Incorporating the Biometric Voice Technology into the E-Government Systems to Enhance the User Verification

KHALID T. AL-SARAYREH AND RAFA E. AL-QUTAISH
Faculty of Information Technology
Applied Science University
Amman 11931
JORDAN
khalid_sar@yahoo.com, rafa@rafa-elayyan.net

MOHAMMED D. AL-MAJALI
College of Educational Sciences
Mu'tah University
Karak 61710
JORDAN
mdmajali@mutah.edu.jo

Abstract:- Many courtiers around the world have started their e-government programs. E-government portals will be increasingly used by the citizens of many countries to access a set of services. Currently, the use of the e-government portals arises many challenges; one of these challenges is the security issues. E-government portals security is a very important characteristic in which it should be taken into account. In this paper, we have incorporated the biometric voice technology into the e-government portals in order to increase the security and enhance the user verification. In this way, the security should be increased since the user needs to use his voice along with his password. Therefore, no any unauthorized person can access the e-government portal even if he/she knows the required password.

Keywords:- Security Systems, E-Government Portals, Biometric Voice, Speaker Verification, Authentication.

1 Introduction

Security approaches can be broken down into two approaches: passive (authentication) and active (identification). Passive approaches are like a shield in that they protect against a clear and present danger such as a hacker attempting to access a computer system, while active approaches are more like prevention via a preemptive strike as in arresting terrorists before they plant a bomb.

All cards, keys, and username/password combinations have a common flaw: anybody can use them. Credit cards can also be easily counterfeited, and even the most sophisticated card can be lost, stolen, or maliciously taken away. The use of PINs and passwords somehow improves the situation, but the fundamental problem with PINs is that they identify a card but not its user. Obtaining both the card and the PIN might be more difficult than obtaining the card alone, but is quite feasible, particularly if the owner of the card is forced to cooperate. Thus, cards, PINs and passwords can hardly provide highly secure solutions. The same flaw applies to the username/password combination: the password really identifies the username, not the actual user! While all of the traditional approaches have their strengths, they also have corresponding weaknesses.

Whereas the requirement for physical access security has existed since time immemorial, computer threats and problems gained prominence during the 1990s due to the explosive growth of Internet, e-commerce and other computer technologies. Table 1 summarizes the computer threats as perceived in 1992 and 2002. The Table shows that the number and severity of threats to networked computers have caused into question traditional approaches to security and demand a new response from the IT to deal with the E-world of tomorrow.

From Table 1, we noted that typing a text password to open a system is not enough to identify the current user because:

1. Authorized persons give their own passwords to unauthorized ones.

2. Clever intruders may guise the password of this system if the given time is enough.
3. Smart programs with intelligence methods that can find the password in short time are available.
4. This password is used by a group of people.
5. Tracking and identifying the unauthorized persons who tried to inter the system.

Table 1: The Computer Threats as Perceived in 1992 and 2002 [1]

Most severe threats in 1992	Most severe threats in 2002	
- Natural Hazards	- Viruses	- Infringement of IP rights
- Inadequate Control over Media	- System penetration: Hacking/Espionage	- Spoofing
- Weak and Ineffective Controls Hacking	- Fictitious people/ Perpetrators	- Implied trust Exploitation
- Access to System by Competitors	- Denial of Service	- Active Wiretap
- Hacking	- Insider abuse of net access	- Sabotage
	- Unauthorized access by Insiders	- Telecom Eavesdropping
	- Natural Hazards	- Repudiation
	- Human Error	- Credit Card Fraud

Speaker verification is verifying a user's identity by his voice, which is assumed to be unique for that person. Even a very secure cryptology system has the chance of being cracked; a unique human voice is very attractive to be used in systems where security should be provided. In this paper, a text-dependent speaker verification system is implemented by using various feature extraction and feature comparison methods.

The feature extraction methods are mel-cepstrum, Linear Prediction (LP) coefficients and fundamental frequency and magnitude. The feature comparison methods are Vector Quantization (VQ) and Dynamic Time Warping (DTW).

Many courtiers around the world just have started their e-government programs. E-government portals will be increasingly used by the citizens of many countries to access a set of services. Currently, there are many challenges of the use of the e-government portals; one of these challenges is the security issues. E-government portals security is a very important characteristic in which it should be taken into account. In this paper, we have incorporated the biometric voice technology into the e-government portals in order to increase the security and enhance the user verification. In this way, the security should be increased since the user needs to use his voice along with his password. Therefore, no any unauthorized person can access the e-government portal even if he gets a password.

The rest of this paper is organized as follows: Section 2 introduces an overview of the previous work in this topic, Section 3 presents an overview of the proposed system, Section 4 describes the methods which we used in our system, Section 5 gives the details of implementation of the speaker verification, Section 6 illustrates some testing examples with their results, Section 7 discusses the results in the previous section. Finally, Section 8 concludes the paper and gives some trends on future works.

2 Literature Review

Voice is a combination of physiological and behavioral biometrics. The features of an individual's voice are based on the shape and size of the appendages (e.g., vocal tracts, mouth, nasal cavities, and lips) that are used in the synthesis of the sound. These physiological characteristics of human speech are invariant for an individual, but the behavioral part of the speech of a person changes over time due to age, medical conditions (such as a common cold), and emotional state, etc. Voice is also not very distinctive and may not be appropriate for large-scale identification.

Based on the spoken text, there are two types of the biometric voice recognition systems, that is, text dependent and text independent. A text dependent voice recognition system is based on the utterance of a fixed predetermined phrase. A text independent voice recognition system recognizes the speaker independent of what he/she speaks, as we used in our proposed system.

A text-independent system is more difficult to design than a text-dependent system but offers more protection against fraud. The problem with the voice recognition is that speech features are sensitive to a number of factors such as background noise, medical conditions, etc [2, 3]. We solved this problem by adding different voices in different situations (for the same person) to learn the system

on the different possibilities of variant voices for the same person.

Furthermore, for the text dependent, Monrose *et al.* [4] have introduced an algorithm to generate a cryptographic key from a user's utterance of a password. In addition, Jian and Ross [5] have stated that the NIST (National Institute of Standards and Technology) in 2002 implemented a text dependent biometric voice system and get 10-20% of false reject rate and 2-5% false accept rate.

3 The Proposed System: A General Overview

A speaker verification system consists of two parts: feature extraction and template based comparison. In the first part, the system records the user's speech and applies short-term analysis to find the parameters that distinguish the user. These parameters are then recorded as the reference template for any future comparison [7]. The next time, when the user needs to be verified, he speaks the same utterance. Short term analysis is again performed, and the parameters are extracted once more. In order to give access to the user, the new extracted parameters are compared with the previously recorded (reference) parameters. If the similarity between the two parameter sets is above a given threshold, the user is verified. Otherwise, access is denied. The block diagram of the system is shown in Figure 1.

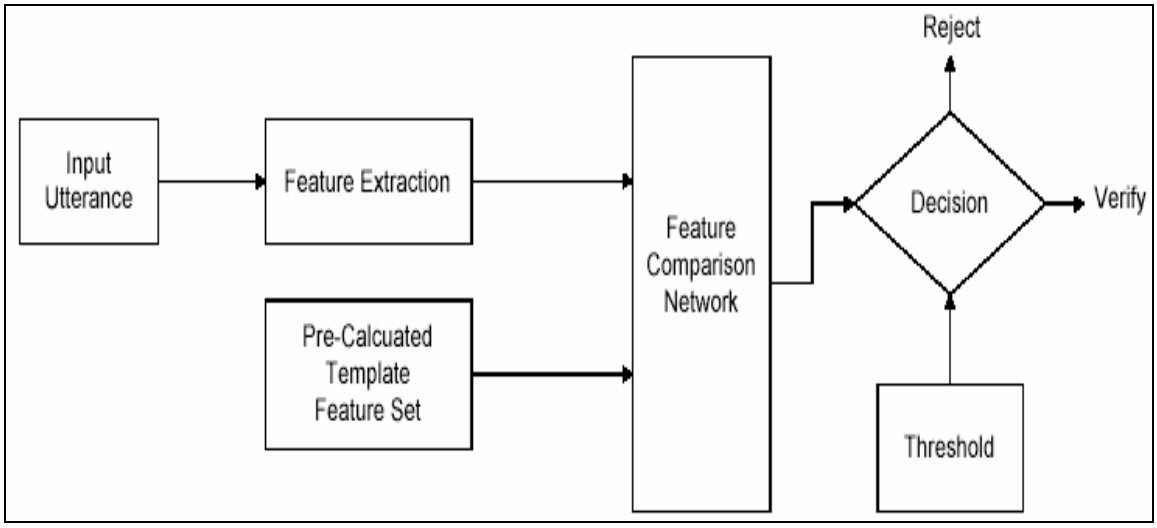

Figure 1: Block diagram of the speaker verification

4. Methods Used in the System

4.1 End Point Detection

The first technique that applies to each speech waveform is end detection. Due to the nature of the recording process, speech waveforms have blank parts in the beginning and in the end. These parts are purely noise [6], and they do not contain any information. As a result, the speech parameters of these parts are independent of the Speaker and identical in terms of statistics. Because of this fact, these parts increase the correlation between different users. This will cause the false acceptance ratio to increase. Moreover, it is easier to overlap identical parts of speech when the actual speech data starts at time t0. Since the duration of the speech is reduced, by using this method, the computational load is also reduced. Endpoint detection algorithms generally use the combination of zero-crossing rate and energy OR the combination of zero-crossing rate and average magnitude of a given utterance. Steps of the end-point detection algorithm are listed as follows:

- Removing the DC part (mean) of the speech signal, which is considered to be an important step because the zero-crossing rate of the signal is calculated and it plays a role in determining where the unvoiced sections of speech exist. If the DC offset is not removed, we will be unable to find the zero-crossing rate of noise in order to eliminate it from our signal.
- Framing speech signal.
- Computing the total zero-crossings number of each frame such that:

$$Z(k) = \sum_{m=1}^{N} \left| \mathrm{sgn}(x_k(m)) - \mathrm{sgn}(x_k(m-1)) \right| \quad (1)$$

Where Z(k) is the average magnitude of the kth frame and sgn(x) is the signum function that finds the sign of x.

- Compute noise statistics (mean & standard deviation) and the maximum of average Magnitude and zero-crossings of the first five frames assuming that the first five frames are inconsiderable parts.
- Find end-points according to the average magnitudes assuming that the average magnitude of the spoken parts is greater than a threshold TA (sum of the mean and standard deviation in our paper) determined by the mean and the standard deviation of the average magnitude of the noisy part. End-points should be in pair such that one of them is the beginning of a specific spoken part and the other one is the end of that part.
- Correct end-points according to the zero-crossings assuming that the zero crossings of spoken parts are smaller than a threshold TZ (sum of the mean and standard deviation in our paper) determined by the mean and the standard deviation of the average magnitude of the noise part. End point detection is applied to all of our speech signals before they are processed.

Figures 2 and 3 below show the plot of a two-speech signal with and without end-detection. In addition, Figure 4 shows the block diagram of the end point detection algorithm.

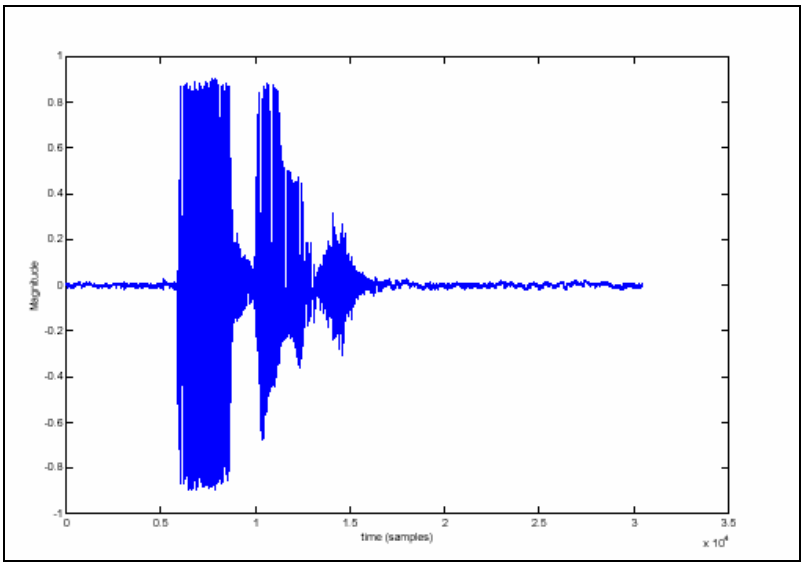

Figure 2: Speech signal before end detection

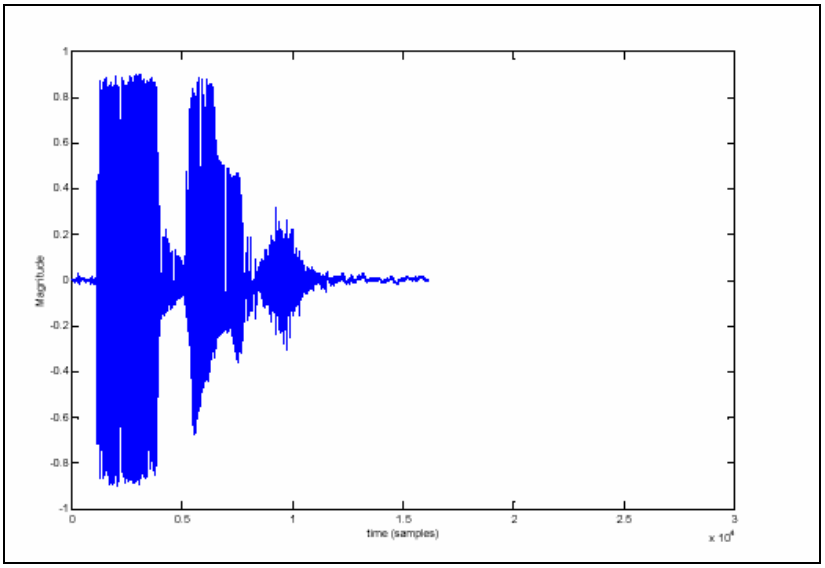

Figure 3: Speech signal after end detection

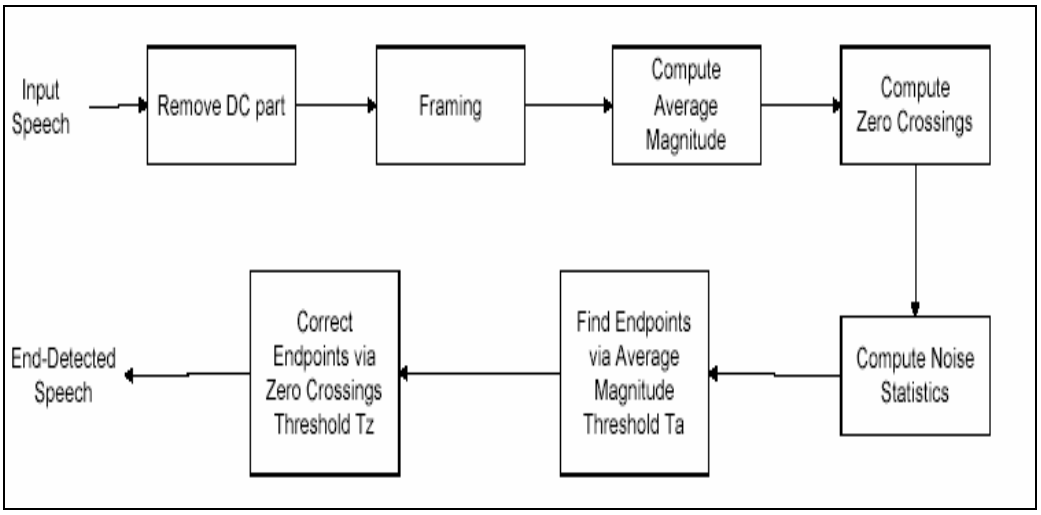

Figure 4: block diagram of end point detection algorithm

4.2 Feature Extraction

In order to compare different users, we need to extract the parameters from the speech signal. These parameters help us to distinguish one user from the others. However, the speech signal must first be divided into frames. The parameters are calculated separately for each frame. For the framing part, we choose to use a hamming window with a window size of 30msec and skip a rate (frame overlap) of 10msec. After each feature extraction algorithm, we get a matrix as a result. The columns of these matrices are the feature vectors for each frame [8].

4.2.1 Mel-Spectrum

Mel-spectrum coefficients are known to model the human ear well, and it is mentioned in the literature few times that the performance of the mel-cepstrum coefficients is better than the performance of other parameters. Mel-cepstrum coefficient extraction is based on the fact that the sensitivity of the human ear varies with frequency. Our ear is more sensitive to low frequency sounds than to high frequency ones. The computation of the extraction of these parameters will be explained next.

1. Apply framing to the end detected speech signal.
2. Then the 512-point fast Fourier transform of each frame is applied and the algorithm of the magnitude response is taken. Due to the symmetry property of the FFT, only the first 256 samples (Si) are processed.
3. The frequency range (0-fs/2 Hz) is divided into uniformly spaced triangular overlapping windows (K windows in total). Fig. 5 illustrates the Mel spectrum scale.

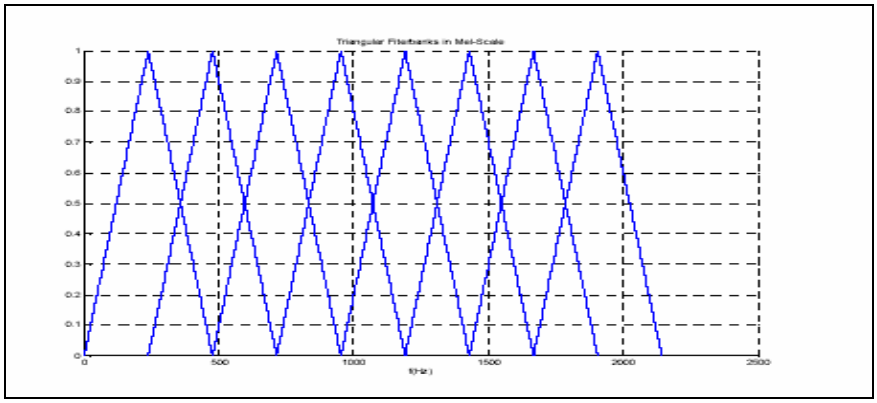

Figure 5: Mel spectrum scale

4. Then the Mel scale is converted back to the linear scale by the following equation:

$$f = 700 \times (10^{1/2595}) \quad (2)$$

In this case, the resulting mel-wrapped frequency filter bank triangles become linearly spaced in low frequencies and exponentially spaced at high frequencies as shown in Figure 6.

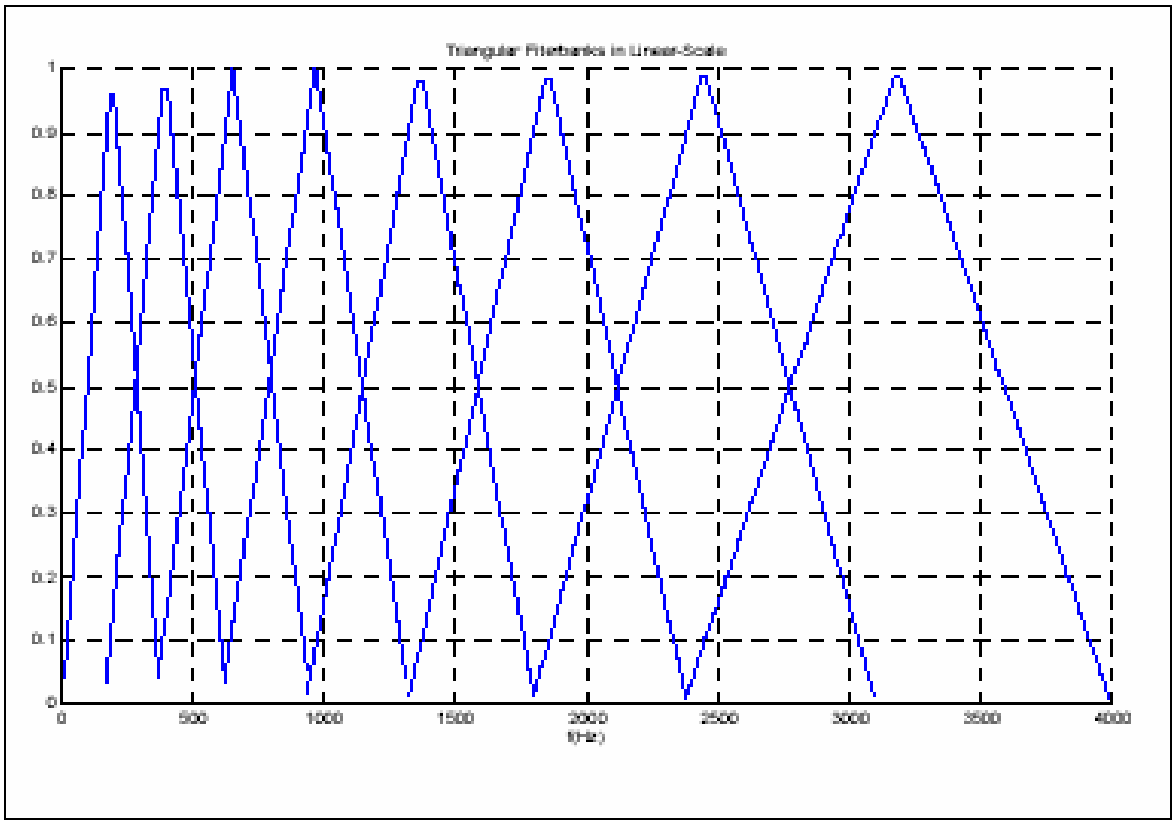

Figure 6: Linear Space

- Each triangular window (wk) is multiplied with the corresponding parts of the FFT, and the resultant window becomes (Sik *wk). Then for each window, a single number is found according to the following equation:

$$m_{ik} = \sum S_{ik} \times W_k \qquad (3)$$

- After applying the above procedure to K window, a vector Pi of length K is obtained such that

$$P_i = [m_{i1} \quad m_{i2} \quad \ldots \quad m_{ik}] \qquad (4)$$

- Finally, the Inverse Discrete Cosine Transform (IDCT) of Pi is calculated, and the first n many coefficients are taken. The obtained mel-cepstrum coefficient for the frame will be:

$$` \quad C_i = [c_{i1} \; c_{i2} \; \ldots \; c_{in}] = IDCT \qquad (5)$$

When this procedure is applied to I frames, the mel-cepstrum matrix is found as:

$$C = \begin{bmatrix} c_{11} & c_{21} & A & c_{I1} \\ c_{12} & c_{22} & A & c_{I2} \\ M & M & O & M \\ c_{1n} & c_{2n} & A & c_{In} \end{bmatrix} \qquad (6)$$

In our simulations, we have used 30 filter banks (K) and 12 Cepstrum coefficients (n).

The block diagram of the feature extraction part is given in Figure 7.

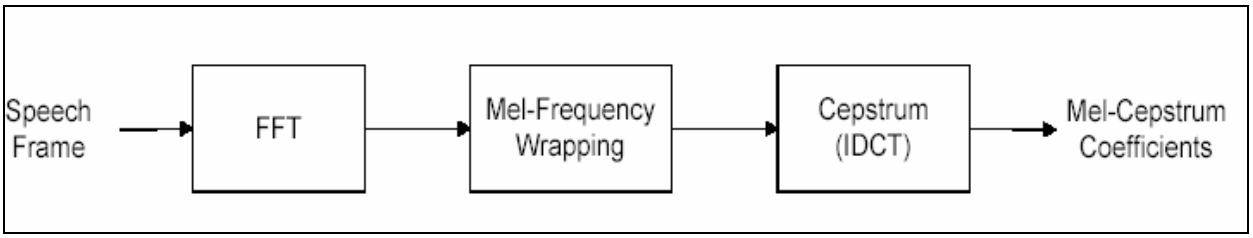

Figure 7: Block diagram of the feature extraction

4.2.2 LP-Coefficients

LPC based feature extraction is the most widely used method by developers of speech recognition systems [9]. The main reason is that speech production can be modeled completely by using LP analysis. Besides, LPC based feature extraction can

also be used in speaker recognition systems, where the main purpose is to extract the vocal tract parameters from a given utterance [10]. In speech synthesis [11], linear prediction coefficients are the coefficients of the FIR filter representing a vocal tract transfer function. Therefore, linear prediction coefficients are suitable to use as a feature set in speaker verification systems.

The general idea of LP is to determine the current sample by a linear combination of p previous samples where the linear combination weights are the linear prediction coefficients [12]. Therefore, the LP polynomial can be written as:

$$\hat{x}(n) = \sum_{k=1}^{p} a(k)x(n-k) \quad (7)$$

Where x is the input speech frame. In our algorithm, LP coefficients must be calculated for each frame. The autocorrelation method is used to extract LP coefficients.

The algorithm of the autocorrelation method is as follows:

- Find (p+1) autocorrelation matrix elements R(k)'s such that:

$$R(k) = \sum_{m=0}^{N-1-k} x(m)x(m+k), \quad \text{k} = 0,1,\ldots \text{p} \quad (8)$$

Where N is equal to the frame size in samples

- Compute p autocorrelation coefficients α(k)'s as follows:

$$\begin{bmatrix} R(0) & R(1) & \ldots & R(p-1) \\ R(1) & R(1) & \ldots & R(p-2) \\ \ldots & \ldots & \ldots & \ldots \\ R(p-1) & R(p-2) & \ldots & R(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \\ \ldots \\ a(p) \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \ldots \\ R(p) \end{bmatrix} \Rightarrow \begin{bmatrix} a(1) \\ a(2) \\ \ldots \\ a(p) \end{bmatrix} = \begin{bmatrix} R(0) & R(1) & \ldots & R(p-1) \\ R(1) & R(1) & \ldots & R(p-2) \\ \ldots & \ldots & \ldots & \ldots \\ R(p-1) & R(p-2) & \ldots & R(0) \end{bmatrix}^{-1} \begin{bmatrix} R(0) \\ R(1) \\ \ldots \\ R(p) \end{bmatrix} \quad (9)$$

- Repeat Steps 1 and 2 for each frame of the utterance.

To see the effect of the linear prediction order in our system, we have used different values for "p". One of the values has been chosen according to the following formula:

$$P = \frac{f_S}{1000} + 2 \quad (10)$$

The sampling frequency is 16kHz, The block diagram of the LPC calculation is given in Figure 8.

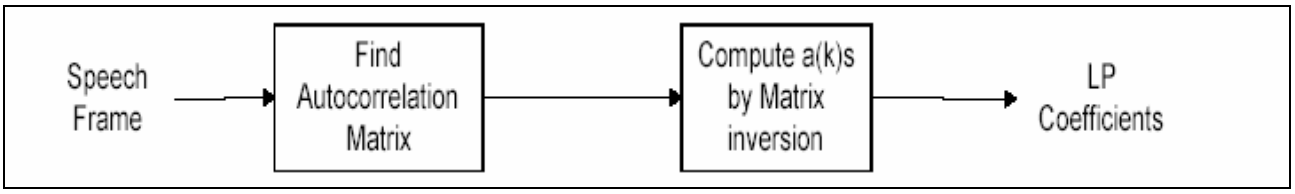

Figure 8: Block diagram of the LPC calculation

4.3 Fundamental Frequency (Pitch) and Magnitude

4.3.1 Short-Time Magnitude

Using an endpoint detection algorithm, the speech is selected from the two input waveforms, and then their short-time magnitudes are determined. The short-time magnitude characterizes the envelope of a speech signal by lowpass, filtering it with a rectangular window [13]. The magnitude function follows these steps:

- The bounds of the signal are determined, and each end is zero-padded.
- The signal is convolved with a rectangular window. As the window is swept across the signal, the magnitude contained within the signal is summed and plotted at the midpoint of the window's location. This provides the speech with the cover needed for the speech to be uttered correctly.

One magnitude plot is discrete time warped onto the other, see Figure 9. The dot product of the two waveforms is computed, and this number is divided by the product of the signals' norms. This calculation results in a percentage of how similar one signal is to another.

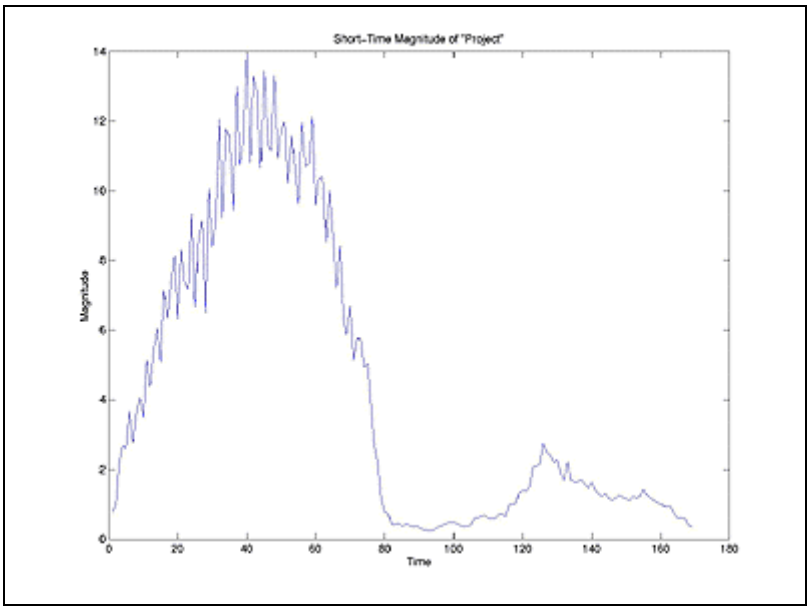

Figure 9: short time magnitude of the signal

4.3.2 Short-Time Frequency

A simple model of the human vocal tract is a cylinder with a flap at one end. When air is forced through the tract, the vibration of the flap is periodic. The inverse of this period is known as the fundamental frequency, or pitch. This frequency, combined with the shape of the vocal tract, produces the tone that distinguishes your voice. Variations in people's vocal tracts result in different fundamentals even when the same word is said. Therefore, pitch is another characteristic of speech that can be matched [14].

Since harmonic peaks occur at integer multiples of the pitch frequency, we can compare peak frequencies at each time t to locate the fundamental to extract pitch from signals, making use of a harmonic-peak-based method.. The implementation finds the three highest-magnitude peaks for each time, then thw implementation computes the differences between them. Since the peaks should be found at multiples of the fundamental, now their differences should represent multiples as well. Thus, the differences should be integer multiples of one another. Using the differences, the implementation derives the closest voice for the fundamental frequency.

The pitch frequency is computed at each time, and this gives the pitch track of the signal. A major advantage of this method is that it is very noise-resistive. Even as the noise increases, the peak frequencies should still be detectable above the noise. It is also easily implemented in MATLAB [15].

The first step is to find the signal's spectrogram. The spectrogram parameters decided here are a window length of 512 points and a sampling rate of 10000 Hz. assuming that the fundamental frequency (pitch) of any person's voice will be at or below 1200 Hz, so when finding the three largest peaks, only consider sub-1200 Hz frequencies, cutting out the rest of the spectrogram. Before that, use the whole spectrogram to find the signal's energy [16].

The signal's energy at each time is very important as it shows the voiced and unvoiced areas of the signal, with voiced areas having the higher energies. Since using our pitch track to compare the pitch between signals, be certain that the comparison held only for the voiced portions, and the areas where the pitch will be distinct between two different people. A plot of energy versus time can actually be used to window the pitch track so that only the voiced portions are taken.

To find the energy versus time window, take the absolute magnitude of the spectrogram and then square it. According to Parseval's Theorem, adding up the squares of the frequencies at each time gives us the energy of the signal there Plotting this versus time gives us our window.

Once this is done, cut the spectrogram and move on to finding the three largest peaks at each time. A frequency is designated as a "peak" if the frequencies directly above and below it have smaller magnitudes than it does. If a frequency is a peak, then its magnitude is compared to the three magnitude values stored in the "peak matrix" (a matrix of magnitudes and locations for the three highest peaks which start out as zeros at each time). If it is greater than the minimum matrix value, then its magnitude and location replace the magnitude and location of the matrix's smallest peak.

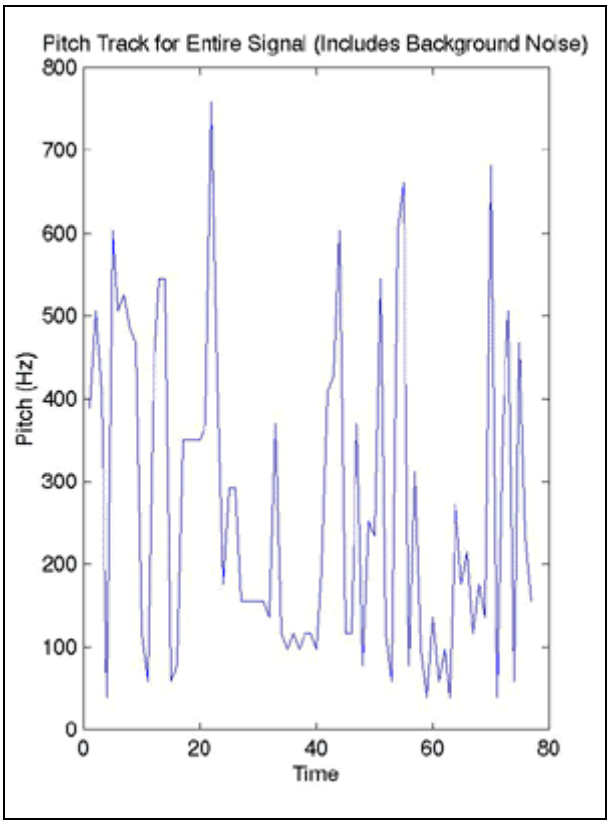

Figure 10: pitch track for entire signal (include background noise)

The matrix of peak values and locations at each time is then fed through the fundamental frequency algorithm, and we have our uncut pitch track (above) [17]. At this point, go back to the energy versus time plot and use it to find the energy threshold of the noise and unvoiced areas that cut out of the pitch track. This is done by finding the mean and standard deviation of the very beginning of the signal (assumed to be noise as the person never begins speaking until at least half a second into the recording due to mental processing time) and using these to develop the threshold. Then, the pitch track is windowed with the energy signal, and everything below the threshold is cut out (below). This gives a pitch track of the voiced portions of the signal. It is now ready for comparison with another signal.

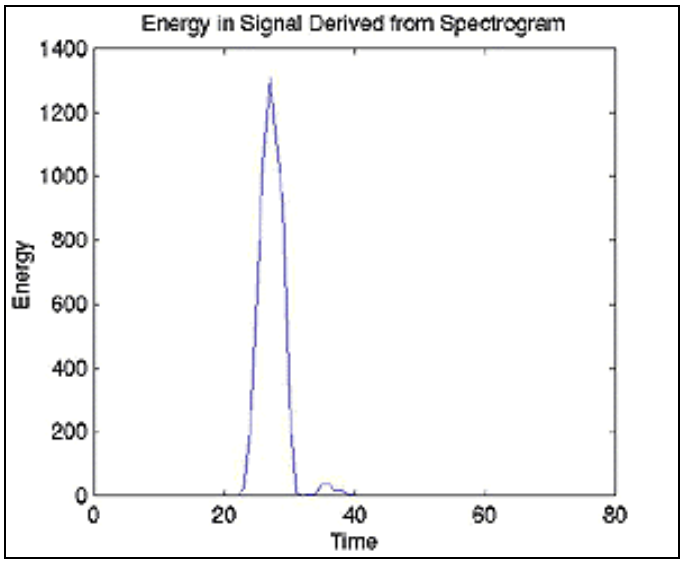

Figure 11: Energy in signal derived from spectrogram

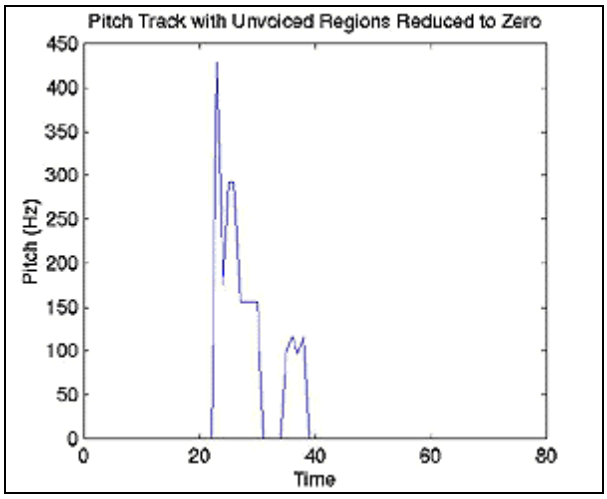

Figure 12: pitch track with unvoiced regions reduced to Zero

Pitch track comparison takes two signals and finds each of their pitch tracks. It then maps the pitch tracks onto one another using dynamic time warping. After mapping, take the dot product of the two tracks and equally divide it by the norms of the tracks to find the percent that they match. This is done twice, mapping the first signal onto the second and then vice versa, and then the highest dot product is taken as the matching correlation.

4.4 Feature Comparison

After the feature extraction step, the similarity between the parameters derived from the spoken utterance [18] and the reference parameters need to be computed. The three most commonly encountered algorithms in the literature are: Dynamic Time Warping (DTW), Hidden Markov Modeling [19] (HMM) and Vector Quantization (VQ). In our paper, we use DTW for comparing the parameter matrices extracted in the previous section.

4.5 Dynamic Time Warping

One of the difficulties in speech recognition is that although different recordings of the same words may include more or less the same sounds in the same order, the precise timing - the durations of each sub word within the word - will not match. As a result, efforts to recognize words by matching them to templates will give inaccurate results if there is no temporal alignment [20].

Although Hidden Markov Models (HMM) have largely superseded it, early speech recognizers used a dynamic-programming technique called Dynamic Time Warping (DTW) to accommodate differences in timing between sample words and templates. The basic principle is to allow a range of 'steps' in the space of (time frames in samples, time frames in templates) and to find the path through that space that maximizes the local match between the aligned time frames, subject to the constraints implicit in the allowable steps. The total `similarity cost' found by this algorithm is a good indication of how well the sample and template match, which can be used to choose the best-matching template.

Because of the emotional instability of the user, some utterances of the user might differ from each other. The reference recording might sound “project”, and the compared recording might sound like “prooject” depending on the mood of the user. Obviously, a simple linear squeezing of this longer password will not match the key signal because the user slowed down the first syllable while he kept a normal speed for the "ject" syllable. (i.e., can compare "Prrrooo" to "Pro" and "ject" to "ject").

DTW is used to align the two different samples of the same utterance in the time domain. Furthermore, the algorithm is given as follows:

- Each column of the parameter matrix is considered as a point in the vector space.
- A Local Distance Matrix (LDM) is defined such that each element of the matrix is a distance from one point (parameter matrix column) to another. As (wrong use of as, unless you mean as it's here, it's there, so it should be without and) the distance metric one is used for the LP coefficients, another distance is used for the MFCC and spectral coefficients. If the numbers of columns in the reference and the new parameter matrices are M and N respectively, the LDM becomes an MxN matrix.
- An accumulated distance matrix is computed such that each element of the matrix contains the corresponding LDM matrix element plus the smallest neighboring accumulated distance.

- To calculate the ADM, we start by equating *ADM(1,1) to LDM(1,1)*
- The other elements of the ADM are calculated as follows:
  ADM (m,n) = LDM(m,n) + min { ADM(m,n-1), ADM(m-1,n-1), ADM(m-1,n)}
- When the lower right corner of the matrix is reached, this value shows the Minimum global distance between the two-parameter matrices.

If two identical matrices are fed as the input to this algorithm, the minimum global distance becomes zero. The output of the DTW algorithm is the minimum global distance value. Small values mean similar parameter matrices, and high values indicate that the two parameter matrices are highly unlikely.

4.6 Decision Making and its Function

There are usually 3 approaches to construct the decision rules:

- Geometric.
- Topological.
- Probabilistic rules.

The most popular speech verification models are based on probabilistic rules [21].

In classical speech verification systems, when no prior information is given on the cost of the different kinds of errors, the Bayes Decision rule is applied by selecting the value of Δ_i in (5) that minimizes the Half Total Error Rate:

$$HTER = 1/2(\%FA + \%FR) \quad (11)$$

where %FA is the rate of false acceptances, and %FR is the rate of false rejects. Note that this cost function changes the relative weight of client and impostor accesses in order to give them equal weight, instead of the one induced by the training data.

If the probabilities are perfectly estimated, then the Bayes Decision is the optimal decision. Unfortunately, this is usually not the case. In that case, the Bayes Decision might not be the optimal solution, and we should thus explore other forms of decision rules. In this thesis, we will discuss two sorts of decision rules, which are based either on linear functions or on more complex functions such as Support Vector Machines.

5. Implementation of the Speaker Verification

5.1 Step 1: Trained the System and Build a Dataset

When a speaker attempts to verify himself with this system, his or her incoming signal is compared to that of a "key" [22]. This key should be a signal that produces a high correlation for both magnitude and pitch data when the authorized user utters the password, but not in cases where:

- the user says the wrong word (the password is forgotten).
- an intruder says either the password or a wrong word.

5.2 Step 2: Verification

Once a key has been established, an authorization attempt breaks down into the following steps:

1. The person (for example, person 1) utters the password using the microphone.

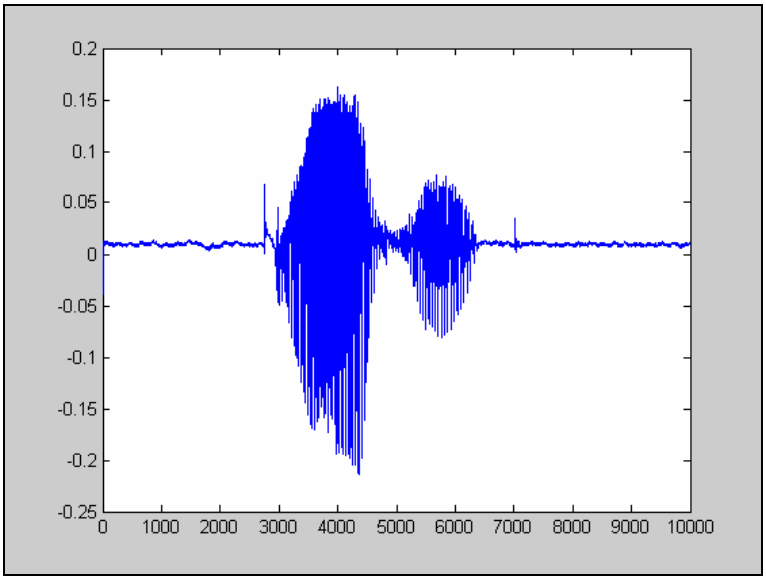

Figure 13: Person 1 signal (sig1)

2. The short-time frequency of the signal is found and recorded just as the pitch track is.

PITCH takes a spectogram-MxN matrix returned from *spectrogram* function and a corresponding sample - Scalar holding sample rate - to create a pitch track and a plot of the fundamental frequency versus time. It then eliminates false pitch estimates by zeroing the areas of low energy because no one is talking or because a fricative voice occurs, see Figures 14, 15 and 16.

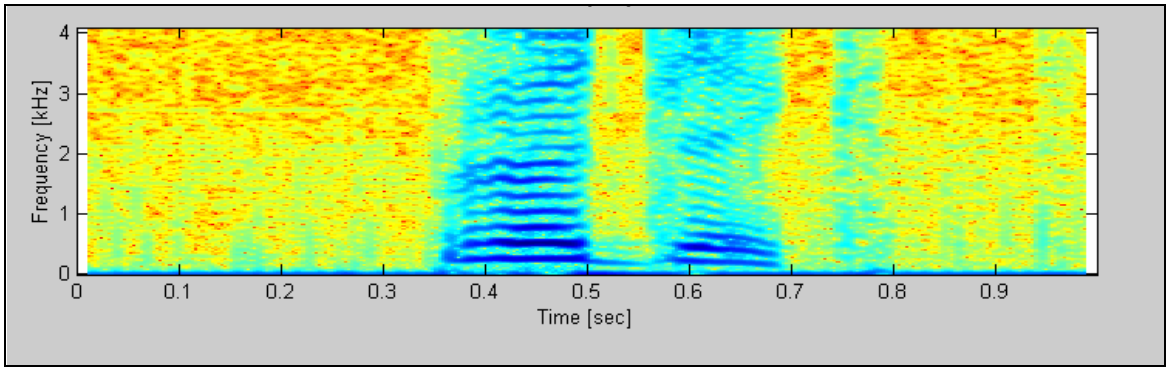

Figure 14: Spectrogram of the key signal

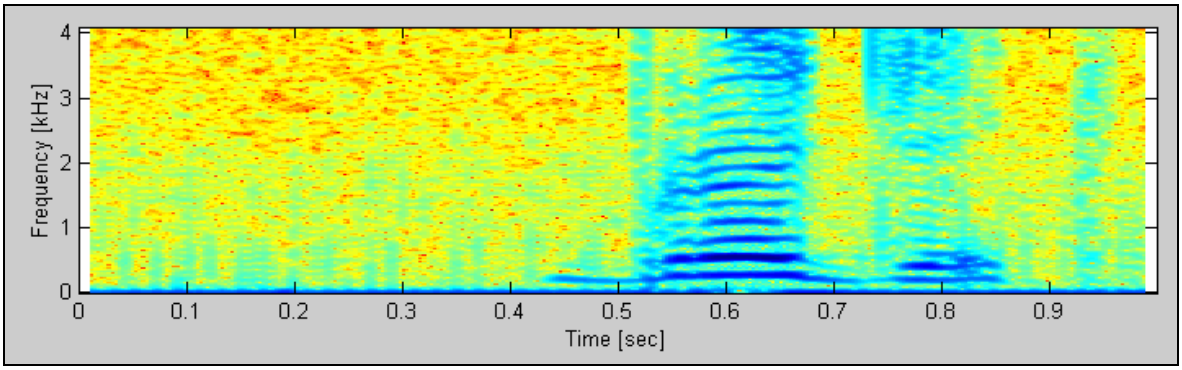

Figure 15: Spectrogram of the new signal

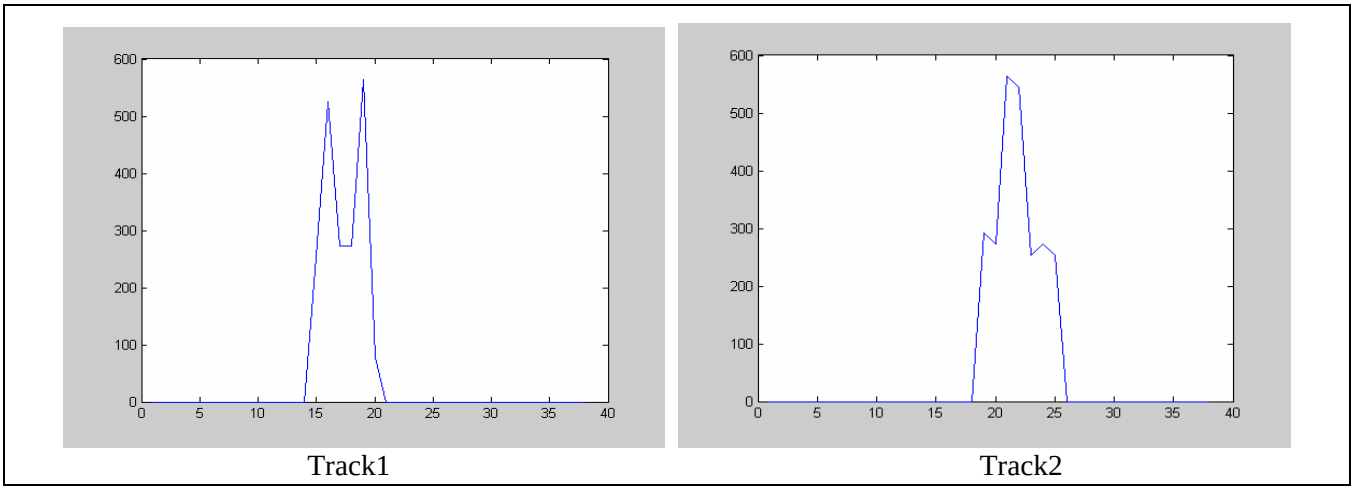

Figure 16: Track signals

To prepare for a Dynamic Time Warp (DTW) [23]:

1- Cut the trailing and leading zeros of track1, see Figure 17.a.

2- Cut the trailing and leading zeros of track2, see Figure 17.b.

3- Perform the Dynamic Time Warping by mapping track2 onto track1, and calculating the dot product, then map track1 onto track2 and calculate the dot product, and finally, map track2

onto track1 and calculate the dot product (see Figure 18).
Where:
track2dtw=track2cut (path2on1)

dotWDTW1 = dot(track1cut,track2dtw)
/(norm(track1cut)*norm(track2dtw));

4- Map track1 onto track2 and calculate the dot product, as in Figure 19.

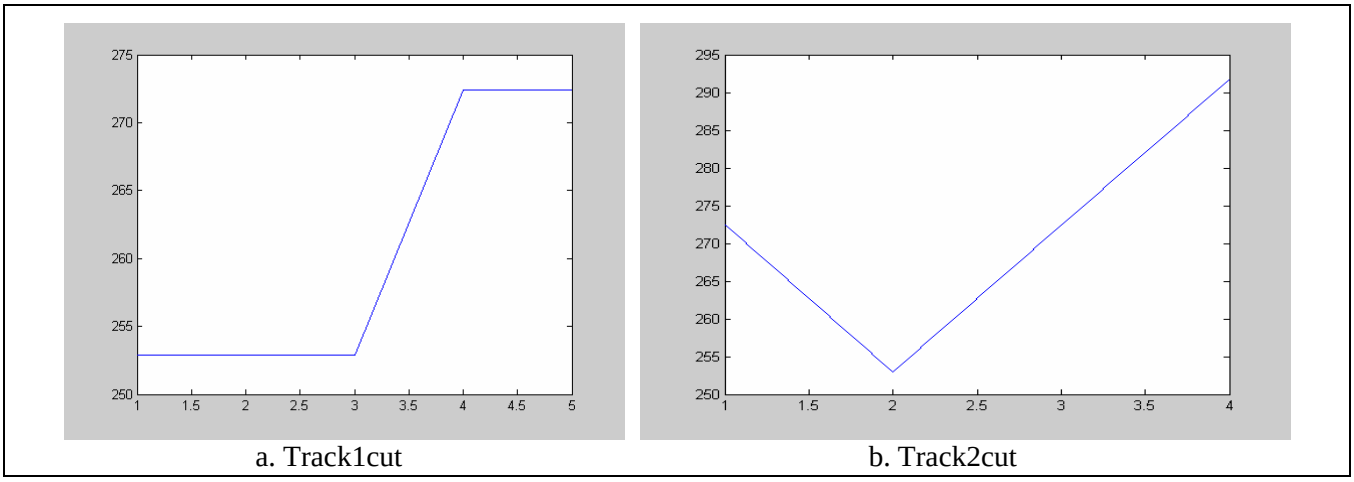

Figure 17: Track signals after cutting

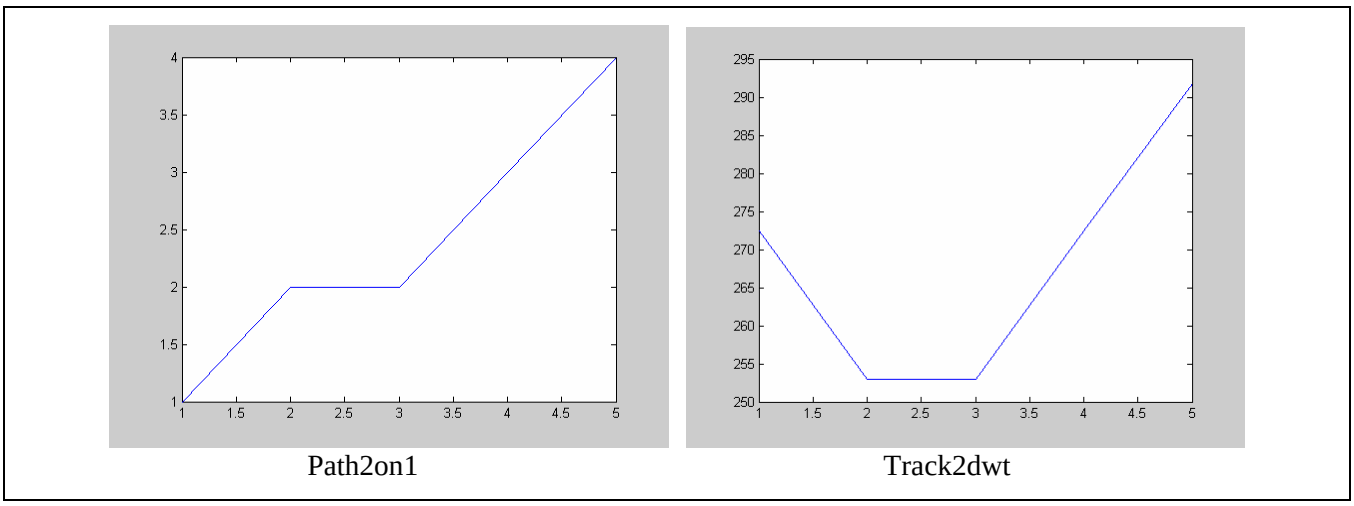

Figure 18: Path2on1 and Track2dtw

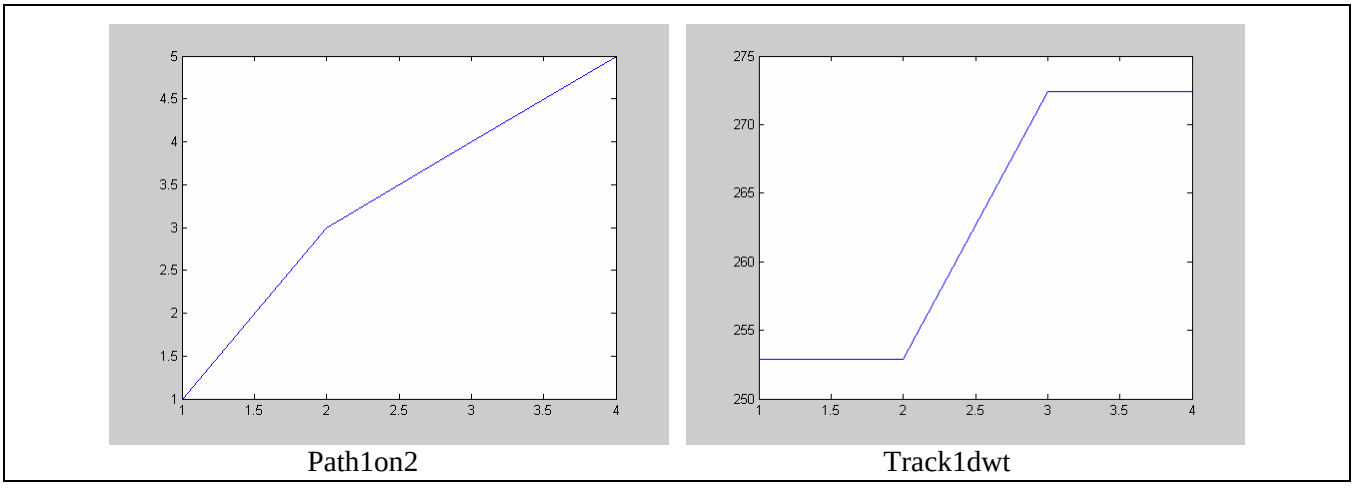

Figure 19: Path1on2 and Track1dtw

Where:
track1dtw = track1cut (path1on2);
dotWDTW2=
dot(track2cut,track1dtw)/(norm(track2cut)*norm
(track1dtw));
Take Larger Dot Product as Result
corr = pitchmaster (sig, locksig)
ans = 0.9996 so that pitch=0.9996

5. The short-time magnitude of the signal is found and recorded, as is the magnitude. Then, find out where to cut the two signals and find the average magnitude for the two cutting signals, as in Figure 20.
6- Perform Dynamic Time Warping (DTW) linking one signal to another in order to let the peaks match up. The time warp is performed both ways (matching sig1 to sig2 and vice versa).
The output will be the best match of the two signals.
1- Let sig1 be the key and sig2 be time-warped to sig1 (see Figure 21.a).
2- Let sig2 be the key and sig1 be time-warped to sig2 (see Figure 21.b).

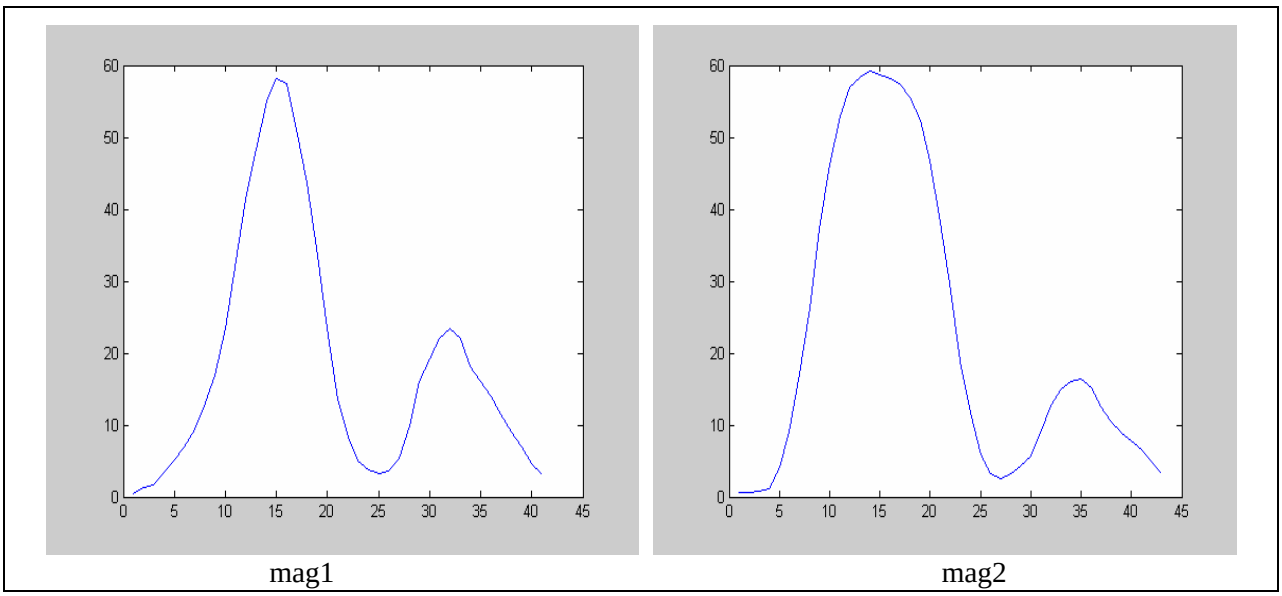

Figure 20: Magnitude for the two cutting signals

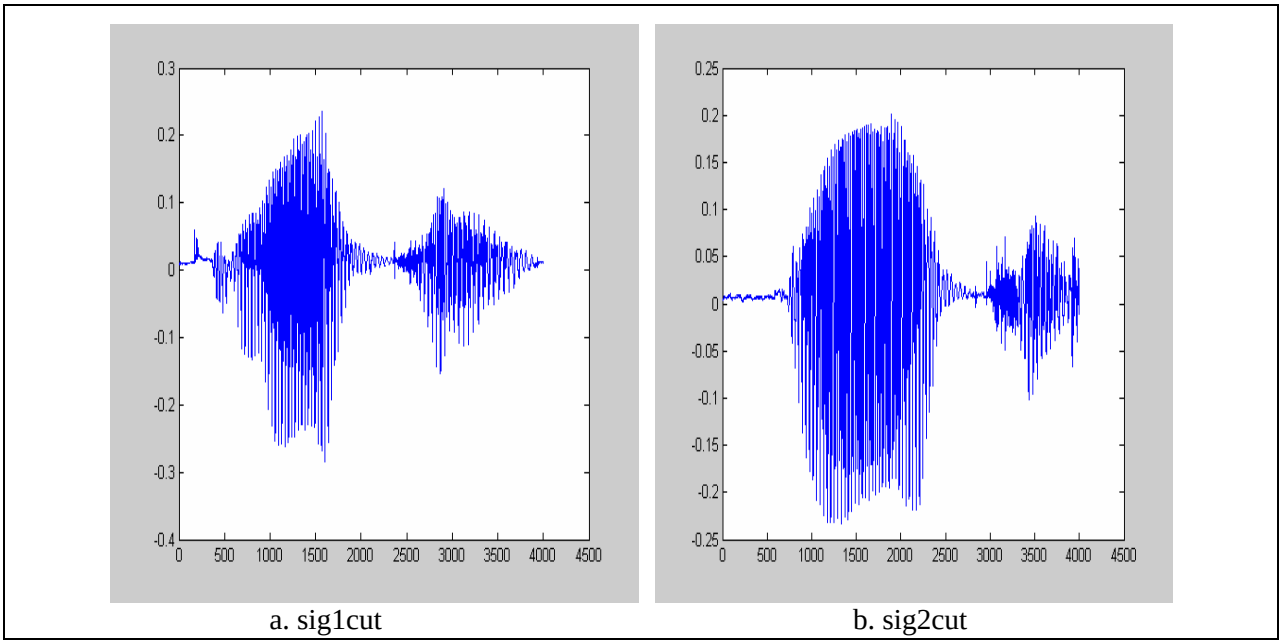

Figure 21: Sign1 and Sign2 cuts

Then for each signal (DTW), the following calculations are performed (see Figure 22):

```
num = dot(mag1,mag2(path1));
den = (norm(mag1))*(norm(mag2(path1)));
final1 = num/den;
num = dot(mag1(path2),mag2);
den = (norm(mag1(path2)))*(norm(mag2));
final2 = num/den;
```

Take the larger result which represents the mag value

```
>> final1
  final1 =  0.9916
>> final2
  final2 =  0.9952
```

So that mag=0.9916

7- These numbers are compared to the thresholds (0.95). If both the magnitude and pitch correlations are above this threshold, the speaker has been verified.

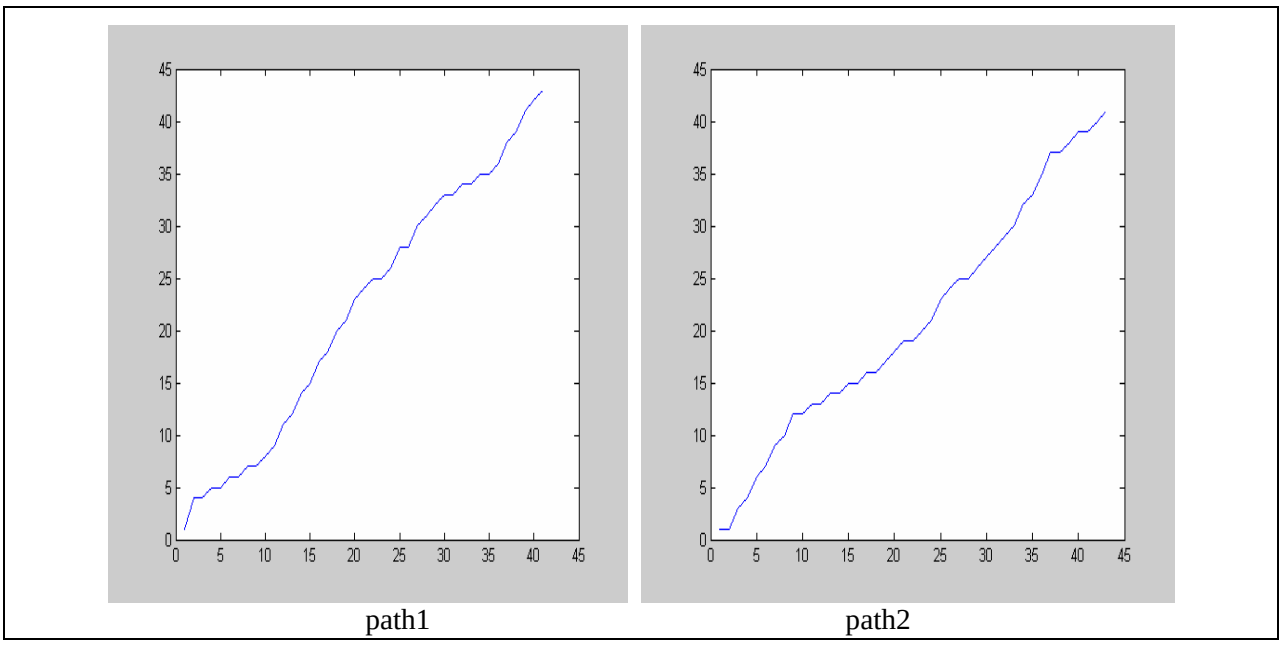

Figure 22: Path1 and Path2

6. Experimentation and Results

To develop such a key, the system is trained for the recognition of the speaker. In this instance, the speaker first chooses a password, which is acquired five separate times as shown in Figure 23. The pitch and magnitude information are recorded for each. The signal that matches the other four signals best in both cases is chosen as the key as shown in Figure 24.

Furthermore, the system will return lower thresholds for matching to magnitude vs. time and pitch vs. time. Thresholds below 0.90 are not returned, and to eliminate the possibility of an extremely high threshold like 0.99, an upper bound is placed on the thresholds of 0.95

In MATLAB, the function makelock.m was written to determine the key signal from five possible signals. The call is: [Lock signal] = makelock (sig1, sig2, sig3, sig4, sig5)

In this instance, Khalid records his chosen password "project" five times and saves them as sig1, sig2, etc. When the call is made, the results are assigned to an array lock (which holds the time signal, which is a large array of points) and two scalars with the lower threshold bounds.

```
>> [Lock, pitchthreshold, magthreshold]
=makelock (sig1, sig2, sig3, sig4,sig5);
>> pitchthreshold
   pitchthreshold=0.9500
>> magthreshold
   magthreshold=0.9500
>> >> sum (sig1-locksig)
   ans =-2.4974
>> sum (sig2-locksig)
   ans = -2.4199
>> sum (sig3-locksig)
   ans = 0
>> sum (sig4-locksig)
   ans = -2.8141
>> sum (sig5-locksig)
   ans = -1.8363
```

Based on the above, sig3 has been chosen as the key.

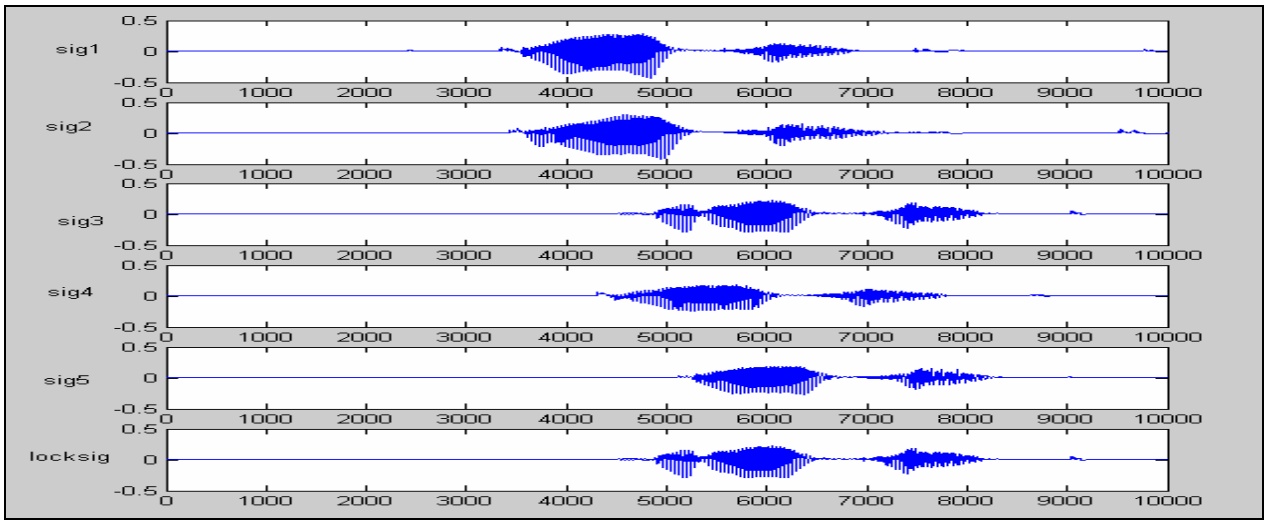

Figure 23: Five Signals of the Same Person

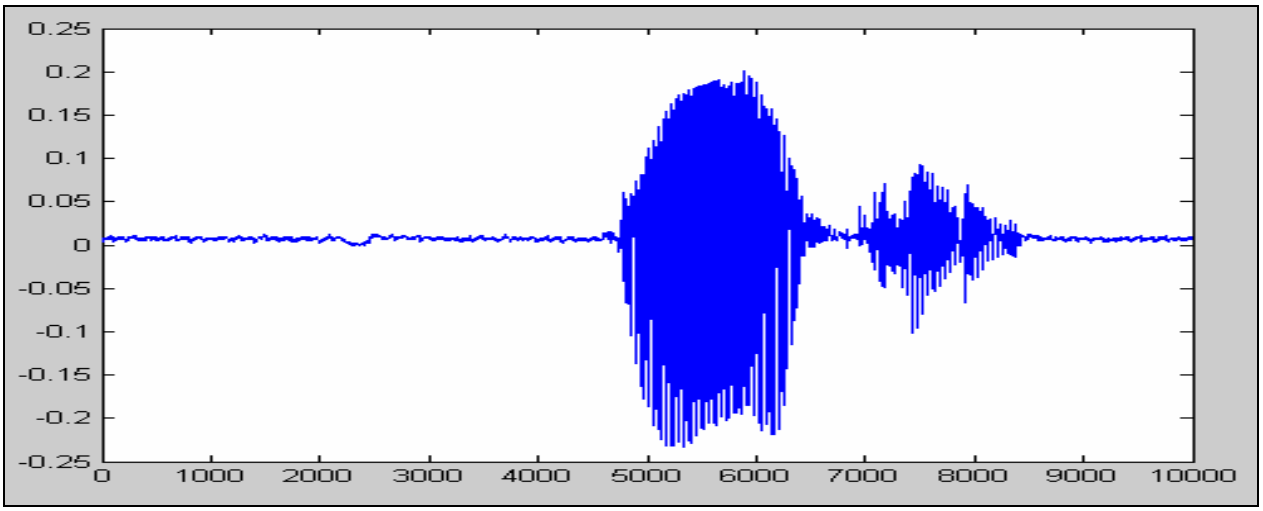

Figure 24: Key Lock Signal

6.1 Examples

>> speak now
Key and results: Pitch 0.99955,
Magnitude 0.98929,
******MATCH*****
>> speak now
Key and results: Pitch 0.99957,
Magnitude 0.98537,
******MATCH*****
>> speak now
Key and results: Pitch 0.94346,
Magnitude 0.97949,
****** NO MATCH ******
>> speak now
Key and results: Pitch 0.95588,
Magnitude 0.95045,
******MATCH*****
>> speak now
Key and results: Pitch 0.99937,
Magnitude 0.98412,
******MATCH*****
>> speak now
Key and results: Pitch 0.95457,
Magnitude 0.82971,
******NO MATCH ******

6.2 Results

To test the system, the two members of the group (Person 1, Person 2) were set as keys during the runtime of the system. The basic results are shown in Table 2 and Table 3.

Table 2: Comparing Ratio

	Person 1	**Person 2**
Person 1	67%	17%
Person 2	19%	74.5%

Table 3: Result table

	Person 1	**Person 2**
Person 1 and Person 2 (with Password)	67%	74.5%
Intruder (with Password)	21%	18.5%

7. Discussion

Through the previous section, we have provided a set of examples to test the systems. In Table 2, we gave two passwords for two persons; as a result we noted that person 1 has accessed the system successfully with ratio 67% using the given password, while person 2 has accessed the system successfully with ratio 74.5%, also with the given password, using 0.95 thresholds. Subsequently, we reduced the threshold to 0.90 and asked person 1 and person 2 to access the system, as a result, person 1 has accessed the system successfully with ratio 17%, while person 2 has accessed successfully with ratio 19%

We gave an intruder the two passwords, and then he has successfully accessed the system with ratios 21% and 18.5% using person 1 and person 2 passwords, respectively. Table 3 represents these results.

In order to accomplish the results in Tables 2 and 3, person 1, person 2 and the intruder have tried to access the system 100 times using the given passwords. Furthermore, person 1 with person 2 password has tried to access the system 100 times, and the same has been done for person 2 with person 1 password. In addition, an intruder has been given the passwords of person 1 and person 2, then he has tried to access the system 200 times using the passwords of person 1 and person 2.

8. Conclusion and Future Work

Many courtiers around the world have just started their e-government programs. E-government portals will be increasingly used by the citizens of many countries to access a set of services. Currently, there are many challenges of the use of the e-government portals; one of these challenges is the security issues. E-government portals security is a very important characteristic in which it should be taken into account. In this paper, we have incorporated the biometric voice technology into the e-government portals in order to increase the security and enhance the user verification. In this way, the security should be increased since the user needs to use his voice along with his password. Therefore, no any unauthorized person can access the e-government portal even if he gets a password.

The proposed system was designed to be installed on the server of the e-government to connect clients (especially G2G) using the voice instead of or beside the traditional username and password. The reason behind that is that most of the financial transactions are done via the portal of e-Government. Using this system, it would not be possible for a person to deny his or her access to the portal as it will authenticate the person and make sure it is the person allowed to access the data. This system was not designed for Business to Government (B2G) and Consumer to Government (C2G) as they both have unlimited number of people, which is not the case in the Government to Government model.

Another application that makes use of all the calculations and theorems mentioned in this thesis is the handicapped chair, which was implemented and scored a great success.

The training should be done over a period of time to take into account differences in background noise, the speaker's health, microphones, and other various factors. Furthermore, an instantaneous key could be skewed because the user will attempt to sound similar at each training session, which will actually produce dissimilarities.

A composite key could help alleviate this problem by taking the five signals that are used in training and producing an "average" of them in terms of magnitude and fundamental frequency. Furthermore, the results could improve over time if the system took successful attempts and added them to the average key. In this case, the recognition could get better over time without extra training sessions.

The current lower thresholds for matching determined by makelock.m are set rather high, in a range between 0.90 and 0.95. The reason why the number is set high is to provide the utmost security for your password in the worst case scenario, which is when it is known to the intruder. As seen in the results, intruders without knowledge of the password were never successful in accessing the system. Therefore, it is possible to set the thresholds lower so that the owner's acceptance rates increase while an intruder's acceptance rate remains nominal.

Finally, a speaker's voice has many other characteristics that make it identifiable to the human ear. Computationally, a system can also try to draw

out such unique differences and match those to users as well. For example, formants are a function of a person's vocal tract, so tracking such data could improve further results. As mentioned earlier, using Linear Predictive Coding (LPC) and spectrum techniques may produce more accurate results that increase the rates of acceptance and denial.

This system can be developed in the future through connecting it to other software programs or through the authentication process, which is done through the Iris or the fingerprint. The voice will be widely used in the coming software programs. The system could also be installed on special chips as an embedded system that would work on its own.

References :

[1] Jain, A. K.; Ross, A. and Prabhakar, S., "An introduction to biometric recognition", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 1, 2004, pp. 4-20.

[2] Monrose, F.; Reiter, M. K.; Li, Q. and Wetzel, S., "Cryptographic Key Generation from Voice", in *Proceedings of the IEEE Symposium on Security and Privacy - S&P'01*, Oakland, CA, USA, May 14-16, 2001, pp. 202-213.

[3] Bagge N. and Donnica C., Text Independent Speaker Recognition, *online: http://www-dsp.rice.edu/courses/elec301/Projects01/speaker_id/index.html*, accessed on March 16, 2008.

[4] Jian A. K. and Ross, A., "Multibiometric Systems", *Communications of the ACM*, Vol. 47, No. 1, 2004, pp. 34-40.

[5] O'Shaughnessy, D., *Speech Communications: Human and Machine*, the Institute of Electrical and Electronics Engineers - IEEE, Inc., New York, 2000.

[6] Guo, S. and Liddel, H, "Support Vector Regression and Classification Based Multi-view Detection and Recognition", in *Proceedings of the IEEE International Conference on Automatic Face & Geusture Recognition*, 2000, pp. 300-305.

[7] Peacocks, R. and Graf, D., An Introduction to Speech and Speaker Recognition, *IEEE Computer Magazine*, Vol. 23, No. 8, 1990, pp. 26-33.

[8] Davis, S. B. and Mermelstein P., "Comparison of parametric representations for Monosyllabic word recognition in continuously spoken sentences", *IEEE Transactions on Signal Processing*, Vol. 28, No. 4, 1980, pp. 357-366.

[9] Gasem, M., Vector Quantization, *online: http://www.geocities.com/mohamedqasem/vectorquantization/vq.html*, accessed on March, 2008.

[10] Xafopoulus, A., *Speaker Verification: an Overview*, Technical Report, Artificial Intelligence & Information Analysis laboratory, Informatics Department, Aristotle Univ. of Thessaloniki, Thessaloniki, Greece, 2001.

[11] Childers, D. G., *Speech Processing and Synthesis Toolboxes*, John Wiley & Sons Inc., New York, 2000.

[12] Gold, B. and Morgan, N., *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, John Wiley & Sons Inc., New York, 2000.

[13] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, N.J., 1993.

[14] Minh, N., An Automatic Speaker Recognition System, *online: http://lcavwww.epfl.ch/~minhdo/asr_project/asr_project.html*, accessed on March 16, 2008.

[15] L.R Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, N.J., 1978.

[16] Matlab VoiveBox, *online: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html*, accessed on March 16, 2008.

[17] Yu, K.; Mason, J. and Oglesby, J., "Speaker Recognition Using Hidden Markov Models, Dynamic Time Warping and Vector Quantization", in *IEE proceedings on Vision, Image and Signal Processing*, Vol. 142, No. 5, 1995, pp. 313-318.

[18] Deller, J. R.; Proakis, J. G and Hansen J. H., *Discrete-Time Processing of Speech Signals*, MacMillan, New York 1993.

[19] Arslan, L., Digital Speech Processing, EE 578 Lecture Notes, Boðaziçi University, Turkey, spring 2001.

[20] Rafael, G. and Richard, W., *Digital Image Processing*, 2nd edition, Prentice-hall, 2002.

[21] Fant, G., Acoustic Theory of Speech Production and Some of its Implications, *Journal of Speech and Hearing Research*, Vol. 4, pp. 303-320, 1961.

[22] Minh, N., An Automatic Speaker Recognition System, Audio Visual Communications Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland, February 2003.

[23] Ross, A. and Jain, A., Biometrics: Speaker Verification, *online: http://biometrics.cse.msu.edu/speaker.html*, accessed on March 16, 2008.