SDP Team 23 MDR Presentation

**Phillip Ashe, Liam Shea,
Jason Nguyen, Paul Mahoney**

Speaker Verification System (SVS)

Phillip Ashe-CSE

Database development and access code

Liam Shea - EE

Feature extraction and analysis

Paul Mahoney-CSE

Feature extraction and analysis

Jason Nguyen - EE

Feature extraction and analysis

Overview - Problem

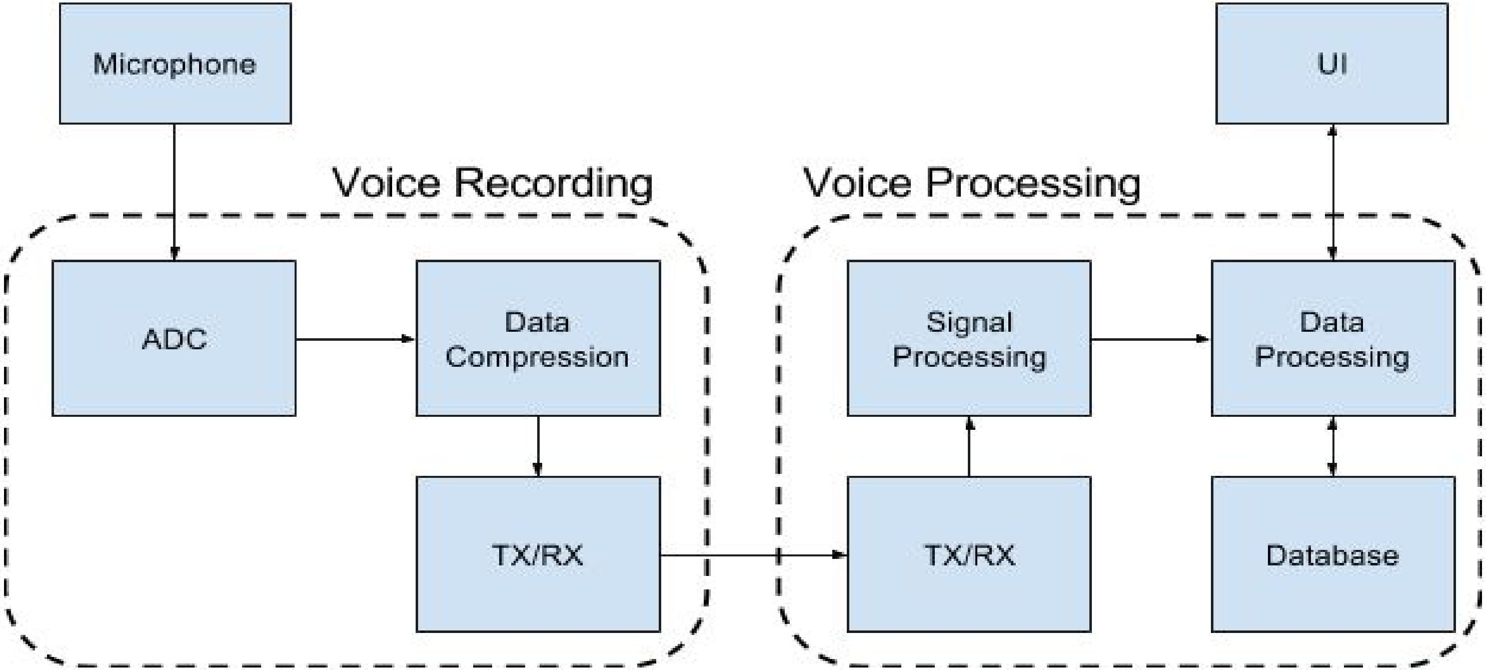
The common multi-step, text-based verification system (password protection and security questions) has many issues:

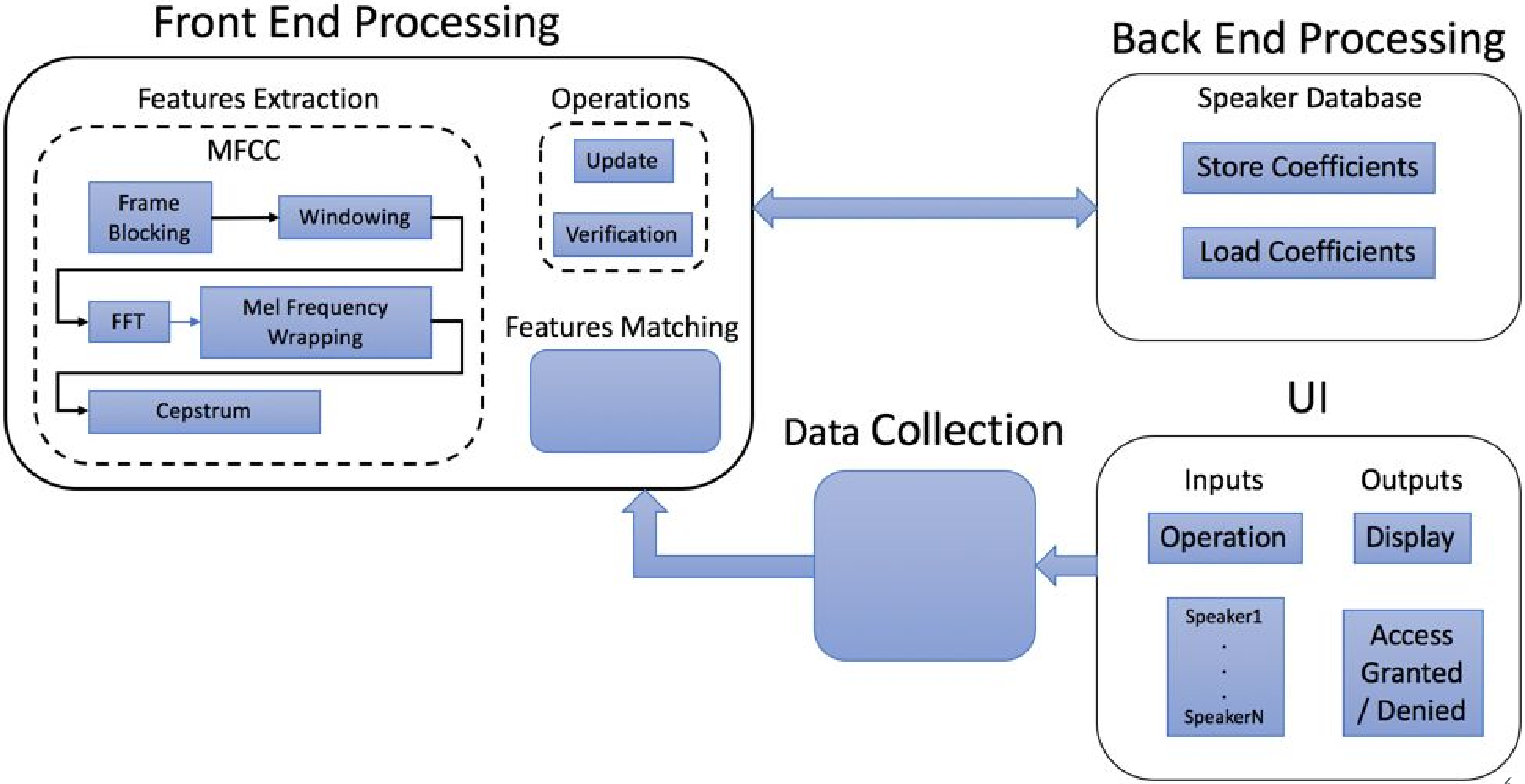
1. Traditional text-based password protection has grown more susceptible to hacking
2. Adding more layers of text-based protection proves to be equally ineffective due to more memorization and advanced cryptography algorithms.
3. Users tend make poor passwords that are easy to decrypt
4. Forcing users to make better passwords leads to forgetfulness and discourages users from using the service
5. Passwords, once stolen, can be used by anyone and the user might not be aware of
6. Cumbersome to enter, worse with increasing password complexity
7. Security questions can be too easy; known by others besides the user

Overview - Solution

1. Using biometrics which are readily available
2. These biometrics are more complex than human developed cryptographic algorithms
3. Among current biometrics-employed security systems, vocal based verification is more widely accepted due to their ease of use and relatively high accuracy
4. Can be incorporated into and strengthens security systems already existing

Previous block diagram





Front End Processing

Features Extraction

MFCC

Frame Blocking

Windowing

FFT

Mel Frequency Wrapping

Cepstrum

Operations

Update

Verification

Features Matching

Back End Processing

Speaker Database

Store Coefficients

Load Coefficients

Data Collection

UI

Inputs

Operation

Speaker1
.
.
.
SpeakerN

Outputs

Display

Access Granted / Denied

System Requirements

1. System shall record voice in single-channel format at sampling rate of 8KHz
2. System shall perform feature extraction on user voice input
3. System shall verify the identify of a speaker with >90% confidence
4. System shall be able to query and insert entries into SQL database
5. System shall allow the user to choose modes of operation, and provide feedback

Subsystem: Front End Processing - Features Extraction

1. We use Mel Frequency Cepstrum Coefficients (MFCC) to extract vocal parameters.
2. The algorithm replicates human perception of sound, while taking into consideration the known variations in human critical hearing frequency, which is linear in below 1000Hz and logarithmic above 1000Hz.

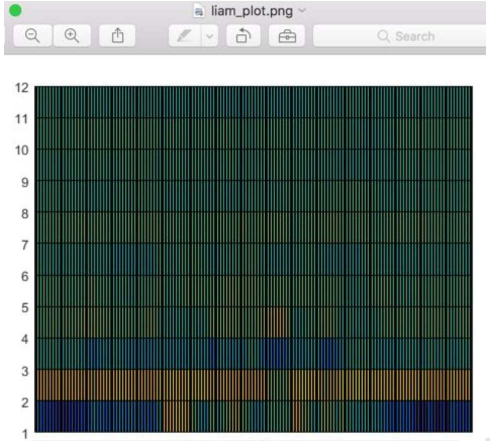
MFCC Algorithm

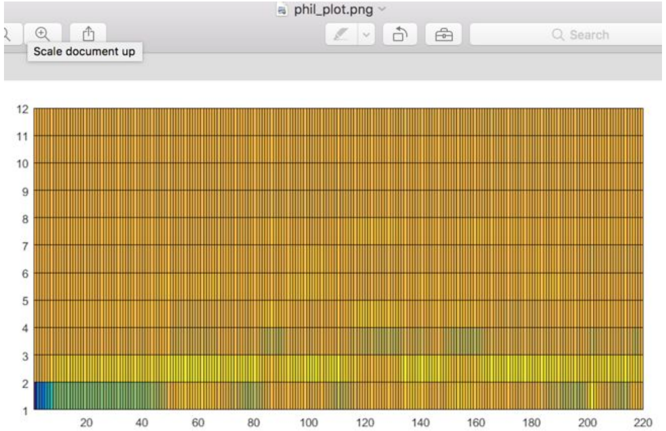
1. Segment input voice signal into frames
    a. Appropriate diarization of the sample yields a period over which the sample has minimal change in signal (statistically stationary)
    b. These stable segments allows us to analyze for a single characteristic representation
2. Apply Hamming window
    a. Hamming window focuses on the center of each frame, reducing distortion caused by discontinuities at both ends of the frame during blocking
3. Convert to frequency domain using FFT

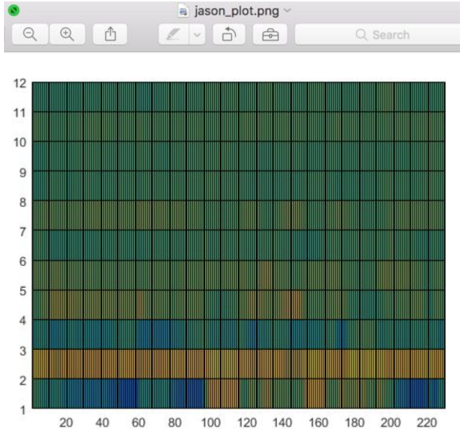
MFCC (cont.)

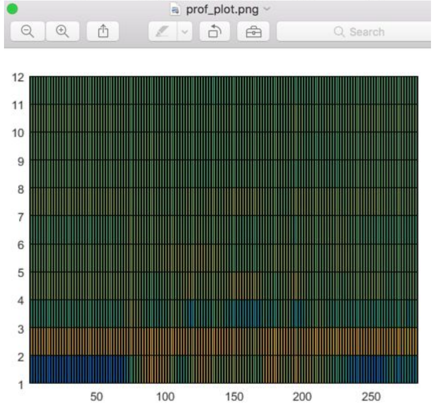
4. Apply mel filter bank to further simulate frequency response of human hearing
   a. Mel filter banks are triangular bandpass filters used to sum up power over a specified spectral region, since human ears can't discern closely spaced frequencies
   b. Filter banks are narrow at low frequencies to capture the linear relationship and become wider at high frequencies (i.e. we notice slight changes in quieter noises but not variations in loud noises)
5. Log mel spectrum is converted to time domain
   a. The result is the MFCC characteristics illustrated in the next slide
   b. Each frame, in time domain, has a distinct series of coefficient that distinguish them from other entries of the same phrase

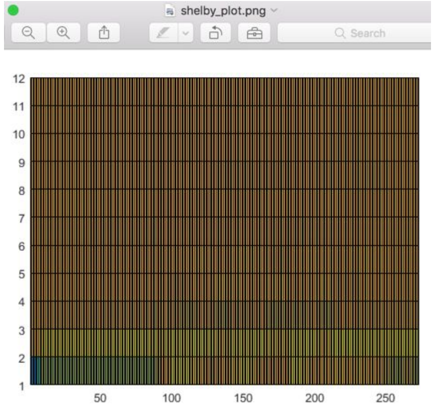
Sample MFCC Plots





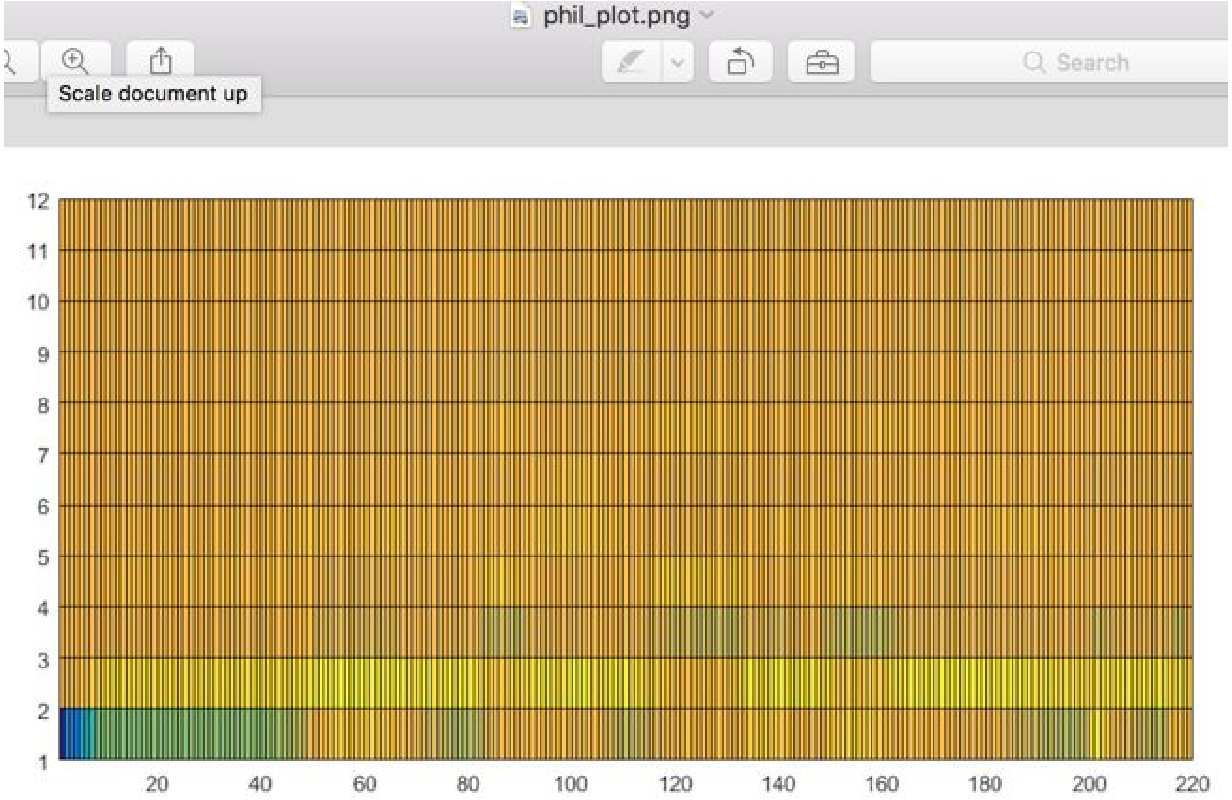






Y axis - Cepstral Coefficient Index
X axis - Frames in time domain

Sample MFCC Plots



Y axis - Cepstral Coefficient Index
X axis - Frames in time domain

Subsystem: Back-End Processing (Database)

Each database entry should contain:

1. Speaker ID
    a. Size:8B
    b. Type: char[]
2. Age of speaker
    a. Size:1B
    b. Type: byte
3. Sex of speaker
    a. Size:1B
    b. Type: char
4. MFCC coefficients
    a. Size:200 frames*12 cepstra/frame*8B/cepstra = 19200B
    b. Type: double[][]

Speaker ID	Age	Sex	MFCCs
‘27781127’	22	‘M’	[...]

This means each DB entry is 19210 Bytes

Subsystem: UI

1. Prompts user to enter their voice
2. Will have option to change mode of operation
    a. New entry Vs. Verification mode
3.
4. Button to begin stop/start recording
5. Notify user of verification result

Proposed MDR Deliverables

1. Demonstrate ability to record voice sample in digital format.
   a. This was carried out using the built-in laptop microphone
   b. The recorded voice samples are contained in arrays of 2 channels, where we take the average
2. Demonstrate communication between Voice Recording and Voice Processing subsytems
   a. The recorded samples are stored on local machine and called in for processing using MATLAB
3. Implementation of signal processing in MATLAB
   a. Feature extraction implementation phase is complete
   b. The result we have is the MFCCs of the samples

Proposed CDR Deliverables

1. Implementation of MFCC algorithm on Raspberry Pi
2. Implementation of training algorithm
3. Database integration
4. UI implementation
5. Hardware integration

Role Assignment for CDR

Paul Mahoney: Implementation of MFCC algorithm on Raspberry PI

Liam Shea: Implementation of User Verification algorithm

Phillip Ashe: Server Implementation and Management

Jason Nguyen: UI / Interprocess Implementation and Hardware Integration

Component: Raspberry PI 3 - Model B

Part of front-end processing subsystem. This performs the bulk of our computations

Important specs:

1. 1.2 GHz 64-bit ARM processor
2. 1GB DRAM
3. Built-in 802.11n module
4. Dedicated GPU (good for FFT)
5. HDMI
6. 40 GPIO pins
7. Supports several flavors of Linux

Component: Display

Part of UI subsystem:

1. LCD Display
2. Touch or non Touch
3. Video received from HDMI
4. GPIO (General purpose I/O)

Component: Microphone

Part of voice recording subsystem

1. MP34DT04TR
   a. Output type: digital, PDM
   b. Frequency range: 100Hz - 10kHz
   c. Voltage supply: 1.6V - 3.6V
   d. Sensitivity: -26dB ±3dB @ 94dB SPL

2. GOTD USB 2.0 Mini Microphone MIC
   a. Noise-cancelling microphone filters out unwanted background noise
   b. Sensitivity:-67 dBV/pBar. -47 dBV/Pascal+/-4dB
   c. Frequency response:100-16kHz

Component: Database server

Part of back end processing subsystem. This is where our database will “live”

Two possible implementations:

1. Physical server
   a. Would need static IP Address
   b. Likely won’t reach max. resource utilization
2. Virtualized (cloud-based) server
   a. Scalable
   b. More reliable
   c. Cheap (Amazon, Google offer free resources for students)

Cost Estimation

RaspberryPi 3 - Model B (1 unit)	:	$35	element14
Raspberry Pi Touch screen display (1 unit)	:	$67	amazon
Microphone chip (TBD)	:	$0.93/unit	digikey
GOTD USB 2.0 Mini Microphone MIC (1 unit)	:	$3.50	amazon
Mid-range AWS VM (2 vCPUs, 8GB RAM)	:	$0.047/hr	amazon