Speaker/Speech Recognition for Login Authentication

CS110 Operating Systems

Project-2c

Guide: Prof. Poonacha P G
Prof. Shrisha Rao

Name	Roll Number	Contact	Email
Priyadharshini V	MT2012104	9663541040	priyadharshini.v@iiitb.org
Sarabjeet Singh	MT2012126	9535942516	sarabjeet.singh@iiitb.org
Sucheta Chatterjee	MT2012141	9535149615	sucheta.chatterjee@iiitb.org
Sumit Sharma	MT2012144	9686189202	sumit.sharma@iiitb.org
Yamini M	MT2012168	9739810659	yamini.m@iiitb.org

Team Leader: Sarabjeet Singh

Contents

1 Introduction

Security has become a major issue in today's world. With the advancement of technology, the dependence on machines for day-to-day activities has increased resulting in escalation of secure interactions with these systems. One of the proposed and widely used solution is biometric authentication. This method is known to provide better security than the traditional text-based login authentication because of its uniqueness to each individual.

Some of the biometric authentication techniques include finger print recognition, face and iris recognition etc. In this project we will be implementing the speaker/speech recognition which will do voice verification along with the speech verification for every individual to prevent malicious user from breaking into the system.

Currently, a large number of biometric authentication softwares are available for windows but not for linux systems. The drivers for processing biometric inputs for linux based systems were released on 2005, and the application level support was far from being ideal[5]. So there is a need to develop a robust biometric authentication system for linux that works under realistic conditions.

2 Problem Statement

2.1 Problem description

Speech/Speaker recognition systems allows the user to recite a pass-phrase and verifies it against a list of stored voice recordings. One major drawback of this technique is that any malicious user can record the authenticated users voice and replay it, thus breaking the systems security. Apart from speech identification, the system has to do speaker verification for rejecting non-registered users.

Another important drawback in these systems is the effect of noise on the input speech. Noise increases the error rate of speaker identification process thus leading to erroneous authentication. A system that authenticates using voice as its biometric input should address these issues.

2.2 Gap Analysis

In [1] a prototype is created to resolve the variation of speaker by providing speech training to the system. The input datasets consisting of speech signal of different intensity are passed on to spectrogram analysis wherein the speech signal amplitude is extracted. The effect of noise is reduced by neutralizing the speech signal. Emotional effects of speaker affect the syllable and accent so the signal is normalized to make it processed by artificial neural networks. Applying artificial neural networks(ANN) speeds up the process. The ANN will recognize the speech signals based on the predefined acoustics parameters. Finally Hidden Markov Model is used to recognize the speaker.

In [2] the speech signal is treated graphically to extract the essential image features. These extracted signal image features are subjected to a preprocessing phase where a standard Pulse Code Modulation with a frequency of 22,050Hz is used for sampling. After sampling, the real signal is segmented. Burgs Model is then applied to these segments. Burgs Model is a LPC principle based frequency spectral estimator. The signal spectrum obtained is then given to analysis by Toeplitz Matrix. This matrix describes the signal image as a feature vector. These vectors are then classified by probabilistic and radial basis function neural network. The input feature vector is then compared to mean feature vectors of given class and is classified based on similarity.

2.3 Project Proposal

We propose a system, wherein a speech model is constructed by extracting features from the voice input using statistical methods. The speech/speaker recognition is done by matching the input feature pattern to the patterns stored in the database. The code books(patterns) are formed by clustering algorithms. Also, to improve security, we generate random pass-phrase each time a user logs in. This results in a two step process: Speech verification and Speaker identification.

3 System Architecture

Figure-1 depicts the overall system architecture and modules which will be executed.

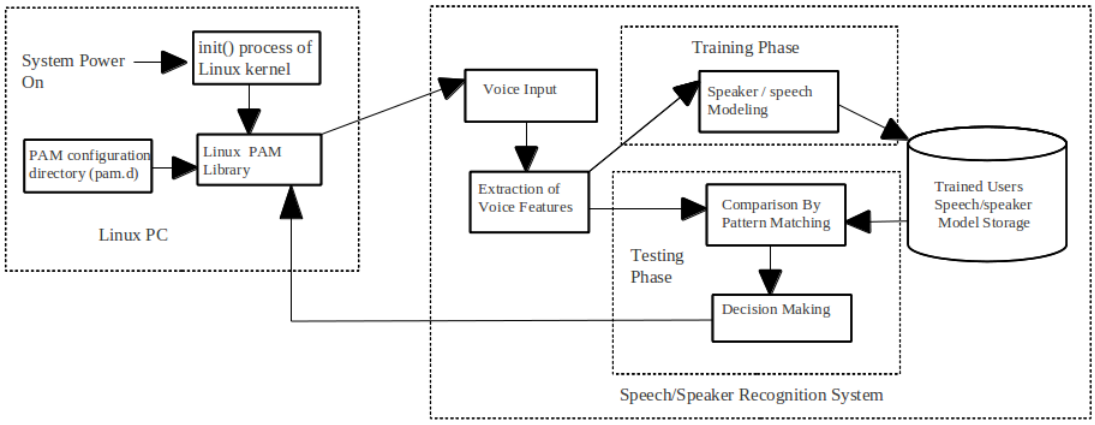

Figure 1: **Architecture Diagram**

The overall system architecture is as follows:

- In the above architecture diagram once the system starts up, the init() process of Linux Kernel calls the Linux PAM library which in turn will invoke the speech/speaker authentication application instead of normal username-password login
- PAM configuration file is required to setting up an authentication scheme for our application. The PAM config file is read when the application initializes the Linux PAM library
- After the system gets invoked, the user will be asked to read out the prompted random text
- Once the input voice has been captured, we will extract the voice features, i.e., formants, which will be used for the two level authentication:
    1. Speaker
    2. Speech

- The formants extracted from the input will be used to build speech/speaker recognition models using clustering algorithms
- These models will be stored in the database. This marks the end of the training phase
- In the testing phase, multiple voice inputs will be taken and formants for each of them will be obtained.
- After obtaining the formants, they will be compared with the stored formants by pattern matching.
- The stored formant with the best correlation value obtained by pattern matching will then be checked with the pre-defined threshold value required to authenticate the user.
- If the threshold is satisfied, the user will be authenticated to access the system, else the user will be asked to give the voice input again

The following subsections describe the system architecture in detail.

3.1 Linux PAM

We use Pluggable Authentication Module (PAM)[5] for integrating our proposed speaker/speech recognition system with Linux. PAM is a framework that sets up the environment the users will work in. And when the users log out, PAM will tear down the working environment in a controlled way. It introduces a layer of middle-ware between the application and the actual authentication mechanism.

PAM library provides four services:

- Authentication management
- Account management
- Session management
- Password management

The authentication management[1] supports different login techniques like fingerprint recognition, face recognition etc apart from the traditional username/password login. The /etc/pam.d, the PAM configuration directory, contains information about the necessary modules to be loaded for an application. So, for PAM to support our speech/speaker system at login stage we need to configure /etc/pam.d/login[2], the PAM configuration file.

3.2 Voice input

At the time of logging in, user will be provided with a random string that he/she will have to read. The string read by the user will be our voice input.

3.3 Extraction of Voice Features

Feature vectors are to be extracted from input speech. These vectors represents speech formants which provide the speech identity. These formants are the resonance frequencies of the vocal tract. This feature extraction can be done by several algorithms[4], some of which are:

- LPC - Linear Predictive Coding
- LPCC - Linear Prediction Cepstral Coding
- MFCC - Mel Frequency Cepstral Coefficients

In this project, we'll be using MFCC algorithm for the voice feature extraction and model formation. The steps involved in MFCC[3] are:

- **Pre-emphasis**

  Human beings generate speech which has random combinations of higher and lower frequencies. The mechanism of speech formation in humans causes the higher frequencies to get suppressed. In order to boost these frequencies pre-emphasis is done. This stage helps in recovery of higher formants.

[1] http://www.tuxradar.com/content/how-pam-works

[2] http://www.linux-mag.com/id/7887/

[3] http://mirlab.org/jang/books/audiosignalprocessing/speechFeatureMfcc.asp?title=12-2+MFCC

- **Framing**

  The speech signal is divided into several smaller frames which can be processed in a shorter duration. The frame size is generally taken as power of two for easier implementation of Fast Fourier Transform in the succeding steps.

- **Hamming Windowing**

  The previously generated frame is multiplied by hamming window to maintain speech continuity. Windowing has two effects on the signal:

  1. It smoothens the signal at the end points to prevent abrupt changes.

  2. It convolutes the speech spectrum and the fourier transform of the window.

- **Fast Fourier Transform(FFT)**

  Since speech signal corresponds to energy distribution over different frequencies, the processing should be done in the frequency domain. So this transformation converts time domain to frequency domain.

- **Filtering**

  The frequency response obtained from FFT is multiplied by set of band pass filters to create different energy spectrums. The filters are then spaced along Mel-Frequency.

- **Discrete Cosine Transform(DCT)**

  The DCT is applied on the energy spectrums obtained to generate Mel cpestral Co-efficients. The formula for DCT is:

$$C_m = \Sigma_{k=1}^{N} cos[m * (k - 0.5) * p/N] * Ek, \tag{1}$$

  where m=1,2, ..., L, N= number of filters , L= number of mel cepstral coefficients

3.4 Speech/Speaker Modeling

Speech/Speaker verification is one of the toughest challenges for the project considering the change in patterns of the same persons voice due to change

in the health conditions, speed at which sentence is delivered, noise in the background, and sometimes even change in the quality of mic and its distance from the mouth of speaker. For extracting the voice features again MFCC will only be used. Broadly, the following two steps will be carried out using MFCC:

- Extracting the voice from the background noise
- Extracting the pitch from the voice to recognize.

3.5 Training Phase

The speaker database is then formed by converting the raw input signal into a sequence of feature vectors. These feature vectors are clustered into a set of codewords. The set of such codewords is called codebook. The clusters are formed by a clustering algorithm. There are number of algorithms for codebook generation such as:

- K-means clustering algorithm
- Generalized Lloyd algorithm (GLA)
- Self Organizing Maps (SOM)
- Pair wise Nearest Neighbor (PNN).

For our implementation, we've planned to use K-means Algorithm. The K-means algorithm partitions the feature vectors into centroids. The algorithm first randomly chooses M cluster-centroids among the T feature vectors. Then each feature vector is assigned to the nearest centroid, and the new centroids are calculated for the new clusters. This procedure is continued until a stopping criterion is met.

3.6 Comparison By Pattern-Matching and Decision

These modules of the architecture constitues the testing phase of the system. In this phase, voice inputs from different users are gathered and formants are extracted from each of them. For each formant extracted, the distortion measure is computed by comparing it with all the stored formants using Euclidean Distance Technique. The stored formant with the least distortion

will be considered as the best match. If this least distortion measure satisfies the constraints of pre-defined threshold distortion level, then the user is authenticated to access the system.

4 Development Plan

Stage-1: Literature Study

- Studying the literature for Voice-capturing techniques, Noise removal and formants extraction techniques and PAM configuration for the integration of speech/speaker authentication system with Linux
- Preparation and finalising the Goal-Document
- Setting up repository for version control

Stage-2: Formants extraction

- Applying MFCC and normalizing the speech signal for the noise removal from the input speech signal
- Extraction of formants from the normalizied signal using MFCC algorithm

Stage-3: Building the Database

- Analysing and selecting suitable database for storing voice formants
- Collect various voice formants from the speaker and store it in database using K-means clustering algorithm

Stage-4: User Verification

- Sample formant from the speaker is compared with the collection of formants in database using the Euclidean Distance Technique
- Speech/speaker verification module is then tested and made ready for alpha-release

Stage-5: Setting up PAM

- PAM conguration le is congured to include the entries required for accessing speech/speaker authenticating application

- PAM library is set up to act as an interface between the application

Stage-6: Testing

- Test each individual module for its functionality with various test cases
- Check the integrated system providing voice input at different noise and emotional levels
- Preparation of first draft of final document and beta release of the project

Stage-7: Final Review and Submission

- Project Completion and final review alongwith submission of project papers and reports

5 Timeline

S.No.	Date	Task
1.	January 28	Submission of Final drafts of goal statement and setup version control repository. Completion of Stage-1.
2.	February 5	Choosing the appropriate method for capturing voice input and modeling formants using MFCC. First brief presentation of project architecture and plans by teams. Completion of Stage-2.
3.	February 25	Building the database from the formants generated by voice-inputs using K-means clustering. Completion of Stage-3.
4.	March 12	Implementing the Testing phase. Stage-4 completion and alpha release
5.	March 22	Integration with Linux by configuring PAM. Stage-5 complete.
6.	March 29	Testing the functionality of the fully integrated system and Beta-release. Completion of Stage-6.
7.	April 18	Project completion and report submission

References

[1] C. P. Lim, S. C. Woo, A. S. Loh, and R. Osman, "Speech recognition using artificial neural networks," *Web Information Systems Engineering, International Conference on*, vol. 1, p. 0419, 2000.

[2] E. Chandra and C. Sunitha, "A review on speech and speaker authentication system using voice signal feature selection and extraction," in *Advance Computing Conference, 2009. IACC 2009. IEEE International*. IEEE, 2009, pp. 1341–1346.

[3] A. Gandossi, W. Liu, and R. Tjahyadi, "A biometric approach to linux login access control," in *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on*, dec. 2006, pp. 1 –5.

[4] S. K.Gaikwad, B. W.Gawali, and P. Yannawar, "Article: A review on speech recognition technique," *International Journal of Computer Applications*, vol. 10, no. 3, pp. 16–24, November 2010.

[5] F. CVUT, "Integration of a biometric user authentication in unix-like systems josef hajas," 2007.

[6] S. Furui, "Vector-quantization-based speech recognition and speaker recognition techniques," in *Signals, Systems and Computers, 1991. 1991 Conference Record of the Twenty-Fifth Asilomar Conference on*, nov 1991, pp. 954 –958 vol.2.