Cryptographic-Speech-key generation architecture improvements

Abstract. In this work we show a performance improvement of our system by taking into account the weights of the mixture of Gaussians of the Hidden Markov Model. Furthermore and independently tunning of each of the phoneme Support Vector Machine (SVM) parameters is performed. In our system the user utters a pass phrase and the phoneme waveform segments are found using the Automatic Speech Recognition Technology. Given the speech model and the phoneme information in the segments, a set of features are created to train an SVM that could generate a cryptographic key. Applying our method to a set of 10, 20, and 30 speakers from the YOHO database, the results show a good improvement compared with our last configuration, improving the robustness in the generation of the cryptographic key.

1 Introduction

The generation of a cryptographic key based on biometrics, i.e. voice, face, fingerprints [13], is nowadays acquiring great importance because of security issues. The advantage of having a cryptographic key based on biometrics is that it simultanously act as a password for access control and as a key for encryption of data that will be stored or transmited. Moreover, given the biometric information it is also possible to generate a private key and a public key. Since in biometrics the characteristics are unique for each individual, the key generated will be difficult to guess. For that reason, having a key generated by a biometric is highly desirable. From all the biometrics, voice was choosen in this research because a user can have the flexibility of changing a pass phrase when he requires it, or the system can also ask for a repetition of a random phrase, preventing unauthorised users access the system.

The results obtained in our previous work showed the potentiality of our system architecture [5–7]. Therefore, the purpose of this paper is to present the outcomes that improve our last results by considering the Gaussian weights in the interface between recognition and classification, and by performing a phoneme classification tunning. In this research the computer system consistently generates a cryptographic key based on the user's utterance and its matching pass phrase. In addition, a more flexible way to produce a key in which the exact control of the assignation of the key values is available.

The main challenge of this research is to find a method to produce a key with the characteristics already described. To achive good results we used speech processing and support vector machine techniques. Firstly, the speech signal is processed using an Automatic Speech Recogniser (ASR), from which a model and

a phoneme based segmentation is obtained. Next, a feature generator handles the ASR output data to obtain suitable sets for the Support Vector Machine (SVM). Finally, the SVM classifies the users and the key is obtained. A general view of the system architecture is shown in Figure 1 and will be discussed in the following sections.

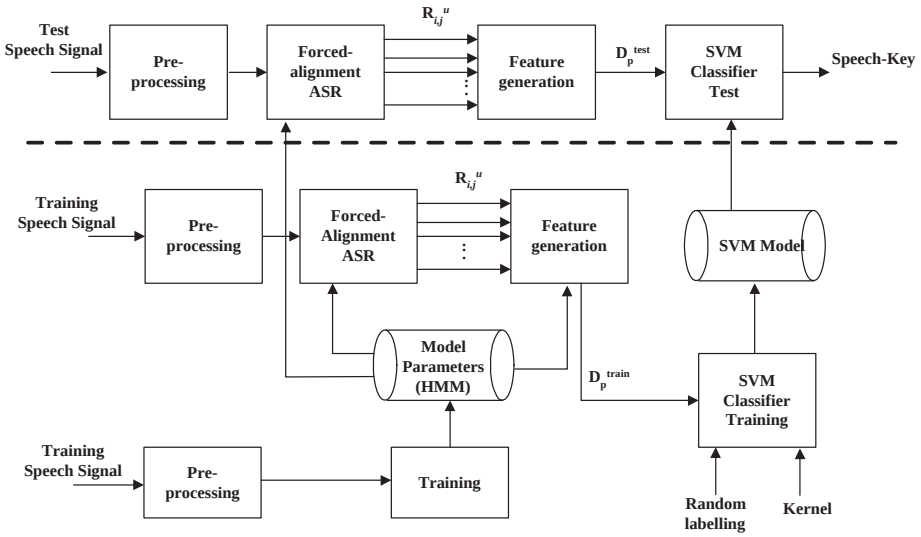

Fig. 1. System Architecture

2 Speech Processing

The primary task of this stage is to obtain the transcription and the starts and ends of the phonemes per user utterance. The speech signal is divided into short windows and the *Mel Frequency Cepstral Coefficients* (MFCC) are obtained. As a result a 13-dimension vector, 12-dimension MFCC 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 is calculated [11].

Afterwards, the ASR configured as a forced alignment recogniser provides a model and the starts and ends of the phonemes in a utterance. The ASR is based on a 3 state, left-right, Gaussian-based continuous Hidden Markov Model (HMM). Instead of words, the phonemes were selected because it is possible to generate larger keys with shorter length sentences. 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,

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

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.

3 Phoneme Feature Generation

Given the phonemes' segments, 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^{tr}_p and D^{test}_p. 80% of the total D_p are elements of D^{tr}_p and the remaining 20% form D^{test}_p. Then, $D^{train}_p = \{[\psi^u_{p,j}, b_{p,j}] \mid \forall\, u, j\}$ where $b_{p,j} \in \{-1, 1\}$ is the key bit or class assigned to the phoneme p of the j-th user.

4 Support Vector Machine

The Support Vector Machine is a particular instance of the kernel machines derived by Vapnik and Chervonenkis [1, 3]. Although SVM has been used for several applications, it has also been employed in biometrics [10, 9]. The basic task of this algorithm is to perform the classification of the input data into one of two classes. In this work, we explored the SVM using a radial basis function (RBF) kernel to classify sets of features. Those features are based on MFCC vectors and are to be transformed into sets of binary numbers (key bits) assigned randomly. The RBF kernel is denoted as

$$K(x_i, x_j) = e^{(-\gamma ||x_i - x_j||^2)},$$

where $\gamma > 0$. The SVM uses also a decision criteria, which depends on C, a tradeoff parameter between error and margin.

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 to optimise the results. In our previous research, we developed some results perfoming a suboptimal tunnning using just a pair of C and γ for all cases. However, in this new approach we use a suboptimal tunning for each phoneme; *i.e* each phoneme will have its own C and γ. Finally, a testing stage is performed using D_p^{test}.

This research considers just binary classes. The final key could be obtained by concatenating the bits produced by each phoneme. For instance, if a user utters two phonemes: */F/* and */AH/*, the final key is $K = \{f(D_{/F/}), f(D_{/AH/})\}$, thus, the output is formed by two bits.

5 Experimental Methodology and Results

The YOHO database [2, 4] was used to perform the experiments. 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. Next, the utterances are processed using the Hidden Markov Models Toolkit (HTK) by Cambridge University Engineering Department [8] configured as a forced-alignment automatic speech recogniser. The important results of the speech processing stage are the mean vectors of the phonemes C_i in Equation 1 given by the HMM and the phoneme starts and ends 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/. We have used 10, 20 and 30 users for our experiments and a mixture of 8 Gaussians to compare the cases.

The D_p sets are formed following the method described. 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 training, the number of vectors picked per user, 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. For this work, the key bit 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.

SVMLight by Thorsten Joachims was used to implement the classifier [12]. The behaviour of the SVM is given in terms of the average classification accuracy on test data for a given number of users. The average classification accuracy is computed by the ratio

$$\eta = \frac{\text{matches on test data for all phonemes and users}}{\text{total number of vectors in test data}}. \quad (2)$$

In this work we perfomed two experiments:

1. The goal of the first experiment was to evaluate the impact of the Gaussian weights. Therefore, we compared the performance of the system with and without considering the weights of the Gaussians. Table 1 shows the results of these experiments for a system with a mixture of 8 Gaussians, and for 10, 20, and 30 users.

number of users	% of η without weight	% of η using weights
10	92.32	92.51
20	89.9	89.99
30	88.79	88.8426

Table 1. Average % of η with and without Gaussian weights for different number of users

2. The purpose of our second experiment is to evaluate the advantage obtained by independently performing the tunning of the SVM parameters for each of the phonemes. Table 2 shows the results of this experiment for a mixture of 8 Gaussian and 10 users.

Phoneme	10user 8gauss weight	PHONE TUNNING
/AH/	92.8389	93.0936
/AO/	94.6542	94.8381
AX/	94.7563	95.3859
AY/	98.0601	98.2973
EH/	94.0936	95.238
ER/	96.376	96.416
EY/	88.9621	89.0155
F/	85.8751	85.9399
IH/	93.6509	93.6531
IY/	93.5343	94.2708
K/	86.146	87.126
N/	97.7116	97.9107
R/	88.2419	89.9046
S/	88.7694	89.3375
T/	91.5536	92.0274
TH/	86.4367	86.7832
UW/	95.5974	95.7973
V/	95.2885	95.4017
W/	91.6403	92.398

Table 2. % of η for different phonemes, using phoneme tunning and 10 users

6 Conclusion

In this research we proposed a method to eficiently generate a cryptographic key from voice. We used the techniques of the automatic speech recogniser and the support vector machines to achieve this purpose.

From the results we have found that the method to distinguish phonemes of specific users is quite good and provides good results for any key and user. The increment of the number of Gaussians and the tunning by phoneme facilitate the classification and better results are obtained.

For further study some exploration on error correction algorithms should be considered. Besides, future studies on a M-ary key can be useful to increase the number of different keys available for each user given a fixed number of phonemes in the passphrase.

References

1. Boser, B., I. Guyon, and V. Vapnik. 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. and V. Vapnik. Support-vector network. Machine Learning 20, 273-297, 1995.
4. Higgins, A., J. Porter and L. Bahler. YOHO Speaker Authentication Final Report. ITT Defense Communications Division, 1989.
5. Garcia-Perera L. P., C. Mex-Perera and J. A. Nolazco-Flores. Multi-speaker voice cryptographic key generation. Accepted for publication in the 3rd ACS/IEEE International Conference on Computer Systems and Applications - January 2005
6. Garcia-Perera L. P., C. Mex-Perera and J. A. Nolazco-Flores. Cryptographic-speech-key generation using the SVM technique over the lp-cepstra speech space. INTERNATIONAL SUMMER SCHOOL "NEURAL NETS E. R. CAIANIELLO" IX COURSE as a TUTORIAL RESEARCH WORKSHOP on Nonlinear Speech Processing: Algorithms and Analysis. October 2004. L. Paola Garcia-Perera, Carlos Mex-Perera, and Juan A. Nolazco-Flores
7. Garcia-Perera L. P., C. Mex-Perera and J. A. Nolazco-Flores. SVM Applied to the Generation of Biometric Speech Key A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 637-644, 2004. Springer-Verlag Berlin Heidelberg 2004
8. Young,S., P. Woodland HTK Hidden Markov Model Toolkit home page. http://htk.eng.cam.ac.uk/
9. E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical Report AIM-1602, MIT A.I. Lab., 1996.
10. E. Osuna, R. Freund, and F. Girosi, Training Support Vector Machines: An Application to Face Recognition, in IEEE Conference on Computer Vision and Pattern Recognition, pp. 130-136, 1997.
11. L.R. Rabiner and B.-H. Juang. Fundamentals of speech recognition. Prentice-Hall, New-Jersey, 1993.
12. T. Joachims, SVMLight: Support Vector Machine, SVM-Light Support Vector Machine http://svmlight.joachims.org/, University of Dortmund, November 1999.
13. U. Uludag, S. Pankanti, S. Prabhakar and A.K. Jain, Biometric cryptosystems: issues and challenges, Proceedings of the IEEE , Volume: 92 , Issue: 6 , June 2004.

7 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.