases and these sets are used as the inputs (global) of the *Hardening* algorithm (see Algorithm 1). Next, the threshold is initialized with Ω, the mean of the linear combination of all components in the DFT features (vectors) of the biometric samples. The β_1 is used as the initialized *hardened DTW template* ($\mathcal{H}$) and the ψ is a list of DFT indexes. For the hardening process, the algorithm will search for a DFT feature in $\mathcal{H}$ that yields the least T when that DFT feature is substituted with 0. Then the index of the substituted DFT is removed from the ψ (see Algorithm 1, lines 2-8). The hardening process includes the *Mapping* algorithm that will return T and D (more detail will be explained in Section 3.2); the next step, the system checks whether T is greater than $D/2$. The above described steps are iterated until T less than or equal to *D/2* (see lines 9-11).

When the recursion is terminated, the algorithm will generate multi-thresholds $\mathcal{T}$ (see Section 3.3) that is used to derive distinguishing descriptors. Next, the IDFT of $\mathcal{H}$ is stored as the hardened template ($\mathcal{H}_T$ in line 13). The algorithm then inputs the $\mathcal{H}_T$ and the multi-thresholds $\mathcal{T}$ into the *Mapping* algorithm. Consequently, it yields the distinguishing descriptors and their relevant indexes (B and *indexes* in line 14). The last step, we select 2^n-1 the least variation of the distinguishing features, where $n = 3, 4, \ldots$, to form a binary string S and a lock data $\mathcal{L}$ (see lines 15-17). Finally, the system securely deletes a set the training parameters using a *Delete* function and stores $\mathcal{L}$, $\mathcal{T}$, $\mathcal{H}_T$, and Ψ in the database.

Algorithm 1 Specification of the Hardening algorithm

```
Input: The biometric samples $\beta_1, \dots, \beta_{l+1}$
Output: The lock data $\mathcal{L}$, multi-thresholds $\mathcal{T}$, hardened template $\mathcal{H}_T$, and relevant indexes $\Psi$
Initialize: $\mathcal{T} \leftarrow \Omega$, $\psi = \{1, \dots, 121\}, \zeta \leftarrow 121, [D, T] \leftarrow m, \mathcal{H} \leftarrow \beta_1$
Hardening($\psi, \mathcal{H}$)
    for $j \leftarrow 1$ to $\zeta$
        $\mathcal{H}' \leftarrow \mathcal{H}$
        $\mathcal{H}'(\psi(j)) \leftarrow 0$
        $[T', D'] \leftarrow$ Mapping($\mathcal{T}, \mathcal{H}'$)
        if $T' < T$
            $T \leftarrow T'$, $D \leftarrow D'$, $index \leftarrow j$
    $\mathcal{H}(\psi(index)) \leftarrow 0$, Remove($\psi(index)$), $\zeta \leftarrow \zeta - 1$
    if $T > D/2$ and $\zeta > 1$
        Hardening($\psi, \mathcal{H}$)
    return $\mathcal{H}$
$\mathcal{T} \leftarrow$ MultiThreshold($\mu, \sigma, \kappa$)
$\mathcal{H}_T \leftarrow$ IDFT($\mathcal{H}$)
$[B, indexes] \leftarrow$ Mapping($\mathcal{T}, \mathcal{H}_T$)
$\Psi \leftarrow \{\Psi(1), \dots, \Psi(2^n - 1)\}$ such that $\sigma(\Psi(i)) < \sigma(\Psi(i+1))$ and $\Psi \subset indexes$
$S \leftarrow B(\Psi)$
$\mathcal{L} \leftarrow E(k) \oplus S$
Delete($\{\beta_1, \dots, \beta_{l+1}\}, p, \mu, \sigma, \kappa,\ B,\ S,\ indexes$)
```

3.2 Mapping the biometric to a binary string

Algorithm 2 is used to map feature vectors to a binary string. First, the algorithm performs DTW between $\mathcal{H}$ and $\beta_k,\ k = 1, \dots, l+1$. The results are represented with f_k. For each frame of f_k, let $f_k(i)$ represents a feature vector, where i = *1, ..., m*, is the number of frame. We compute $f'_k(i)$ from the linear combination of all components in $f_k(i)$ and then set a biometric feature $\phi_k(i) = f'_k(i) - \mathcal{T}(i)$ where $\mathcal{T}$ is a set of thresholds (see lines 2-5). Binarization is the next step. The $\phi_k(i)$ is mapped to a feature descriptor, $b_k(i)$, by testing whether $\phi_k(i)$ is positive or negative. It will be mapped to 1 if it is positive and 0 otherwise (see lines 6-7). The last step is to define distinguishing features that the user can reliably generate. In other words, any binary strings derived from the distinguishing features of any β_k should be identical. Therefore, a bitwise XORing of the binary strings will be 0. For this reason, we determine XORing of, $b_k(i)$, k = *2, ..., l+1.* If the XORing of $b_k(i)$ is zero, the i^{th} feature will be a distinguishing feature and we set $B(i) = b_k(i)$ (see lines 8-13). Here, the B is a set of distinguishing descriptors of length D (see line 14). Next, the algorithm returns a set of indexes of the distinguishing features (*indexes*), B, and D. Finally, the hardened template is examined and the algorithm returns the number of bits T that corresponds to distinguishing descriptors (see line 15).

3.3 Multi-thresholds generation

We select a set of thresholds in such a way that the entropy of the biometric template is maximized. According to Jain et al., the entropy of the biometric template can be understood as a measure of the number of different identities that are distinguishable by a biometric system.[13] Hence, the set of thresholds that is used in mapping process should yield a binary string that appears to be random in a context of cryptography.

We first generate pseudo-random bits $p \in \{0, 1\}^m$ using Blum Blum Shub (BBS) algorithm.[3] Next, a set of thresholds is selected based on the criteria that a query biometric will be mapped to a binary string that is close to p. Finally, the pseudo-random bits will be securely deleted. As the *Mapping* algorithm simply maps a feature to 1 if the feature is greater than a threshold and 0 otherwise, hence we select a threshold to be lower than the mean of that feature if a corresponding pseudo-random bit is 1 and greater than the mean otherwise. Specifically, to generate the multi-thresholds for any users, the *MultiThreshold* function is used in *Hardening* and *Mapping* algorithm. Let $\mu(i)$ and $\sigma(i)$ be the mean and standard deviation of the linear combination of all features of i^{th} frame over l training utterances, the function executes as follows:

Algorithm 2 Specification of the Mapping algorithm

```
Input: $\mathcal{T}, \mathcal{H}$
Output: $T, D, B, indexes$
$indexes \leftarrow \{\}, \beta_1 \leftarrow \mathcal{H}$
for $k \leftarrow 1$ to $l+1$
    $f_k \leftarrow$ perform DTW of $\mathcal{H}$ and $\beta_k$
    for $i \leftarrow 1$ to $m$
        $f'_k(i) \leftarrow$linear combination of all components in $f_k(i)$
        $\phi_k(i) \leftarrow f'_k(i)$-$\mathcal{T}(i)$
        if $\phi_k(i) > 0$
            $b_k(i) \leftarrow 1$ otherwise $b_k(i) \leftarrow 0$
for $i \leftarrow 1$ to $m$
    $b(i) \leftarrow 0$
    for $k \leftarrow 3$ to $l+1$
        $b(i) \leftarrow b(i) + (b_2(i) \oplus b_k(i))$
    if $b(i) = 0$
        $B(i) \leftarrow b_2(i)$, $indexes \leftarrow indexes \cup i$
$D \leftarrow$ range $indexes$
$T \leftarrow$ the number of bits such that $b_1(indexes(i)) \oplus B(indexes(i)) = 0$
return $T, D, B, indexes$
```

1. Generate pseudo-random bits $p \in \{0,1\}^m$ using BBS algorithm.[3]
2. Set the multi-thresholds $\mathcal{T}(i) = \mu(i)+(-1^{p(i)})\ \kappa_i\sigma(i)$ for some parameter $\kappa_i\ > 0$ which maximize the distinguishing descriptor and minimize the error rate.
3. Securely delete pseudo-random bits

3.4 Biometric key retrieval

The biometric key retrieval process is in the verification phase indicated in Figure 2. The user requests the template from the database that contains the hardened template, the multi-thresholds, and the lock data. Then the system performs DTW to a user's pass-phrase. The signal that resulted from DTW is executed using the algorithm similar to Section 3.2 to generate feature descriptors, and the feature descriptors of the distinguishing features (feature descriptors of the relevant indexes in the Ψ) will be XORed with the lock data. The next step is the decoding process. If the error is within the tolerance, the key can be correctly reconstructed. To check whether the key is identical to the key generated in the training phase, a number of researchers[1,10,22] checked the hash function. In the training phase, the initialized key, k, was stored as h(k). Once the key k', is regenerated from the verification phase, the system checks to see whether h(k) = h(k'). If h(k) = h(k'), the key, k', is correct.

4. DATASETS

We will compare BKB with other speaker verifications using Equal Error Rate (EER) with two databases: The MIT mobile device speaker verification corpus (MDB)[32] and A data set in quiet environment (QDB). The MDB is a public database available by MIT. The QDB is our database collected over a month period.

4.1 The MIT mobile device speaker verification corpus

This database was collected from 48 speakers (22 females and 26 males). The utterances were recorded in three acoustic environments: office, lobby, and intersection via two types of microphones: external earpiece headset and built-in mobile device. The database consists of two sets: a set of enrolled users and a set of dedicated imposters. For the enrolled set, speech data was collected over two sessions on separate days (20 minutes for each session). For the imposter set, users participated in a single 20 minutes session. There are six lists of pass-phrases that were varied by three environments and two types of microphones. We select the first list to

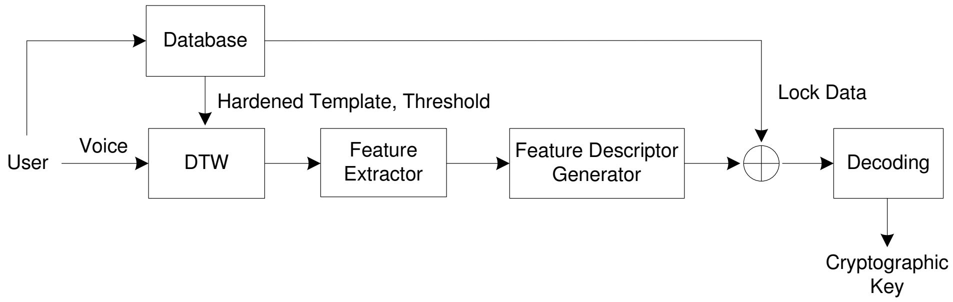

Figure 2. Biometric key retrieval in verification phase.

our experiment because it provided pass-phrases that were said by the same speaker multiple times under the same environment (office). So, we can use this list in the training and the testing phase.

4.2 A data set in quiet environment

This database contains 4,320 recordings collected on a laptop computer via an external earpiece headset microphone from 6 male speakers during several rounds. The data collection was taken in the graduate study room at Lehigh University's Library that can be referred to as quiet environment. In the first round, the subjects were asked to say their 5 pass-phrases. Each pass-phrase was uttered 10 times. In addition, they were asked to say 270 short sentences to make a speech corpus. In the second round, they were asked to say their same set of pass-phrases. Each was uttered five times. Furthermore, they were asked to say other subjects' pass-phrases. Each was uttered five times. Lastly, they were asked to imitate the other subject pass-phrases by listening to the pass-phrases that we replayed to them. Each pass-phrase was uttered five times. By listening to imitated pass-phrases, we selected the best imitator for the third round. The best imitator was asked to mimic the target speaker's pass-phrases. Each pass-phrase was uttered five times.

5. EXPERIMENTS AND RESULTS

5.1 Experiments setup

We compare BKB with other speaker verification systems: Dynamic Time Warping (DTW),[8] Vector Quantization (VQ),[28] and Gaussian Mixture Model with Universal Background Model(GMM-UBM).[26] For all constructions of the speaker verifications, we use 13 orders Mel-Frequency Ceptral Coefficient (MFCC) for training and verification.

For BKB, 121 DFT elements of a full template are reduced to an average of 9 and 11 for the MDB and QDB. We set the length of the binary string to 511 bits. For the MDB, we can generate 139 bits on average for each feature; we need 4 features to generate 511 bits. For our setting, 4 features are the Short-Term Energy, the 13 orders MFCC, the 12 orders Linear Prediction Coefficient (LPC), and the DFT. Nevertheless, some pass-phrases cannot generate a binary string of length 511. In this case, we use a zero padding scheme to adjust the lengths of the binary string of these pass-phrases to that length. For the QDB, we can generate 221 bits for each feature. However, we use the same features in the MDB.

For DTW, we use the first utterance as the keying signal and perform DTW to the rest. The results are averaged and stored as the matching template. The distance between an input and the template is determined by using the Euclidean distance. The system decides whether to accept or reject the speaker by comparing the Euclidean distance to the decision threshold.

For VQ, the acoustic models of speakers are created by partitioning a collection of acoustic feature vectors into C clusters.[28] Each cluster is represented by an average vector or centroid denoted by c_i for $i = 1, \ldots, C$. For our setting, the K-means clustering is used to quantize the training vectors. We investigate the performance of VQ in our datasets by setting the number of codebooks to 10, 20, 30, 40, and 50. The performance with 30 codebooks yields the best results. Therefore, we set $C = 30$. The distance between an input vector and the

Table 1. Equal Error Rates (EERs) of the various methods in MDB and QDB. The third and fourth columns are the EERs of random pass-phrase and imposter trial. (*) in the second column indicates insecure template.

Database	Method	EER (%)		Error Corrected (bits)
		Random	Imposter	{Random, Imposter}
MDB	DTW*	2.96	11.55	-
	BKB	**5.49**	**11.96**	{42, 38}
	VQ	7.71	16.67	-
	GMM	5.36	13.37	-
QDB	DTW*	3.83	7.60	-
	BKB	**6.81**	**9.31**	{58, 54}
	VQ	10.62	12.12	-
	GMM	6.06	9.50	-

nearest centroid is determined by using the Euclidean distance. The system decides whether to accept or reject the speaker by comparing the distance to the decision threshold.

The GMM model consists of a finite number of Gaussian distributions parameterized by their priori probability π_j, mean vectors μ_j, and covariance matrices Σ_j.[26] In this experiment, we use nodal covariance matrices. We initialize the speaker models using the K-means clustering, then the parameters are estimated by using the EM algorithm.[5] Given an input vector $X = \{x_1, \ldots, x_m\}$, the matching score for GMM is given by the log-likelihood of the GMM $L = log\ p(X|\lambda_j) - log\ p(X|\lambda_{j\prime})$ where $\lambda_j = (\pi_j, \mu_j, \Sigma_j)$ and $\lambda_{j\prime} = (\pi_{j\prime}, \mu_{j\prime}, \Sigma_{j\prime})$ are the model of speaker j and the background model of speaker j. The training utterances of all speakers except speaker j are used to create the background model and the rest is used to create the speaker model of speaker j. We use the GMM mixture order = 10 for the reason similar to the setting of the VQ. The system decides whether to accept or reject the speaker by comparing the log-likelihood to the decision threshold.

5.2 Performance

For the QDB, we use five pass-phrases from each speaker in our experiment, a total of 5*6 = 30 different pass-phrases. Six recordings from the first round are used to train the system. We choose the number of training pass-phrase to six as using more than six recordings does not significantly improve performance. Instead, it just increases the computation time. Similarly, using the number of training pass-phrase less than six noticeably degrades performance. Five recordings from the second round are used for verification. Five recordings of the same pass-phrase uttered by other speakers in the second round are used to evaluate the *imposter trial*, in total of 5*5=25 recordings for each pass-phrase. We randomly select 25 other pass-phrases from other speakers that do not correspond to the verification pass-phrase to evaluate the *random pass-phrase trial*. For the MDB, we use six recordings to train the systems for the same reason in the QDB. Two recordings are used for verification. To investigate the performance of the system, we use the same pass-phrase uttered by other speakers to evaluate the imposter trial. The number of imposters that is available in the database varies from 1 to 6. In addition, we use 47 pass-phrases of other speakers that are different from the verification pass-phrase to evaluate the random pass-phrase trial.

Table 1 shows the recognition performance of the DTW, VQ, GMM-UBM, and BKB. Even though the EERs are high, the results are inline with Woo et al.'s work.[32] For BKB, we conducted the experiments 30 times to reduce statistical fluctuation caused by the different set of random numbers used to set multi-thresholds. Therefore, the results were averaged. The standard deviations of EERs are 0.99 and 0.78 in the MDB and QDB for the imposter trial and 0.74 and 0.71 for the random pass-phrase trial. From the results, the DTW yields the best performance while BKB has the second best results for the imposter trial. However, the difference of the imposter trial between the DTW and BKB is slight (0.41% and 1.71% for the MDB and QDB). As indicated in Table 1, the DTW method is not secure, it left all the biometric information (a full set of DTW template) in the system. Hence, its slightly better performance has no merit because the security and privacy are significantly lost.

It is clear that BKB is noticeably better than VQ and slightly better than GMM-UBM when we compare the recognition performances (Table 1) for the imposter trial. Furthermore, both VQ and GMM-UBM left some

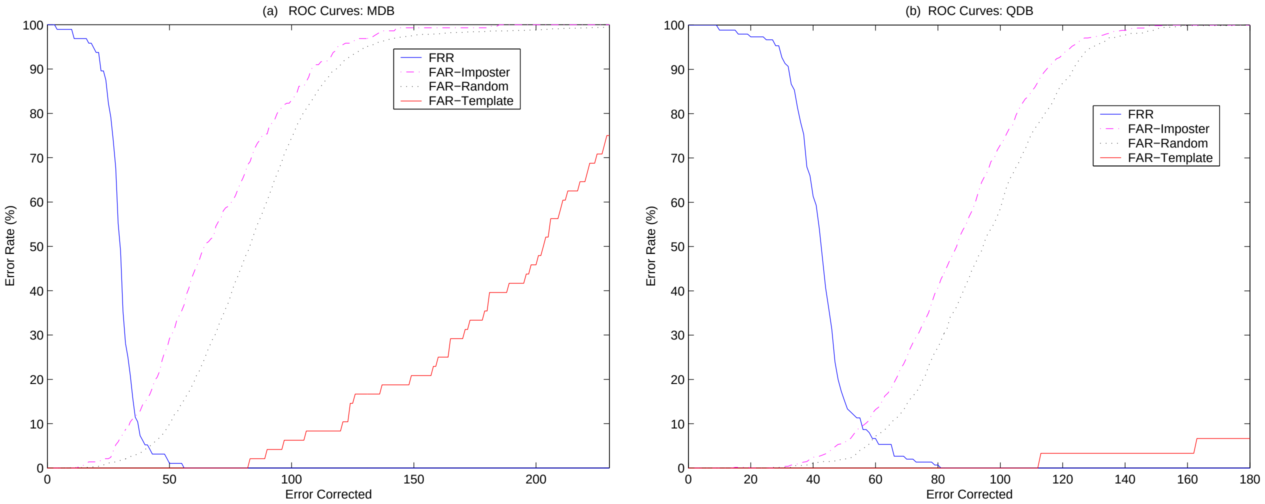

Figure 3. The performance of BKB against attackers using true pass-phrases (Imposter), random pass-phrases (Random), and the templates (Template): (a) MDB (b) QDB.

information that attackers could use to attack the system. For VQ, the attackers can use the centroid and weight function to search for the input signals that are close to the VQ template. Similarly, the GMM left the mean vectors, the variations and the priori probabilities that the attackers can exploit to gain access to the system. We plan to investigate this issue in the future.

The operating points of BKB for imposter trial are 38 and 54 bits for the MDB and QDB (Table 1). We use t-error-correcting BCH[15] which denoted by *BCH(n, k, t)* where n is a block length, k is the key, and t is correctable bits. Hence, *BCH(511, 229, 38)*, and *BCH(511, 139, 54)* are employed for the MDB and QDB. We can generate the cryptographic key up to 229 and 139 bits that exceed the requirement of 128-bit Advanced Encryption Standard (AES).

Figure 3 shows the plots of the recognition performance of BKB against various attacks including the attackers who acquire the hardened template. Assuming that the attackers use a hardened template to derive the key the same way as in the random pass-phrase and imposter trial, the EERs of the template attack are 0% in both databases. However, more analysis of the security of the template is provided in Section 5.3 where the attackers have perfect knowledge of the correlation of the features.

5.3 Security analysis

The security of the scheme is based on the template protection. Our scheme falls under the hybrid schemes. First, the DTW template is protected using a non-invertible transformation scheme. The algorithm will search for a set of optimal features in order to use them as the hardened template. Next, the key binding scheme is applied to protect the key, and then the training data will be securely deleted from the system. It is computationally hard to decode the key without any knowledge of biometric data.[13]

We can estimate the security of the scheme using the sphere packing bound[18] similar to Hao's work.[10] Let z be the uncertainty of voice and w be the error bits that can be corrected by the system, the lower bound can be set to : $\frac{2^z}{\sum_{i=1}^{w}\binom{z}{i}}$.

To estimate the lower bound, we use two verification recordings of each speaker in the MDB. We carry out 4,512 of inter-speaker comparisons to evaluate the uncertainty. The following steps are the uncertainty analysis.[4] For more detail, we refer to Daugman's work.[4] For each comparison, the Normalized Hamming Distance (*NHD*) between two binary templates, A and B, is given as: $NHD = \mathcal{D}_H(A,\ B)$ where $\mathcal{D}_H(A, B)$ is a function to calculate the Hamming distance between A and B. Hence, $NHD = 0$ would represent a perfect match. Figure 4 (a) shows the distribution of the *NHD* of inter-speaker comparisons where $p = 0.5281$ is mean and $\sigma = 0.0455$ is standard deviation of *NHD*. The result in this figure is close to a binomial distribution as shown in Figure

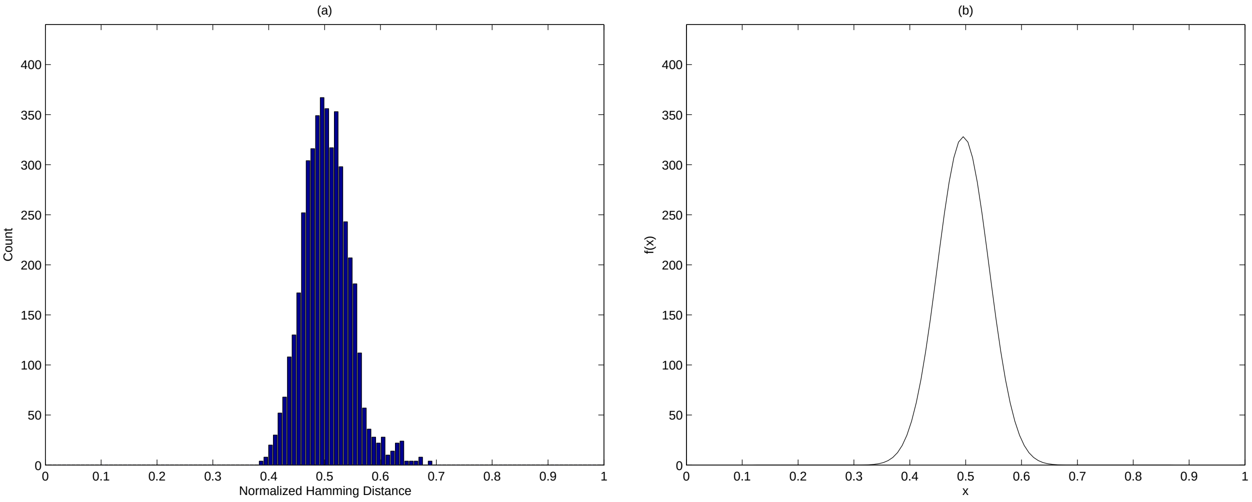

Figure 4. The comparison of a distribution of Normalized Hamming Distance and a binomial distribution (a) Distribution of Normalized Hamming Distances obtained from 4,512 comparisons of inter-speaker in the MDB and (b) A binomial distribution with $p = 0.5$ and $N = 120$ degrees-of-freedom.

4 (b) which has the the fractional function: $f(x) = \binom{N}{y} p^y (1-p)^{N-y},\ x = y/N$, where $N = 120$ and $p = 0.5$. These results indicate that the difference between the binary templates is likely distributed to be a binomial experiment of 120 repeated trials with $p = 0.5$. Therefore, for a binary string of 511 bits, it has approximately 120 degrees-of-freedom. From Table 1, the system should be able to correct the error up to 38 bits (imposter trial), that is approximately 8%. Here, z is 110 bits and w is 10 bits. The estimated entropy is 73 bits.

However, some information may leak from the hardened template and multi-thresholds. Recall that any bits in the binary template were set to be close to pseudo-random bits p, any given bits in the binary template were equally likely to be 1 or 0. If a binary string generated from the hardened template is random, an expected difference of agreeing bits between the binary template and the binary string derived from the hardened template should be close to 256. Upon testing in the MDB, the expected difference is 303 bits, the result implies that some information leaks from the hardened template. In the worst case, we assume that the attacker can correctly locate these bits. Hence, 47 (303-256) bits or $\frac{47}{511} \times 100 = 9.20$ % leak from the hardened template. As a result, the estimated entropy will be 66 bits.

We further carry out 4,512 of inter-speaker comparisons using the global-threshold scheme where the threshold is fixed. The estimated entropy is 16 bits. Table 2 summarizes the security when we compare the multi-thresholds to the global-threshold scheme in the MDB. It is clear that the entropy of the multi-thresholds scheme is significantly improved.

Table 2. The security of the multi-thresholds and the global-threshold scheme in the MDB.

	Multi-thresholds	Global-threshold
Estimated Entropy (bits)	**66**	16

Another security issue we are concerned is the security against potential attacks. Based on the knowledge of the authentic speaker and algorithms, the attacker may exploit these to attack the system. Examples include a generative attack or Hill Climbing. Therefore, these attacks should be investigated.

6. CONCLUSIONS

We addressed two problems in a cryptosystem. First, the problem of the feature correlation could be mitigated by using the proposed multi-thresholds. As a result, the randomness of the key (entropy) was increased from 16 to 66 bits. Second, we addressed the challenge in using DTW in a cryptosystem, more specifically, that the template must be useful to create a warping function, while it must not be usable for an attacker to derive the cryptographic key. A solution, the hardened template was proposed. We showed that the EERs against the attackers using the hardened template were 0%. We compared our system with DTW, VQ, and GMM-UBM speaker verifications. The DTW yielded the best performance while ours had the second best results for the imposter trial. However, the difference between the DTW and ours was slight (0.41% and 1.71% for the MDB and QDB). We noted that the DTW speaker verification is not secure and it leaves all the biometric information (a full set of DFT template) in the system. Hence, its slightly better performance has no merit because the security and privacy are significantly lost.

As this work was a preliminary investigation, the security against potential attacks needs to be further explored. We will start from a naive adversary who does not have knowledge of an authentic speaker to a highly skilled adversary who knows the speaker's information, has the speaker's voice samples, acquires the speaker's template, and knows an algorithm of the speaker verification system. In particular, a generative attack is the most serious attack we are concerned with. We will investigate an analysis-synthesis forgery which the highly skilled adversary can exploit the information such as feature vectors from the template and a statistical probability from the voice of target samples to re-generate a forgery to fool the systems.

7. ACKNOWLEDGMENTS

The authors would like to thank Jim Glass who provided us with the MIT mobile device speaker verification corpus for this research. We gratefully acknowledge the anonymous subjects who devoted time to our data collection. We would also like to thank the anonymous reviewers for their comments. Lastly, the authors would like to thank Candice Quinones, an adjunct of English as a Second Language (ESL) at Lehigh University, for her review of the previous version of this manuscript.

REFERENCES

[1] L. Ballard, S. Kamara, F. Monrose, and M. K. Reiter. Towards practical biometric key generation with randomized biometric templates. In *Proceedings of 15th ACM Conference on Computer and Communications Security*, pages 235-244, Alexandria, VA, October 2008.

[2] L. Ballard, S. Kamara, and M. K. Reiter. The practical subtleties of biometric key generation. In *Proceedings of The 17th Annual USENIX Security Symposium*, pages 61-74, San Jose, CA, August 2008.

[3] L. Blum, M. Blum, and M. Shub. Comparision of two pseudo-random number generators. In *R. L. Rivest, A. Sherman, and D. Chaum, editors, Proc. Crypto'82*, pages 61-78, Plenum Press, New York, 1983.

[4] J. Daugman. The important of being random: statistical principles of iris recognition. *Pattern Recognition*, 36(2): 279-291, 2003.

[5] A. Demster, N. Lair, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc.*, Vol. 39, pages 1-38, 1977.

[6] D. Feldmeier and P. Karn. UNIX password security-ten years later. In *Advances in Cryptology-CRYPTO'89*, pages 44-63. Springer Verlag, London, UK, 1989.

[7] S. Furui. *Digital Speech Processing, Synthesis and Recognition.* Marcel Dekker, Inc., New York, 2001.

[8] S. Furui. Ceptral analysis Technique for Automatic speaker verification. *IEEE Transactions on Acoustics, Speech, Signal Processing*, ASSP-29(2): 254-272, April 1981.

[9] L. P. Garcia-Perera, J. C. Mex-Perera, and J. A. Nolazco-Flores. Multi-speaker voice cryptographic key generation. In *the 3rd ACS/IEEE International Conference on Computer System and Application.* Pages 93-98, 2005.

[10] F. Hao. *On Using Fuzzy Data in Security Mechanisms.* PhD thesis, University of Cambridge, April 2007.

[11] F. Hao, C. W. Chan. Private key generation from on-line handwritten signatures. *Information Management & Computer Security*, Issue 10, No. 2, pages 159-164, 2002.

[12] F. Hao, R. Anderson, and J. Daugman. Combining cryptography with biometrics effectively. *IEEE Transactions on Computer*, 55(9):1081-1088, September 2006.

[13] A. K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP Journal on Advances in Signal Processing Special Issue on Biometrics*, January 2008.

[14] A. Juels and M. Sudan. A fuzzy commitment scheme. In *Proceeding of the 6th ACM Conference on Computer and Communication Security*, pages 28-36, November, 1999.

[15] S. Lin, and D.J. Costello, Jr. *Error Control Coding Fundamentals and Applications.* Prentice-Hall, N.J., 1983.

[16] E. Maiorana, P. Campisi, and A. Neri, Template protection for dynamic time warping based biometric signature authentication. In *Proceedings of the 16th international conference on Digital Signal Processing*, Santorini, Greece, pages 526-531, 2009.

[17] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*, Springer-Verlag, 2003.

[18] F. J. McEliece and N. J. A. Sloane. *The theory of Error correcting codes*, North Holland, 1991.

[19] F. Monrose, M. K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 73-82, November 1999.

[20] F. Monrose, M. K. Reiter, Q. Li , and S. Wetzel. Using voice to generate cryptographic keys: A position paper. In *Proceedings of Odyssey 2001, The Speaker Verification Workshop*, June 2001.

[21] F. Monrose, M. K. Reiter, Q. Li , and S. Wetzel. Cryptographic key generation from voice (extended abstract). In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, May 2001.

[22] F. Monrose, M. K. Reiter, Q. Li, D. Lopresti, and C. Shih. Towards speech-generated cryptographic keys on resource constrained devices (extended abstract). In *Proceedings of the 11th USENIX Security Symposium*, August 2002.

[23] A. Nagar, K. Nandakumar, and A. K. Jain, Biometric template transformation: a security analysis. In *Proc. SPIE, Electronic Image, Media Forensics and Security XII*, San Jose, CA, January 2010.

[24] M. Pandit and J. Kittler. Feature selection for a DTW-based speaker verification system. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pages 769-772. Seattle, WA, May 1998.

[25] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern analysis and machine intelligence*, 29(4): 561-572, April 2007.

[26] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. Digital Signal Processing 10(1): 19-41, 2000.

[27] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, Signal Processing*, ASSP-26(1): 43-49, February 1978.

[28] F. K. Soong, A. E. Rosenberg, B. Juang, and L. Rabiner. A vector quantization approach to speaker recognition. AT&T Technical Journal 65. pages 14-26. 1987.

[29] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. K. Vijaya Kumar. Biometric encryption^{TM} using image processing. In *Proc. SPIE, Optical Security and Counterfeit Deterrence Techniques II*, Vol. 3314, pages 178-188, San Jose, CA, 1998.

[30] A. B. J. Teoh and L. Chong, Secure speech template protection in speaker verification system. *Speech communication.* 52(2): 150-163, February 2010.

[31] U. Uludag, S. Pankanti, and A. K. Jain. Biometric cryptosystems: Issues and challenges. In *Proceedings of the IEEE*, Vol. 92, no. 6, pages 948-960, June 2004.

[32] R. H. Woo, A. Park, and T. J. Hazen. The MIT mobile device speaker verification corpus: data collection and preliminary experiments. In *Proceedings of Odssey, The Speaker and Language Recognition Workshop*, San Juan, Puerto Rico, June 2006.