Speech Cryptographic Key Regeneration based on Password

K. Inthavisas
Department of Computer Science
& Engineering, Lehigh University
kei206@lehigh.edu

D. Lopresti
Department of Computer Science
& Engineering, Lehigh University
dal9@lehigh.edu

Abstract

In this paper, we propose a way to combine a password with a speech biometric cryptosystem. We present two schemes to enhance verification performance in a biometric cryptosystem using password. Both can resist a password brute-force search if biometrics are not compromised. Even if the biometrics are compromised, attackers have to spend many more attempts in searching for cryptographic keys when we compare ours with a traditional password-based approach. In addition, the experimental results show that the verification performance is significantly improved.

1. Introduction

To date, it is well known that biometric systems are vulnerable to attack [15]. In particular, the security of a stored template is seriously concerning. To alleviate this problem, researchers proposed a biometric cryptosystem to secure the template. However, the verification performance is degraded. Moreover, the error rate is unacceptable, for example the work in [10]. Even though the authors showed that the verification performance was slightly degraded when it was compared with the unprotected template approach, its error rate was still high.

Thus far, one of the promising ways to authenticate users is to combine a biometric cryptosystem with the other factors: knowledge or token. Therefore, the performance is improved in the case that the biometric and the input factors are not compromised simultaneously.

We consider the knowledge-based approach to be another factor in improving a biometric cryptosystem because the users do not need to carry a token. However, we need to deal with weak passwords selected by users. For this issue, we will show that the proposed scheme offers better properties than a traditional password-based approach when the biometric is compromised.

In this paper, we present Speech Cryptographic Key Regeneration based on user's Passwords (SCKRP). The SCKRP is a cryptographic framework that binds a biometric template with a pseudo-random key to create a protected template. We propose two schemes to enhance verification performance in a biometric cryptosystem using password. The proposed schemes are: transformation and permutation. Both can resist password brute-force search if biometrics are not compromised. Even if the biometrics are compromised, the security meets the same level of the password approach. On the other hand, the security provided by the biometric cryptosystem is not affected even when the password is compromised. We utilize Dynamic Time Warping (DTW) in our scheme. A DTW-based biometric user authentication system needs a DTW template to set up a warping function for a query biometric. In addition, a matching template is required to examine similarity. We utilize a hardened template proposed in [10] to protect the DTW template. For the matching template, it is protected by cryptographic framework. Next, the hardened template and query biometrics will be transformed using a password. We then introduce a scheme in mapping behavioral biometric measurements (feature vector) to a binary string which can be combined with a pseudo-random key for a cryptographic purpose. These steps are detailed in Section 3.

We evaluate SCKRP verification performance using Equal Error Rate (EER) with a public database: The MIT mobile device speaker verification corpus [21] available from MIT. This dataset is detailed in Section 4.

We consider three different scenarios in evaluating the SCKRP: I) Genuine: When an adversary does not access genuine biometrics and passwords. II) Compromised passwords: When an adversary accesses genuine passwords. III) Compromised biometrics: When an adversary acquires genuine biometrics. Then, we compare the system with unprotected Dynamic Time Warping-based speaker authentication [8]. Next, we compare ours with the protected approach in [10]. These experiments are detailed in Section 5. Finally, the results and security analysis are illustrated in Section 6.

978-1-4577-1359-0/11/$26.00

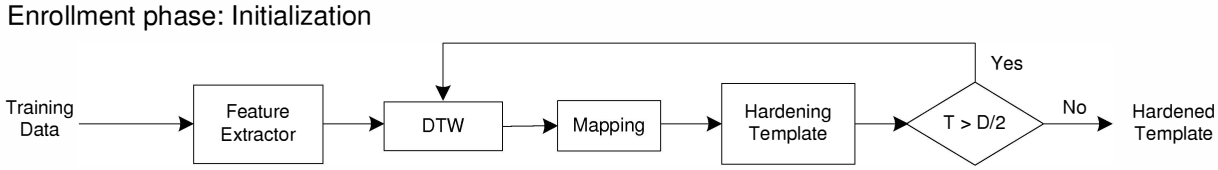

Figure 1. Enrollment phase: Initialization.

2. Related work

Numerous researchers proposed schemes to protect biometric templates by incorporation with random keys or passwords [19, 17, 18, 1, 15, 12]. These systems satisfy the criteria we raised in the previous section 1) the attackers cannot discriminate the correct password from incorrect when they use a brute-force search to find the key without the knowledge of biometrics. 2) when the password is compromised, it cannot be used to reveal the key.

Ballard et. al. [1] used a password to encrypt selected biometric features and some helper data to their key generation scheme. Their construction follows the approach similar to [2] where a low-entropy password is used to encrypt a high-entropy string. The features were specified as indexes into a table, and then a subset of the features was randomly assigned to each user. The feature indexes of this subset was encrypted in the template with a password using a cipher with arbitrary finite domain [3]. In this way, any passwords that were used to decrypt the template yield a subset of features indexes that falls within the global table. The authors ensured the indistinguishable from decryption with the correct and incorrect password by assigning any given feature with the same probability across the population to a user. Therefore, in both cases, a decrypted template appears as a random permutation on a subset of feature indexes. They demonstrated that their scheme did better than the previous approaches against some attacks even when the password was compromised.

Nandakumar et. al. [15] proposed a scheme to secure a fingerprint with a password. The password was used to select a transformation function to secure the fingerprint template. The transformed template was then secured using fuzzy vault framework. Finally, they used a key derived from a password to encrypt the vault. By using their scheme, the attackers are required to know the correct password before they can guess the key. Even if the correct password is selected, the security of the scheme is still at the same level as before using a password. Benefits of their scheme include template revocability, prevention of cross-matching, enhanced security and a reduction in the False Accept Rate. However, their scheme noticeably affects the False Reject Rate.

By utilizing the idea from Hao et al [9]., Kanade et al. proposed a three factors scheme (biometric, smartcard, and password) to apply to iris codes where a password was used to permute the key [12]. They could generate the key of 198 bits (compare to 140 bits in [9]) with estimated entropy of 83 bits (compare to 44 bits in [9]). Unfortunately, their scheme creates a security loophole which allows the attacker to crack the helper data without any additional information [20].

Teoh and Chong [17] proposed secure speech template protection in speaker verification system. The speech template was hidden through the random subspace projection process. In this process, a speech feature matrix is integrated with a user-specific key to obtain a random-projected matrix which cannot be inverted to the original speech feature matrix. The random-projected matrix is used to form a speaker probabilistic model and a decision threshold. They showed that the verification performance was very high. However, it would make some attacks, such as hill-climbing easier, as the system left the decision threshold and random-projected vectors matching process. In the case that the token is stolen, the attacker may make small changes in the input imposter's feature matrix and check to see how the match score changes. After a number of iterations, the attacker may be able to acquire a feature matrix that is close to the original.

To address problems mentioned above, we propose schemes to combine a speech biometric cryptosystem with a password. We first transform the biometric using a password. Then, the transformed version is mapped to a binary string. In this way, the transformation process forces the attackers to run dynamic programming every time they try another different password. Next, the biometric information is permuted using a password in such a way that the attackers cannot discriminate the correct password from brute-force search even when the biometrics are compromised. Lastly, a cryptographic key and the biometric information are hidden using a fuzzy commitment framework to protect the matching template.

3. Speech Cryptographic Key Regeneration based on Password (SCKRP)

The SCKRP can be overviewed as two phases: Enrollment and Verification. The biometric key regeneration is in the enrollment phase that comprises of two stages: Initialization and Regeneration. The first stage is used to protect

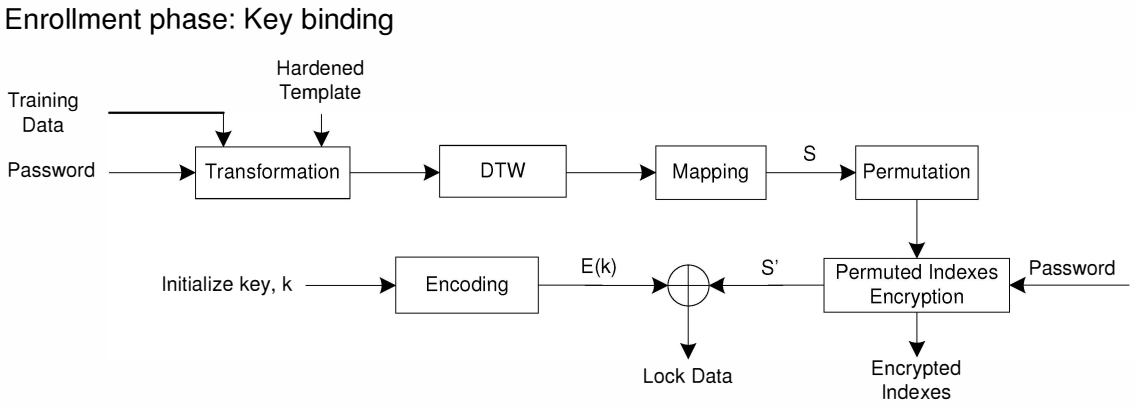

Figure 2. Enrollment phase: Regeneration.

the DTW template through a hardening process illustrated in Figure 1. The second stage illustrated in Figure 2 is used to protect the matching template; we apply a password in the second stage.

3.1. Enrollment: Initialization

For the Initialization stage (Figure 1), we follow the scheme in [10] detailed in the following. 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. 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 set to 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 reference signal which is stored as 121 DFT features of m frames, called the *DTW template*. Then, the system performs Dynamic Time Warping (DTW) [16] to the rest of training utterances to minimize distance between the reference signal and training utterance. The feature vectors of each training utterance will be mapped to a binary string of length m, a frame per bit, called a *feature descriptor*. For mapping process, we first generate pseudo-random bits $p \in \{0,1\}^m$. Next, a set of thresholds (multi-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. Lastly, l 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 a *distinguishing descriptor*.

For a hardening process, we initialized the template by using a full set of DFT features as an initialized DTW template. However, we are not able to use the full template as attackers can utilize it in gaining access to the system. Hence, the template has to be perturbed which is called *hardening the template*. 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 $D/2$. In [10], the authors showed that, under this condition, the hardened template was secure even if the attacker acquired the templates and had perfect knowledge of correlation of features. For this reason, if T is greater than $D/2$, one of template's feature vectors will be removed. After each step in hardening the template, the hardened DTW template will be the keying signal of the training pass-phrase and the process will be re-started until the condition is met. Finally, the result is stored as a *hardened template*.

3.2. Enrollment: Regeneration

This stage (Figure 2) consists of three main steps: transformation, permutation, and key binding. Firstly, random numbers derived from the user's password are used to transform the hardened template and training pass-phrases. The transformed biometrics are then mapped to binary strings. Then, a distinguishing descriptor (a binary string that users can reliably generate) is defined. Secondly, the distinguishing descriptor is encrypted with a password. Finally, the encrypted binary string is used to secure the cryptographic key using fuzzy commitment framework [11]. These steps

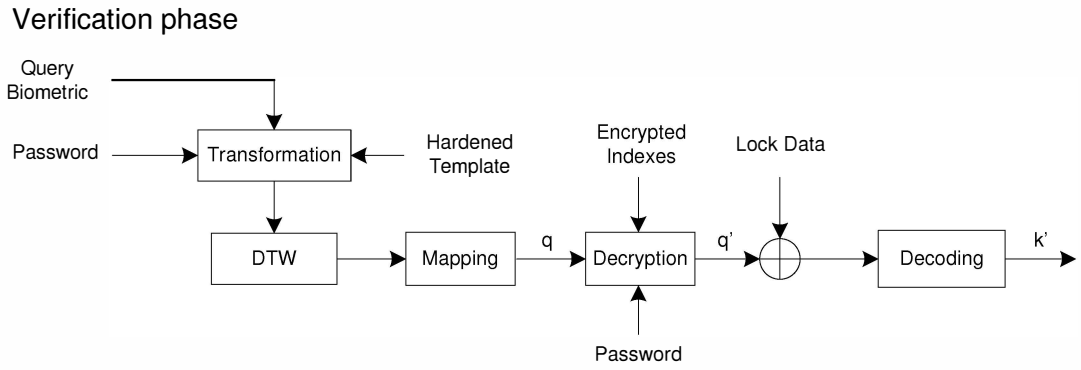

Figure 3. Biometric key retrieval in verification phase.

are detailed in the following.

3.2.1 Transformation

The system first generates two sets of pseudo-random numbers, $S = \{-1, 1\}^{2m}$ where m (the same m in the initialization stage) is the number of frames of the hardened template. Hence, if a training biometric is greater than $2m$ frames, the users will be asked to re-utter their pass-phrases.

Two sets, say S_1 and S_2, are arranged in a two-column table: S_1 in the first column and S_2 in the second column. Then the system uses an eighth-character password to generate a $2m$ bit binary string $R = \{r_i \in \{0, 1\}, i = 1, \ldots, 2m\}$. Next, the R will be used to select the random numbers in the table: select the number in the first column if $r_i = 1$ and otherwise in the second column. Lastly, the selected random numbers will be used to transform feature vectors. More precisely, let $\mathcal{H} = \{f_i, i = 1, \ldots, m\}$ be a set of feature vectors of the hardened template where the vector $f_i = [f_i(1), \ldots, f_i(121)]^T$. The transformed version of $\mathcal{H}$ can be represented by $\mathcal{T} = \{(-1)^{r_i} \cdot f_i + f_i, i = 1, \ldots, m\}$. For a set of training vectors $X = \{x_i, i = 1, \ldots, n \leq 2m\}$ where the vector $x_i = [x_i(1), \ldots, x_i(121)]^T$, the transformed version is $\mathcal{Q} = \{(-1)^{r_i} \cdot x_i + x_i, i = 1, \ldots, n\}$. Using $\mathcal{T}$ as a reference template, the system performs DTW to the $\mathcal{Q}$. The result will be used in mapping and generating distinguishing descriptor the same way as described earlier in the initialization stage. Next, we select 2^n-1 bits, where $n = 3, 4, \ldots$, based on feature variation to form a binary string $S = \{b_i \in \{0, 1\}, i = 1, \ldots, L = 2^n - 1\}$.

3.2.2 Permutation

For the second step, the indexes of the S will be randomly permuted in the context of cryptography. Next, the permuted indexes are used to arrange the binary string S and we refer the result to as an arranged binary string S'. Finally, we employ a prefix cipher [3] with domain and range in $[1, L]$ where $[1, L]$ denote a set of integers from 1 to L in encryption and decryption the permuted indexes with a password $P \in \mathcal{K}$.

The prefix cipher consists of two functions: $E : \mathcal{K} \times [1, L] \rightarrow [1, L]$ and $D : \mathcal{K} \times [1, L] \rightarrow [1, L]$. Therefore, if we refer the encrypted permuted indexes to as $\mathcal{M}$, every possible password when it is used to decrypt $\mathcal{M}$, will yield an integer string that consists of non-repeated random integers in $[1, L]$. By utilizing this scheme, the attackers cannot discriminate the correct password from brute-force search as the decrypted template appears as a random permutation on a subset of the indexes.

3.2.3 Key binding

To combine the arranged binary string S' with cryptographic key is the last step. The system first generate a pseudo-random bit k and then encoded properly denoted by $E(k)$ of length L (see Figure 2). In our case, we use BCH code [13]. 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.

3.3. Verification

The biometric key retrieval process is in the verification phase illustrated in Figure 3. The user requests the template from the database that contains the hardened template, the multi-thresholds, and the lock data. A user's password will be used to transform the hardened template and a query biometric the same way in Section 3.2.1. Once the transformed

versions are set, the system performs DTW. Then, the result will be mapped to a feature descriptor q. Next, the encrypted permutation indexes $\mathcal{M}$ will be decrypted with the password; the result is used to re-arrange the feature descriptor q and we refer the re-arranged result to as q'. Then, the q' 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, 9, 14] checked the hash function. In the training phase, the initialized key 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. The system authenticates the user.

4. The MIT mobile device speaker verification corpus

This database [21] was collected from 48 speakers (22 females and 26 males). The utterances were recorded in three acoustic environments: office, lobby, and street 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.

We use six recordings from the set of enrolled users to train the system and two recordings are used for verification. To investigate the performance of the system, we use the same pass-phrase uttered by other speakers (the set of dedicated imposters) to evaluate the imposter trial. The number of imposters that is available in the database varies from one to six.

5. Experimental setup

We compare the SCKRP with other speaker verification systems: Dynamic Time Warping (DTW) [8] and Dynamic Time Warping-based Biometric key Binding (DBKB) [10]. 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 the DBKB, 121 DFT elements of a full template are reduced to an average of nine. We set the length of the binary string to 511 bits. For our dataset, we can generate 139 bits on average for each feature; we need 4 features to generate 511 bits. For our setting, four features are the Short-Term Energy, the 13 order MFCC, the 12 order Linear Prediction Coefficient (LPC), and the DFT. We use *t*-error-correcting BCH [13] which denoted by *BCH(n, k, t)* where *n* is a block length, *k* is the key, and *t* is correctable bits. For this system, we employ *BCH(511, 229, 38)*.

For the SCKRP, the same parameters in the DBKB are utilized except the correctable bits t illustrated in Table 1 is varied to an operating point of each scenario. We will evaluate verification performance using Equal Error Rate (EER). However, the dataset does not include users' passwords. In our experiments, we have to select passwords which are likely to be used in the real world application. For this reason, we select eight character users' passwords based on difficulty levels [5]. Six classes of users' passwords and their distribution are: 1) one word (23%), 2) combination of two or more word (6%), 3) familiar numbers such as a social security number, street address, birth date, etc. (21%), 4) unfamiliar numbers (10%), 5) string of numbers and letter (34%), 6) string of numbers, letter, and symbols (6%).

6. Experimental results

We will first investigate in the case that one of the applied password schemes is excluded (one-layer scheme). Then, we will investigate the two-layer scheme SCKRP (transformation and permutation schemes).

6.1. One-layer scheme

Table 1 shows the EERs of the DTW [8], DBKB [10], and SCKRP. In the case of the permutation scheme only, the error rate of the compromised password scenario (II) does not differ from the DTW and DBKB system. In the other scenarios (I and III), the verification performance meets the same level of the password approach.

In the case of the transformation scheme only, the error rate of the compromised password scenario (II) also does not differ from other systems, but the error rates in the other scenarios (I and III) are noticeably degraded when we compare them with the previous case. As we introduced, the transformation layer is designed to slow down attackers who try to brute-force search the key. Therefore, it is necessary to keep this layer. In the next section, we will show that we can address this drawback when two schemes are combined.

6.2. Two-layer scheme

Table 2 shows the EERs of the two-layer scheme SCKRP. In the case that the password and biometrics are

Table 1. EERs of speaker verification systems against imposter attack for Dynamic Time Warping-based (DTW), Dynamic Time Warping-based Biometric key Binding (DBKB), and our approach (SCKRP) in the case that one of the applied password layer is excluded. Scenario I: genuine, Scenario II: compromised password, and Scenario III: compromised biometric

Method	Scenario	EER (%)	Error Corrected
DTW[8]	I	11.18	-
$DBKB$[10]	I	11.96	38 bits
$Permuted$ $SCKRP$	I	**0.00**	53 bits
	II	11.06	38 bits
	III	**0.00**	53 bits
$Transformed$ $SCKRP$	I	3.96	46 bits
	II	11.64	38 bits
	III	9.89	38 bits

Table 2. EERs of two-layer SCKRP against imposter attack. Scenario I: genuine, Scenario II: compromised password, and Scenario III: compromised biometric

Method	Scenario	EER (%)	Error Corrected
$Two-layer$ $SCKRP$	I	**0.00**	53 bits
	II	11.11	39 bits
	III	**0.00**	53 bits

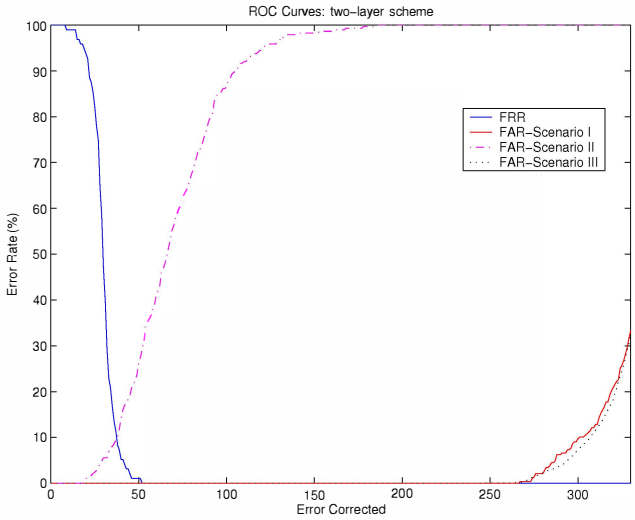

Figure 4. ROC curves of two-layer scheme. Scenario I: genuine, Scenario II: compromised password, and Scenario III: compromised biometric.

not compromised (scenario I), the verification performance of the SCKRP clearly outperforms the other systems. For the compromised biometric case (scenario III), the error rate is still the same as scenario I. For the compromised password case (scenario II), the verification performance of the SCKRP does not differ when we compare it with other systems. These results are also illustrated in Figure 4.

6.3. Security analysis

In this section, we investigate the security of two-layer SCKRP with three scenarios. For the case of the genuine (scenario I), we use the same approach presented by Inthavisas and Lopresti [10] to estimate the entropy. Hence, the security of the scheme can be estimated using the sphere packing bound $\mathcal{BF}=\frac{2^z}{\sum_{i=1}^{w}\binom{z}{i}}$ where z is the uncertainty of voice and w is the error bits that can be corrected by the system [9]. We carry out 4,512 of inter-speaker comparisons (the same dataset as used by Inthavisas and Lopresti) to evaluate the uncertainty. For a binary string of 511 bits, the uncertainty of our template is 125 bits. From Table 2, the system should be able to correct the error up to 39 bits, that is approximately 8%. Here, z is 125 bits and w is 10 bits. The estimated entropy is 76 bits, which is much better than Inthavisas and Lopresti approach (51 bits).

For the case of the compromised password (scenario II), the estimated entropy is 77 bits. However, 33.07%, which is determined using the analysis technique proposed in [10], leaks from the hardened template. Therefore, the estimated entropy is 51 bits, which is the same as reported in [10]. Even though 51 bits of entropy can easily be enumerated using todays computational resources, this space is determined under assumption that an attacker knows all users' passwords in the system. Therefore, it is very difficult for the attacker to get all.

For the case of the compromised biometric (scenario III), the estimated entropy is between 18-30 bits [4]. However, the attackers have to spend many more attempts for two reasons. First, the SCKRP is a biometric-based system; it prevents the attacker who is content to find the password of any users in the system (the weakest link). More precisely, the attackers randomly try the most probable password with every user in the system and try other passwords until they find the first match. For the SCKRP, they cannot do that as applying the same password to different biometrics yields different results. Second, the transformation process forces the attackers to run dynamic programming every time they try other different passwords. In contrast, if the transformation process is excluded, they can run dynamic programming only once. Then, they apply passwords to the warping signal and check to see whether the result matches the template. As a result, this case (the transformation layer is excluded) does not differ from the traditional password-based approach.

Overall processes create a greater computational load for an attacker. Even if this also makes users wait more time for authentication, it makes much more time for the attackers as they have to try every possible password.

7. Conclusions

We have proposed a way to combine a speech biometric cryptosystem with a password. The system consists of three layers. For the first layer, the biometric is transformed using a password. Then, we map the transformed version to a binary string. For the second layer, the result from the second layer is permuted using a password in such a way that the attackers cannot discriminate the correct password from brute-force search if the biometrics are not compromised. For the third layer, a cryptographic key and the binary string are hidden using a fuzzy commitment framework. The experimental results show that the verification performance of the system meets the same level of a traditional password-based approach if a biometric and password are not compromised simultaneously. Furthermore, the system increases the computational time for attackers to search for the key. Even if the attackers acquire the biometrics, they have been forced to align a query biometric each time they guess the password.

As this work was a preliminary investigation, the security against potential attacks needs to be further explored, in particular, a generative attack. In addition, we plan to investigate the impact of user passwords on the system.

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] S. M. Bellovin and M. Merritt. Encrypted Key Exchange: Password-Based Protocals Secure Against Dictionary Attacks. In *Proceedings of the 1992 IEEE Symposium on Security and Privacy*, pages 72-84, Washington, DC, USA, 1992.

[3] J. Black and P. Rogaway. Ciphers with Arbitrary Finite Domains. In *Proceedings of the The Cryptographer's Track at the RSA Conference on Topics in Cryptology*, pages 114-130. Springer-Verlag, 2002.

[4] W. E. Burr, D. F. Dodson, and W. T. Polk. Information Security: Electronic Authentication Guideline. *NIST Special Report 800-63*, April 2006.

[5] D. S. Carstens, P. R. McCauley-Bell, L. C. Malone, and R. F. DeMara. Evaluation of the human impact of password authentication practices on information security. *Informing science journal*, 7:67-85, 2004.

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

[10] K. Inthavisas and D. Lopresti. Speech Biometric Mapping for Key Binding Cryptosystem. In *SPIE, Biometric Technology for Human Identification VIII*, Orlando, FL, April 2011.

[11] 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.

[12] S. Kanade, D. Camara, E. Krichen, D. Petrovska-Delacretaz, and B. Dorrizzi. Three factor scheme for biometric-based cryptographic key regeneration using iris. In *biometrics Symposium*, pages 59-64, Tampa, FL, September, 2008.

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

[14] 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.

[15] K. Nandakumar, A. Nagar, and A. K. Jain. Hardening fingerprint-based fuzzy vault using password. In *proceedings of 2nd International Conference on Biometrics (ICB)*, pages 927 - 937, Seoul, South Korea, August 2007.

[16] 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.

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

[18] M. Savvides and B. V. K. Vijaya Kumar. Cancelable biometric filters for face recognition. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04)*, Vol. 3, pages 922-925, August 2004.

[19] O. T. Song, A. B. J. Teoh, and D. C. L. Ngo. Application-specific key release scheme from biometrics. *International Journal of Network Security*, vol. 6, no. 2, pages 127-133, March 2008.

[20] A. Stoianov. Security of error correcting code for biometric encryption. In *8th Annual International Conference on Privacy Security and Trust*, pages 231-235, Ottawa, Canada, August 2010.

[21] 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.