Phoneme Spotting for Speech-Based Crypto-key Generation

L. Paola García-Perera, Juan A. Nolazco-Flores, and Carlos Mex-Perera

Computer Science Department, ITESM, Campus Monterrey,
Av. Eugenio Garza Sada 2501 Sur, Col. Tecnológico,
Monterrey, N.L., México, C.P. 64849
{paola.garcia, jnolazco, carlosmex}@itesm.mx

Abstract. In this research we propose to use phoneme spotting to improve the results in the generation of a cryptographic key. Phoneme spotting selects the phonemes with highest accuracy in the user classification task. The key bits are constructed by using the Automatic Speech Recognition and Support Vector Machines. Firstly, a speech recogniser detects the phoneme limits in each speech utterance. Afterwards, the support vector machine performs a user classification and generates a key. By selecting the highest accuracy phonemes for a a set of 10, 20, 30 and 50 speakers randomly chosen from the YOHO database, it is possible to generate reliable cryptographic keys.

1 Introduction

The key generation based on biometrics is now acquiring more importance since it can solve the problems of traditional cryptosystems authentication. For instance, the automatic speech key generation can be applied for secure telephone calls, file storage, voice e-mail retrieval and digital right management. The necessity of having a key which can not be forgotten, and that can be kept secure is one of the main goals of today key generation. Current biometric authentication uses the intrinsic attributes of the users to provide solution to this security items [12].

For the purpose of this research, speech is the biometric used. It was chosen among the others because it has the flexibility that by changing the uttered sentence, the key automatically changes. Using the Automatic Speech Recognition (ASR) it is possible to obtain the starting and ending time of each phoneme given a utterance and a speech model. Afterwards, a feature adaptation is needed which can convert a set of vectors in a characteristic and final feature. Finally, a user classification task is performed by the Support Vector Machine (SVM).

Monrose *et. al* [6] showed a first method in which a partition plane for the feature vector space was suggested to generate binary biometric keys based on speech. However, a plane that can produce the same key is difficult to find due to the fact that infinite planes are possible. A more flexible way to produce a key - in which the exact control of the assignation of the key values is available - is always attractive. The main challenge of the general research is to find a

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 770–777, 2005.

suitable method to generate a cryptographic-speech-key that should repeatedly generate the same key every time a user produces the same utterance under certain conditions.

Therefore, the objective of this proposal is to improve the accuracy results in a cryptographic key generation task by using the phoneme spotting. In a similar way ASR uses word spotting to find key words, it is possible to use phoneme spotting [15]. In our case, it is used to make a selection of the highest phoneme accuracies. The phoneme spotting has the ability to locate a set of key phonemes (meaning the phonemes with the highest accuracy) during the training stage. However, selecting the phonemes with highest performance has the drawback that larger pass phrases are required. This issue is not a real problem since the system performs much better, and the pass phrases are not being memorised by the user (the system can give a random sentence that a user can utter).

The system architecture is depicted in Figure 1 and will be discussed in the following sections. The part under the dotted line shows the training phase that is performed offline. The upper part shows the online phase. In the training stage the *speech processing* and *recognition* techniques are used to obtain the model parameters and the starts and ends of the phonemes in each user utterance. Afterwards, using the model parameters and the phoneme segmentation, the feature generation is performed. Next, the *Support Vector Machine* (SVM) classifier and the phoneme selection produces its own new model according to a specific kernel and bit specifications. From all those models, the ones that give the highest results per phoneme are selected and compose the final SVM model. Finally, using the last SVM model the key is generated. The online stage is similar to the training, but a filtering of the unwanted phonemes is also included. This scheme will repeatedly produce the same key if a user utters the same pass phrase.

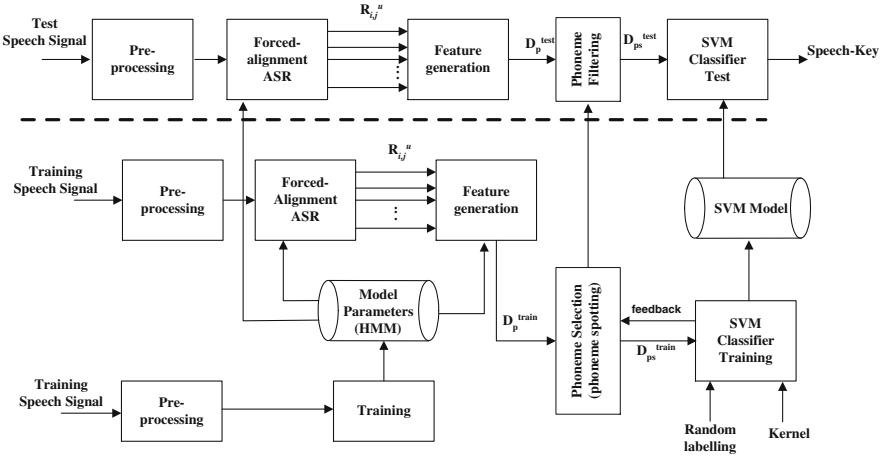

Fig. 1. System Architecture

2 Speech Processing and Phoneme Feature Generation

The ASR is one of the most important parts of our research. Firstly, the speech signal is divided into short windows and the *Mel frequency cepstral coefficients* (MFCC) are obtained. As a result an n-dimension vector, $(n-1)$-dimension MFCCs followed by one energy coefficient is formed. To emphasize the dynamic features of the speech in time, the time-derivative (Δ) and the time-acceleration (Δ^2) of each parameter are calculated [11].

Afterwards, a forced alignment configuration of an ASR is used to obtain a model and the starts and ends of the phonemes per utterance. For this research, the phonemes were selected instead of words since it is possible to generate larger keys with shorter length sentences.

In this training phase the system learns the patterns that represent the speech sound. Depending on the application the units can be words, phonemes, or syllables. The Hidden Markov Model (HMM) is the leading technique for acoustic modelling [10]. An HMM is characterised by the following, see Figure 2:

$A = \{a_{ij}\}, a_{ij} = Prob\{q_j \text{ at } t+1 | q_i \text{ at } t\}$ state transition probability distribution
$B = \{b_j(O_t)\}, b_j(O_t) =$ observation probability distribution
$\pi = \{\pi_i\} = Prob\{q_i \text{ at } t = 1\}$ initial state distribution
$O = \{O_1, O_2, ..., O_T\} =$ observation sequence (input sequence)
$T =$ length of observation sequence
$Q = \{q_1, q_2, ..., q_N\}$ hidden states in the model
$N =$ number of states

The compact notation $\lambda = (A, B, \pi)$ is used to represent an HMM [9]. The parameter set N, M, A, B, and π is calculated using the training data and it defines a probability measure $Prob(O|\lambda)$.

The resulting model has the inherent characteristics of real speech. The output distributions of the HMM are commonly represented by Gaussian Mixture Densities with means and covariances as important parameters, see Figure 3. Depending on the application one or more Gaussians can be included per state. But also, one or more states are also possible for a given reference sound. To determine the parameters of the model and reach convergence it is necessary to first make a guess of their value. Then, more accurate results can be found by optimising the likelihood function and using Baum-Welch re-estimation algorithm.

Assuming the phonemes are modelled with a three-state left-to-right HMM, and assuming the middle state is the most stable part of the phoneme representation, let,

$$C_i = \frac{1}{K} \sum_{l=1}^{K} W_l G_l, \tag{1}$$

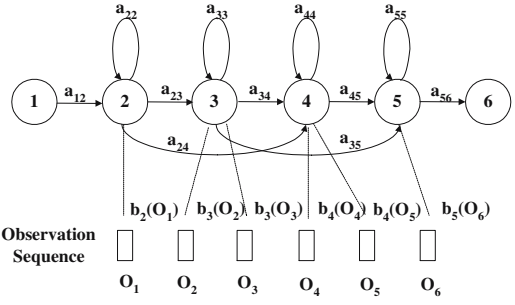

Fig. 2. Left-to-right HMM,*1...6* states, **a** transition probabilities, **b** output probabilities, **O** observation sequence

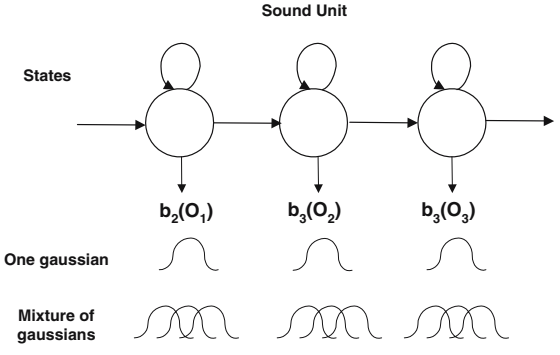

Fig. 3. HMM for a sound unit

where G is the mean of a Gaussian, K is the total number of Gaussians available in that state, W_l is the weight of the Gaussian and i is the index associated to each phoneme.

Given the phonemes' starts and ends, the MFCCs for each phoneme in the utterances can be arranged forming the sets $R^u_{i,j}$, where i is the index associated to each phoneme, j is the j-th user, and u is an index that starts in zero and increments every time the user utters the phoneme i.

Then, the feature vector is defined as

$$\psi^u_{i,j} = \mu(R^u_{i,j}) - C_i$$

where $\mu(R^u_{i,j})$ is the mean vector of the data in the MFCC set $R^u_{i,j}$, and $C_i \in \mathcal{C}_P$ is known as the matching phoneme mean vector of the model. Let us denote the set of vectors,

$$D_p = \{\psi^u_{p,j} \mid \forall\, u, j\}$$

where p is a specific phoneme.

Afterwards, this set is divided in subsets: D_p^{tr} and D_p^{test}. 80% of the total D_p are elements of D_p^{tr} and the remaining 20% form D_p^{test}. Then, $D_p^{train} = \{[\psi_{p,j}^u, b_{p,j}] \mid \forall\ u, j\}$ where $b_{p,j} \in \{-1, 1\}$ is the key bit or class assigned to the phoneme p of the j-th user.

3 Support Vector Machine

The classifier named *Support Vector Machine* (SVM) *Classifier* is a method used for pattern recognition, and was first developed by Vapnik and Chervonenkis [1,3]. Although SVM has been used for several applications, it has also been employed in biometrics [8,7]. For this technique, given the observation inputs and a function-based model, the goal of the basic SVM is to classify these inputs into one of two classes. Firstly, the following set of pairs are defined $\{x_i, y_i\}$; where $x_i \in \mathbb{R}^n$ are the training vectors and $y_i = \{-1, 1\}$ are the labels. The SVM learning algorithm finds an hyperplane (w, b) such that,

$$\min_{x_i, b, \xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

where ξ_i is a slack variable and C is a positive real constant known as a tradeoff parameter between error and margin.

To extend the linear method to a nonlinear technique, the input data is mapped into a higher dimensional space by function ϕ. However, exact specification of ϕ is not needed; instead, the expression known as kernel $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is defined. There are different types of kernels as the linear, polynomial, radial basis function (RBF) and sigmoid. In this research, we study just SVM techinque using radial basis function (RBF) kernel to transform a feature, based on a MFCC-vector, to a binary number (key bit) assigned randomly. The RBF kernel is denoted as $K(x_i, x_j) = e^{(-\gamma ||x_i - x_j||^2)}$, where $\gamma > 0$.

The methodology used to implement the SVM training is as follows. Firstly, the training set for each phoneme (D_p^{train}) is formed by assigning a one-bit random label ($b_{p,j}$) to each user. Since a random generator of the values (-1 or 1) is used, the assignation is different for each user. The advantage of this random assignation is that the key entropy grows significantly. Afterwards, by employing a grid search the parameters C and γ are tuned.

The SVM average classification accuracy is computed by the ratio

$$\eta = \frac{\alpha}{\beta}. \tag{2}$$

where α is the number of times that the classification output matches the correct phoneme class on the test data and β is the total number of phonemes to be classified.

By performing the statistics and choosing an appropiate group of phonemes that compute the highest results in the trainning stage, with output D_{ps}^{train}, a key with high performance can be obtained. Just this selection of phonemes will be able to generate the key in the test stage.

Finally a phoneme feature filtering is performed using D_p^{test}. The sets D_{ps}^{test} are computed according to the models obtained in the training phase. This research considers just binary classes and the final key could be obtained by concatenating the bits produced by each selected phoneme. For instance, if a user utters three phonemes: */F/*, */AO/*, and */R/*, and just */F/* and */R/* are selected the final final key is $K = \{f(D_{/F/}), f(D_{/R/})\}$. Thus, the output is formed by two bits.

4 Experimental Methodology and Results

For the purpose of this research the YOHO database was used to perform the experiments [2,4]. YOHO contains clean voice utterances of 138 speakers of different nationalities. It is a combination lock phrases (for instance, "Thirty-Two, Forty-One, Twenty-Five") with 4 enrollment sessions per subject and 24 phrases per enrollment session; 10 verification sessions per subject and 4 phrases per verification session. Given 18768 sentences, 13248 sentences were used for training and 5520 sentences for testing.

The ASR was implemented using the Hidden Markov Models Toolkit (HTK) by Cambridge University Engineering Department [5] configured as a forced-alignment automatic speech recogniser. The important results of the speech processing stage are the twenty sets of mean vectors of the mixture of Gaussians per phoneme given by the HMM and the phoneme segmentation of the utterances. The phonemes used are: /AH/, /AX/, /AY/, /EH/, /ER/, /EY/, /F/, /IH/, /IY/,/K/, /N/, /R/, /S/, /T/, /TH/, /UW/, /V/, /W/. Following

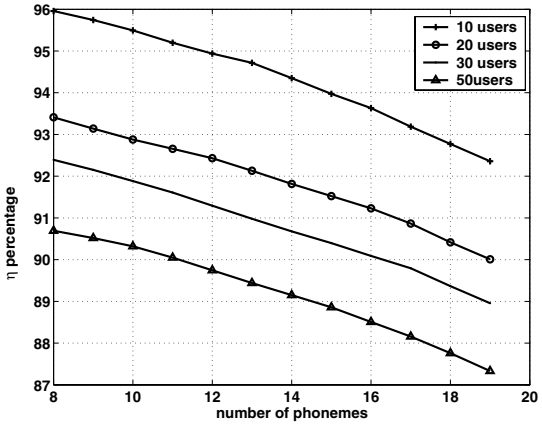

Fig. 4. HMM for a sound unit

the method already described, the D_p sets are formed. It is important to note that the cardinality of each D_p set can be different since the number of equal phoneme utterances can vary from user to user. Next, subsets D_p^{train} and D_p^{test} are constructed. For the training stage, the number of vectors picked per user and per phoneme for generating the model is the same. Each user has the same probability to produce the correct bit per phoneme. However, the number of testing vectors that each user provided can be different.

Following the method a key bit assignation is required. For the purpose of this research, the assignation is arbitrary. Thus, the keys have liberty of assignation, therefore the keys entropy can be easily maximised if they are given in a random fashion with a uniform probability distribution.

The classification of vectors D_{ps}^{train} and D_{ps}^{test} was performed using SVMlight [14]. The behaviour of the SVM is given in terms of Equation 2.

Using the principle of phoneme spotting, the phonemes with the highest accuracy and its SVM model are selected. The accuracy results η are computed for the selected phonemes. The statistics were computed as follows: 500 trials were performed for 10 and 20 users, and 1000 trails were performed for 30 and 50 users. Afterwards, the models that developed the lowest accuracy values are removed. The results for 10, 20, 30 50 users are depicted in Figure 4.

As shown, using phoneme spotting the results become better for all the cases. For instance, for 10 users the key accuracy goes from 92.3% to 95.9%. This is also the behaviour for the different number of users. The most complex experiment was performed using 50 users, but the result shows that 90% accuracy can be achieved.

If less phonemes are taken in account it is possible to compute keys with high accuracies. However, it has the drawback that when just a few phonemes are taken in account the utterances should be larger enough to have cryptographic validity. We have choosen to stop in 8 phonemes, so it is possible to have realiable combinations of phonemes to create the key.

5 Conclusion

We presented an scheme to improve the generation of a cryptographic key from speech signal. With this method we showed that an improvement is possible if just a selection of phonemes (phoneme spotting) is used in the training phase. Results for 10, 20, 30 and 50 speakers, from the YOHO database, were shown.

For future research, we plan to study the clustering of the phonemes to improve the classification task. It is also important to improve the SVM kernel or use artificial neural networks. Moreover, it is important to study the robustness of our system under noisy conditions. Besides, future studies on a M-ary key may be useful to increase the number of different keys available for each user given a fixed number of phonemes in the pass phrase.

Acknowledgments

The authors would like to acknowledge the Cátedra de Seguridad, ITESM, Campus Monterrey and the CONACyT project CONACyT-2002-C01-41372 who partially supported this work.

References

1. Boser, B., Guyon I. and Vapnik V.: A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, (1992)
2. Campbell, J. P., Jr.: Features and Measures for Speaker Recognition. Ph.D. Dissertation, Oklahoma State University, (1992)
3. Cortes, C., Vapnik V.: Support-vector network. Machine Learning 20, (1995) 273-297
4. Higgins, A., J. Porter J. and Bahler L.: YOHO Speaker Authentication Final Report. ITT Defense Communications Division (1989)
5. Young,S., P. Woodland HTK Hidden Markov Model Toolkit home page. http://htk.eng.cam.ac.uk/
6. Monrose F., Reiter M. K., Li Q., Wetzel S.. Cryptographic Key Generation From Voice. Proceedings of the IEEE Conference on Security and Privacy, Oakland, CA. (2001)
7. E. Osuna, Freund R., and Girosi F.: Support vector machines: Training and applications. Technical Report AIM-1602, MIT A.I. Lab. (1996)
8. E. Osuna, Freund R., and Girosi F.: Training Support Vector Machines: An Application to Face Recognition, in IEEE Conference on Computer Vision and Pattern Recognition, (1997) 130-136
9. Furui S. Digital Speech Processing, Synthesis, and Recognition. Mercel Dekker,inc. New York, 2001.
10. Rabiner L. R. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286, February 1989.
11. Rabiner L. R. and Juang B.-H.: Fundamentals of speech recognition. Prentice-Hall, New-Jersey (1993)
12. Uludag U., Pankanti S., Prabhakar S. and Jain A.K.: Biometric cryptosystems: issues and challenges, Proceedings of the IEEE , Volume: 92 , Issue: 6 (2004)
13. Lee K., Hon H., and Reddy R.: An overview of the SPHINX speech recognition system, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 38, No. 1, (1990) 35 - 45
14. Joachims T., SVMLight: Support Vector Machine, SVM-Light Support Vector Machine http://svmlight.joachims.org/, University of Dortmund, (1999)
15. Wilcox L., Smith I and Bush M, Wordspotting for voice editing and audio indexing, CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press, New York, NY, USA(1992), 655–656.