Proceedings of the World Congress on Engineering 2013 Vol II,
WCE 2013, July 3 - 5, 2013, London, U.K.

Cryptographic Key Regeneration from Speech

K. Inthavisas and N. Sungprasert *

Abstract—**We propose a mapping algorithm using multi-thresholds that are determined by incorporation with pseudo-random bits. Hence, the algorithm can generate a binary string to appear to be random in the context of cryptography. Then, the binary string is used to protect a speech biometric template. We evaluate our scheme with two speech datasets. We compare our template with the other protected templates: time-domain and one-way function templates. The experimental results show that the error rate of our template is noticeably lower than the others. The randomness of the cryptographic key generated from our scheme is approximately 4 times better when compared to the global-threshold scheme.**

Keywords: cryptography, speech, biometrics, user authentication.

1 Introduction

To date, it is well known that biometric authentication systems are vulnerable to attack . In particular, the security of biometric templates is a topic of rapidly growing importance in the area of user authentication.

A typical biometric authentication system consists of two phases: enrollment and verification. During the enrollment phase, a user provides the system with biometric data, from which features are extracted and a template is created and stored. During the verification phase, users who claim to be authentic users would scan their biometric data again, and the same feature extraction algorithm is applied. The results are then compared with the stored template. If they are sufficiently similar, the matching algorithm accepts the user or rejects otherwise. The problem is that storing the biometric features directly as templates would not be secure because the reference or matching template can be inverted to the original signal. To address these problems, biometrics are used to combine or generate a cryptographic key to apply to a user authentication system.

The main problem of using speech biometric is duration. The duration of the same biometric provided by the same user at different times always varies with non-linear expansion and contraction. The solution is to use the Dynamic Time Warping (DTW) technique to set up a non-linear mapping of one signal to another by minimizing the distance between two signals [15]. To utilize DTW, we need a template as a keying signal to set up a warping function for incoming inputs. This process needs the template to define the distance, but the template may leak information to an attacker. For this reason, this information must be stored in a secure fashion, on a token such as a smart card with tamper resistant or by making a strong template that cannot be transformed to the original signal. Our design is focused on the strong template.

Basically, we can store time-domain features such as Energy and Zero Crossing Rate [6]. These features cannot be inverted to the original signal or transformed to frequency-domain features that are used to derive the cryptographic key. However, the robustness is decreased, because the speech signal is usually distorted by noise. For robustness, we have to use frequency-domain features as a template, but those features can be inverted to the original signal. In this paper, we propose the frequency-domain features as a stored template that cannot be inverted to the original template for a dynamic time warping (DTW) based user authentication system. The hardening algorithm, see Section 3.3, 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, defined in Section 2, will be used to create a stored or hardened template.

The other problem is the correlation among features. In [6], the author reported that "an iris code usually has a run length of 8 consecutive '1's or '0's." In other words, the binary seem to repeat the previous result. For speech, 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. We address this problem by proposing a mapping algorithm using multi-thresholds that are determined from pseudo-random bits. Hence, the algorithm can generate a binary string that an observer cannot predict.

In this work, we focus on how to reliably, securely, and randomly (in the context of cryptography) generate a bi-

*K. Inthavisas and N. Sungprasert are with the Department of Computer Engineering, Rajamangala University of Technology Srivijaya, Muang, Songkhla, 90000, Thailand e-mail: keerati.i@rmutsv.ac.th.

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

cryptographic key from speech. 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. Feature extraction is the first process to derive cepstrum and DFT features. This process involves speech processing detailed in Section 2. In Section 3, we describe the scheme to generate a cryptographic key from the extracted features. First, we describe the scheme to generate multi-thresholds, and then we describe the mapping algorithm. Lastly, we describe the hardening scheme. In Section 4, we describe the biometric key retrieval process. The experiments and results are provided in Section 5. Section 6 is conclusion.

2 Speech processing

A speech signal is usually represented by a function of time $s_a(t)$, in which t denotes time. The first step is the transformation of an analogue signal to digital. This process is called A/D conversion. The analogue signal is usually sampled at 8kHz. This means that only a frequency less than 4kHz will be reconstructed according to sampling theorem [3]. Hence, we use a low-pass digital filter with a cut-off at 4 kHz to strip the higher frequencies from the signal. If we denote the sampling period as P, the digital signal will be represented by $s(n) = s_a(nP)$, $n = 0, \ldots, N\text{-}1$. 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 α ranges between 0.9 to 1 [5]. 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 [5]. The last step is feature extraction where the frequency-domain features are extracted from the signal.

A basic feature of voice is the Discrete Fourier Transform (DFT). The Discrete Fourier Transform of N points signal $x(n)$ for $k = 0, \ldots, N\text{-}1$ can be defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp \frac{-j2\pi nk}{N} \tag{1}$$

The inverse transform for $n = 0, \ldots, N\text{-}1$ can be defined as

$$x(n) = \sum_{k=0}^{N-1} X(k) \exp \frac{-j2\pi nk}{N} \tag{2}$$

According to the real function property [13], if $x(n)$ is real and $x(n)$ and $X(k)$ are transform pairs, then

$$X(-k) = X(N-k) \tag{3}$$

This symmetric property, equation (3), can be exploited to decrease the computation required to transform a real sequence. To derive DFT, there is no need to compute X for $N/2 < k < N$, since these values can be found from the first half of X.

The most efficient feature to identify a speaker is known as cepstral coefficients or cepstrum [9]. Cepstrum physically represents the movement of articulators (the teeth, alveolar ride, hard palate, and velum) of speakers. Its use is popular because of low correlation. Hence, it is appropriate to apply it for a cryptographic purpose. Cepstrum can be defined as the Inverse Fourier Transform of log-energy of Fourier Transform of a signal $s(n)$ [5]. By definition $c(v) = F^{-1}\{log|F\{\mathrm{s(n)}\}|^2\}$ where F and F^{-1} denote Fourier and Inverse Fourier Transform.

3 Biometric key regeneration

Our design can be overviewed as two phases: training and verifying. The biometric key regeneration is in the training phase indicated in Fig. 1. Users provide their training pass-phrases that are repeated $l+1$ times to the system. Feature extraction is the first process to derive cepstrum and DFT features. This process involves digital signal processing detailed in Section 2. For the sampling rate of 8 kHz, we use 240 samples per frame that are shifted every 80 samples. Each frame contains 12 cepstrum and 121 DFT features. The system is initialized by using one of the training utterances as the keying signal stored as DFT features of m frames, called *DTW template*, then performs DTW to the rest of training utterances. We use cepstrum features derived from warped signals (utterances) in the mapping process. The cepstrum features of each utterance will be mapped to a binary string of length m called a *feature descriptor*. Lastly, l feature descriptors are used to define *distinguishing features*, features that the user can reliably generate. The binary string of distinguishing features derived from the training utterances is called *distinguishing descriptor*. The mapping and defining distinguishing descriptor procedure are detailed in Section 3.2.

We initialized the template by using a full set of DFT features as a DTW template. To the rest of this paper, we refer a full set of DFT features template to as a *full template*. Ideally, if the template is completely useful to derive the cryptographic key, all bits of distinguishing features derived from the template will correspond to all bits of the distinguishing descriptor. Practically, most of them will correspond if we use a full template. Hence, the template has to be perturbed which is what we call *hard-*

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

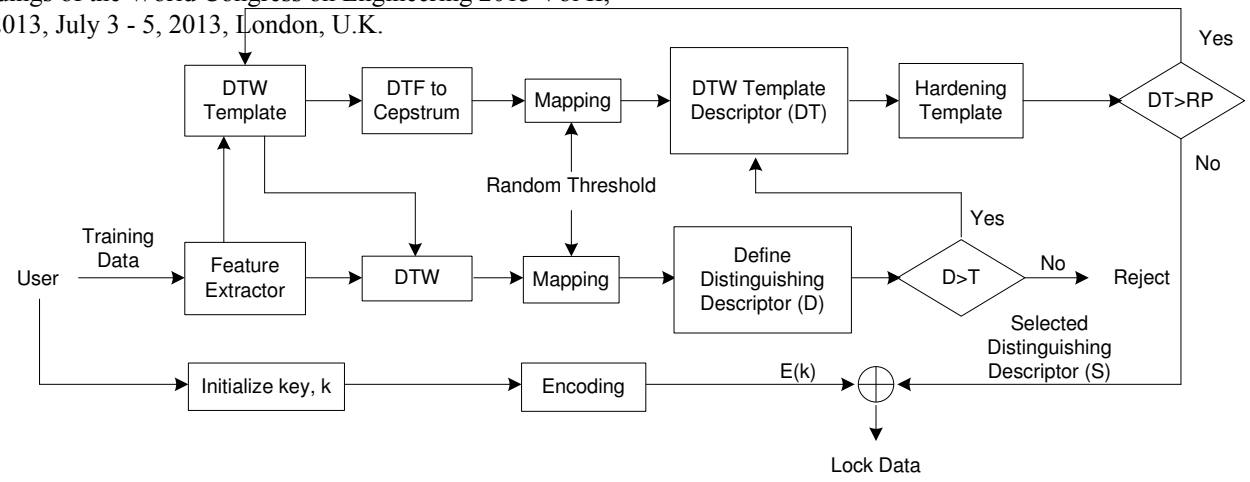

Figure 1: Biometric key regeneration in training phase.

ening the template. As the simplest attack is a random pass-phrase attack, we set the goal of hardening the template by the following statement: "The attacker utilizing a hardened template should not be better than a random pass-phrase attack where the attacker randomly select the other pass-phrases except the correct pass-phrase to generate the key."

Specifically, let the number of bit derived from a random pass-phrase and a DTW template that corresponds to the distinguishing descriptor be RP and DT; the system should yield DT as less than or equal to RP where RP is a fixed threshold. The RP threshold can be determined by experimentation that provides a full template as the keying signal and then defines a distinguishing feature. The number of bits of a random pass-phrase descriptor (a feature descriptor derived from a random pass-phrase) corresponding to the distinguishing descriptor on average is used as the RP threshold.

To guarantee that utilizing a frequency-domain feature as a hardened template is more robust than utilizing a time-domain feature template as we claim. The distinguishing descriptor D derived from our scheme should exceed some thresholds. This threshold can also be determined from the experimentation. At least, this threshold should be greater than a time-domain distinguishing descriptor TD derived by using a time-domain feature template. However, it cannot exceed a frequency-domain distinguishing descriptor FD derived by using a full template in frequency-domain. Hence, the threshold should lie between TD and FD. For now, let that the suitable threshold is T. If these conditions hold, $DT \leq RP$ and $D > T$, the template will not help the attackers as they just using a random pass-phrase attack is easier (better). For this reason, if DT is greater than RP, the template will be hardened as detailed in Section 3.3. In addition, the biometric will be rejected if D is less than or equal to T, because the system cannot find a suitable length of the distinguishing feature from the biometric. After each step in hardening the template, the hardened DTW template, or DTW template in Figure 1 will be the keying signal of the training pass-phrase and the process will be re-started until the conditions are met. Finally, the IDFT of the hardened DTW template is stored as a *hardened template* and 2^n-1 distinguishing descriptor, where $n = 3$, 4, ..., 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 [10]. The encoding code *E(k)* has to tolerate error within Hamming distance (H), a maximum number of bit differences between a distinguishing descriptor and a feature descriptor of a legitimate user. For the next step, the distinguishing descriptor 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 a feature descriptor S' that is sufficiently similar to a distinguishing descriptor within Hamming distance ($|S - S'| \leq H$) can unlock the $\mathcal{L}$ and correctly decode the key. We refer to the fuzzy commitment scheme [8] for more detail.

3.1 Multi-thresholds generation

We select a set of thresholds in such a way that the entropy of the biometric template is maximized. According to [7], 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. 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 algorithm in [1]. Next, a set of thresholds is selected based on the criteria that a query biometric will

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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 user j, let $\mu_j(i)$ and $\sigma_j(i)$ be the mean and standard deviation of the linear combination of the cepstrum features of i^{th} frame over l training utterances, the algorithm executes as follows:

1. Generate pseudo-random bits $p \in \{0,1\}^m$ using algorithm in [1].
2. Set the multi-thresholds $T_j(i) = \mu_j(i) + (-1^{p(i)})k\sigma_j(i)$ for some parameter $k > 0$
3. Securely delete pseudo-random bits

3.2 Mapping the biometric to a binary string

The following algorithm is used to map cepstrum features to a binary string and to define D, the distinguishing descriptor for user j with l training utterances.

1. Perform DTW to the training utterances.
2. For each frame of k^{th} training utterances, let $f_{j,k}(i)$ represented the cepstrum feature, where $i = 1, \ldots, m$ is the number of frame. Compute $f'_{j,k}(i)$ from the linear combination of $f_{j,k}(i)$.
3. Generate multi-thresholds $T_j(i)$, $i = 1, \ldots, m$ using the algorithm in Section 3.1.
4. Compute the i^{th} feature, $\phi_{j,k}(i) = f'_{j,k}(i) - T_j(i)$.
5. Binarize $\phi_{j,k}(i)$ to the feature descriptor, $b_{j,k}(i)$, by testing whether $\phi_{j,k}(i)$ is positive or negative. Map to '1' if it is positive and '0' otherwise.
6. For the training utterances, determine XORing of , $b_{j,k}(i)$, for $k = 1, \ldots, l$. If the XORing of $b_{j,k}(i)$ is zero, the i^{th} feature will be distinguishing feature and set $B_j(i) = b_{j,k}(i)$, otherwise $B_j(i) = \perp$.
7. Determine D, the number of bits that $B_j(i) \neq \perp$. If D is less than or equal to T, reject the biometric.

3.3 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 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 transformed to original template. The straightforward way is to enumerate over m frames of the original template then choose a set of optimal features that yield $DT \leq RP$, but the computational time is not possible. Hence, the optimal search algorithm should be employed. In [4, 14], there are good examples of a search algorithm for selecting the feature. 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. By using SBS, it is easy to terminate the program under the assumption we described earlier. We remove a DFT feature that maximizse DT each step until DT less than or equal to RP. Notice that in the algorithm we state that "remove the feature that minimizes DT", because when this feature is removed, DT derived from the rest of the features is minimized. The algorithm for hardening a template is described by the following steps:

1. Initialize by setting one of the training utterances as a DTW template, a set of DFT features.
2. Remove one of the features from the DTW template that minimizes DT by performing the algorithm in Section 3.2.
3. While $DT > RP$ go to step 2.
4. Terminate, the IDFT of the remaining features is stored as the hardened template.
5. Define 2^n-1 the least variation of the distinguishing features, where $n = 3, 4, \ldots$, to form binary string S.
6. Set the lock data, $\mathcal{L} = E(k) \bigoplus S$, where $E(k)$ is the encoded key and $\bigoplus$ denotes XOR operation.
7. Securely delete a set the training utterances.
8. Store $\mathcal{L}$, T_j, and the hardened template in the database.

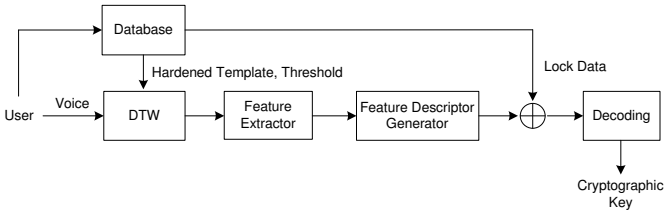

Figure 2: Biometric key retrieval in verification phase.

4 Biometric key retrieval

The biometric key retrieval process is in the verification phase indicated in Fig. 2. The user requests the template from the database that contains the hardened template, the multi-thresholds, and the lock data. Then the system

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

performs DTW to user's pass-phrase. The signal that resulted from DTW is executed using the algorithm in Section 3.2 to generate the feature descriptor, and the feature descriptor of the distinguishing feature will be XORed with the lock data. The next step is the decoding process. If the error is within Hamming distance, the key can be correctly reconstructed. To check whether the key is identical to the key generated in the training phase, we checked the hash function [12]. In the training phase, the initialized key, k, of user j was stored as $\mathrm{h}(k)$. Once the key k', is regenerated from the verification phase, the system checks to see whether $\mathrm{h}(k) = \mathrm{h}(k')$. If $\mathrm{h}(k) = \mathrm{h}(k')$, the key, k', is correct.

5 Experiments and results

We evaluate the performance using Equal Error Rate (EER) which is the rate at which a False Acceptance Rate (FAR) and a False Rejection Rate (FRR) are equal. The FAR is the percentage of the time that the system accepts the wrong speaker or one who is not authorized to access the system. In the same way, the FRR is the percentage of the time that the system rejects the authorized speaker. Two databases are used in experiments: The MIT mobile device speaker verification corpus (MDB) [16] and A data set in quiet environment (QDB). MDB is a public database available by MIT. QDB is our database collected over a month period.

5.1 Datasets

5.1.1 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 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.

5.1.2 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. In the third round, we selected the best imitator to mimic the target speaker's pass-phrases. Each pass-phrase was uttered five times.

5.2 Experimental setup

For MDB, we use a pass-phrase that is repeated four times in session I as the training pass-phrase. The same pass-phrase is used as a verification pass-phrase that is repeated four times in session II. To investigate the performance of the system, we use the same pass-phrase uttered by other speakers in session I to evaluate the imposter trial. The number of imposters that is available in the database varies from 1 to 6. In addition, we use six pass-phrases that are different from the verification pass-phrase to evaluate the random pass-phrase trial.

For 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 reduce 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. 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.

We set the length of binary string to 127 and 255 bits for MDB and QDB. Nevertheless, some pass-phrases cannot generate the binary string of that length. In this case, the algorithm should reject these pass-phrases. However, this experiment aims to compare the performance of various templates. The system performance is not a critical issue so that we use a zero padding scheme to adjust the lengths of binary string of these pass-phrases to those lengths. In our case, we use BCH code so that we can set the code word to 127 and 255 bits.

We compare the performance of the hardened template with the full, the time-domain, and the one-way function template. Note that, the full template is an insecure tem-

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 1: Equal Error Rate (EER) and Error Corrected by BCH Error Correction Code of the various templates with MDB and QDB. (*) in the second column indicates insecure template.

Database	Template	EER (%)	
		Random	Imposter
MDB	Full*	3.62	12.75
	Hardened	4.43	13.14
	Time-domain	6.22	15.59
	One-way	8.24	21.44
QDB	Full*	2.87	12.87
	Hardened	3.80	13.80
	Time-domain	5.60	15.20
	One-way	7.13	20.27

plate and the others are secure. The full template contains m frames of 121 DFT features. For the hardened template, 121 features of the full template are reduced to 9 and 11 on average for MDB and QDB. We use the time-domain energy as the time-domain template. For the one-way function template, the one-way function we use is a simple hash function that uses the vector of each frame (the full template) as the input and outputs the summation of the vector of each frame.

5.3 Experimental results

TABLE 1 shows the recognition performance of the full, the hardened, the time-domain, and the one-way function template with MDB and QDB. The results show that the EER of our scheme noticeably outperforms the time-domain and the one-way function template. These results are also illustrated in Fig. 3 for MDB. When comparing our scheme to the full template, the recognition performance of our scheme is slightly degraded. These results guarantee that the hardened template slightly degrades performance. However, the full template is an insecure template.

The quality of sound does not noticeably affect the recognition performance of the proposed template when compared with the other templates. This implies that the quality of sound is not critical to the performance of the system. Even if we test up to 30 pass-phrases in QDB, we cannot conclude now that the quality of sound is effective, as the number of speakers participating in QDB is too small. However, when we compare a small scale to a large scale database, the compared results of the various templates are similar. We will investigate this issue with a large scale database with high quality sound in the near future.

The EER of the imposter trial in MDB (13.14% in TABLE 1) seems high when compared to 10.97% of the work in [16]. Several things need to be explained. First, that work [16] used the speaker dependent, text dependent model where each phone model was created during enrollment phase. Even though it yields better performance, those models expose the speaker information to attackers. Second, the performance is not a primary concern in our work. However, we suggest the ways to promote the performance in the conclusion section. Lastly, our scheme can be used in a wide variety of applications, including file encryption, access to virtual private network, and user authentication.

5.4 Security of the template

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 template. These features cannot be transformed to the original template so that the security of a template protection scheme suggested in [11] is met. 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 [7].

We can estimate the security of the template using the sphere packing bound similar to [6]. 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 four recordings in session II for each speaker in MDB. To evaluate the security of the hardened template for the multi-thresholds scheme, two speakers are rejected because the binary strings are below 127 bits. We are left with a total of 46 speakers. We carry out 16,560 of inter-speaker comparisons to evaluate the uncertainty similar to [2]. For a binary string of 127 bits, the uncertainty of our template is 65 bits. From TABLE 1, the system should be able to correct the error up to 27 bits (imposter trial), that is approximately 21%. Here, z is 65 bits and w is 14 bits. The estimated entropy of the template is 21 bits. We further carry out 12,480 of inter-speaker comparisons using the global-threshold scheme. Eight speakers are rejected for the same reason. The estimated entropy of the template is 5 bits. We note that those speakers are rejected to reflect more precise entropy of the template. They are not rejected because of the hardening process. TABLE 2 summarizes the security of the various templates when we compare the multi-thresholds to the global-threshold scheme in MDB. It is clear that the entropy of the multi-thresholds scheme of the various templates is significantly improved.

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

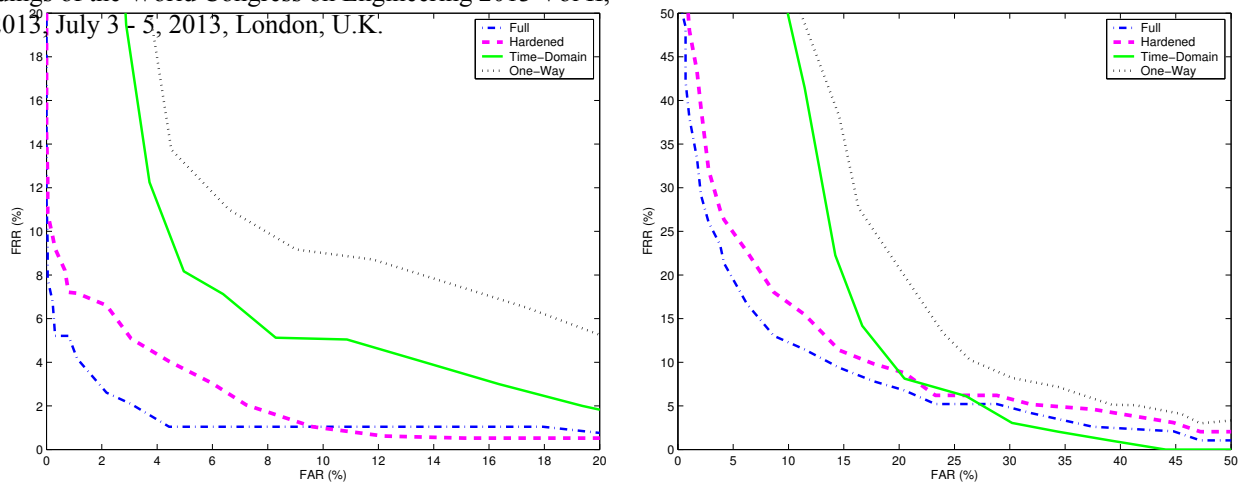

Figure 3: Comparison of the performance of the full, the hardened, the time-domain, and the one-way function template on MDB: (a) Random pass-phrase trial. (b) Imposter trial.

Table 2: The security of the various templates when we compare the multi-thresholds to the global-threshold scheme in MDB.

Template	No. Speakers		Entropy (bits)	
	Multi	Global	**Multi**	Global
Full	46	42	19	6
Hardened	46	40	**21**	**5**
Time-domain	46	22	21	6
One-way	46	17	23	6

6 Conclusions and Future Work

We address two problems in a cryptosystem. First, the problem of the feature correlation can be mitigated by using the proposed multi-thresholds. As a result, the randomness of the key (entropy) is increased from 5 to 21 bits. Second, we address the challenge in using DTW in a cryptosystem, more specifically, the template to create a warping function must not be able to transform to an original template, while the template should not degrades the performance. A solution, the hardened template is proposed. We compared our template with full, time-domain, one-way function templates. The full template yielded the best performance while ours had the second best results. However, the difference between the full template and ours was slight (0.39 and 0.93 for imposter trial in MDB and QDB). We noted that the full template is not secure and it leaves all the biometric information (a full set of DFT template) in the system.

There are a number of future works to be investigated. First, we need more experiments to identify the best feature that will offer the best performance as the EER of the system is still high. More specifically, we will explore features to use as a template. These features must be frequency-domain features due to our design. Next, we will explore features to generate the key. These features are frequency-domain and time-domain features. Alternatively, we may use a random key to promote the performance. Second, the security against potential attacks needs to be further explored. In particular, a generative attack is the most serious attack we concern. Lastly, we will explore the mentioned issues with larger numbers of users and higher-quality speech.

References

[1] 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, New York, 1983. Plenum Press.

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

[3] J.R. Deller, Jr., J. H. L. Hansen, and J. G. Proakis. *Discrete-Time Processing of Speech Signals*. Macmilland Pub. Co., New York, 1993.

[4] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall, Englewood Cliffs, NJ, 1982.

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

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

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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

[8] 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.

[9] T. Kinnunen. *Spectral Featrues for Automatic Text-Independent Speaker Recognition*. PhD thesis, Department of Computer Science, University of Joensuu, Findland December 2003.

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

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

[12] 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.

[13] T. W. Parsons. *Voice and Speech Processing*. McGraw-Hill, New York, 1987.

[14] 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.

[15] 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.

[16] 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.

ISBN: 978-988-19252-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)